VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA ANOMÁLIÍ V UŽIVATELSKÉM CHOVÁNÍ

USER BEHAVIOR ANOMALY DETECTION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. LUKÁŠ PETROVIČ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. JAN PLUSKAL**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání diplomové práce

Student: **Petrovič Lukáš, Bc.**
Program: Informační technologie Obor: Počítačové sítě a komunikace
Název: **Analýza anomálií v uživatelském chování**
User Behavior Anomaly Detection
Kategorie: Data mining

Zadání:

1. Seznamte se s problematikou analýzy chování uživatelů a stávajícím stavem v této oblasti.
2. Prostudujte metody použitelné pro analýzu chování.
3. Navrhněte model pro popis chování uživatele a jeho vhodnou vizualizaci.
4. Navrhněte způsob vzájemného porovnávání modelů, jejich rozdělení na shluky (clustering) a hledání anomálií.
5. Implementujte navržený systém.
6. Ověřte implementaci na reálných datech a diskutujte výsledky.

Literatura:

- Lees, F. (2012). *Lees' Loss prevention in the process industries: Hazard identification, assessment and control*. Butterworth-Heinemann.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 15.
- Liao, T. W. (2005). Clustering of time series data-a survey. *Pattern recognition*, *38*(11), 1857-1874.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Pluskal Jan, Ing.**
Konzultant: Minařík Miloš, Ing., UPSY FIT VUT
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 28. října 2018

Abstrakt

Cieľom tejto práce je vytvoriť aplikáciu, ktorá umožňuje modelovať používateľské chovanie a následne vyhľadávať anomálie v jeho chovaní. Vstupom aplikácie je zoznam akcií, ktoré používateľ vykonal na svojom pracovnom zariadení. Z týchto informácií a udalostí, ktoré na jeho zariadení nastali sa vytvorí model chovania v určitom čase. Následne je tento model porovnávaný v rozdielnych časoch, prípadne s modelmi iných používateľov. Z tohto porovnania môžme získať dodatočné informácie o používateľovi a taktiež môžeme detekovať anomálne chovanie používateľa. Informácie o anomáliách môžu pomôcť pri tvorbe bezpečnostného programu, ktorý sa stará o zamedzenie úniku cenných informácií z prostredia firemnej siete.

Abstract

The aim of this work is to create an application that allows modeling of user behavior and subsequent search for anomalies in this behavior. An application entry is a list of actions the user has executed on his workstation. From this information and from information about the events that occurred on this device the behavioral model for a specific time is created. Subsequently, this model is compared to models in different time periods or with other users' models. From this comparison, we can get additional information about user behavior and also detect anomalous behavior. The information about the anomalies is useful to build security software that prevents valuable data from being stolen (from the corporate enviroment).

Kľúčové slová

Analýza chovania používateľa, Dolovanie dát, Detekcia anomálií, Strojové učenie

Keywords

User Behavior Analysis, Data Mining, Anomaly detection, Machine learning

Citácia

PETROVIČ, Lukáš. *Analýza anomálií v uživatelském chování.* Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal

Analýza anomálií v uživatelském chování

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením Ing. Jana Pluskala. Uviedol som všetky literárne pramene, z ktorých som čerpal.

.....

Lukáš Petrovič

19. mája 2019

Obsah

Kapitola 1

Úvod

Analýza používateľského chovania je jednou z oblastí, ktorá súvisí s produktami *Security Information and Event Management* (*SIEM*). Jednou z hlavných vlastností týchto systémov je pokročilé zaznamenávanie udalostí, ktoré nastali v systéme. Medzi ďalšie úlohy týchto systémov patrí analýza informácií a tvorba hlásení z viacerých dátových zdrojov, akými sú napríklad sieťové prvky alebo operačné systémy [3].

Tieto systémy sú v mnohých prostrediach náročné na správu. Správne implementovaná centralizovaná správa záznamov musí byť opakovane ladená pre zachovanie jej efektívnosti. Staršie *SIEM* systémy vytvárali veľké množstvo záznamov so súvisiacimi kontextuálnymi informáciami. Tieto záznamy boli zasielané bezpečnostným analytikom, ktorý boli zahltený množstvom falošných pozitív a duplikovaných informácií. So stále narastajúcim množstvom dát, ktoré je potrebne spracovávať, bola táto záťaž ďalej neúnosná. Aj z tohto dôvodu je automatická analýza používateľského chovania stále viac spomínaná na trhu bezpečnostného softvéru [17].

Táto práca sa snaží nadviazať na už existujúci vývoj v tejto oblasti. Výsledok práce poskytne prehľad konkrétnych metód pre prevenciu uniku dát a vyhľadávanie anomálií v používateľskom chovaní. Práca zahŕňa vytvorenie aplikácie, ktorá umožňuje modelovať používateľské chovanie a následne vyhľadávať anomálnu aktivitu v tomto chovaní. Aplikácia prijíma zoznam akcií, ktoré reprezentujú používateľovu aktivitu na jeho pracovnom zariadení. Z týchto informácií a udalostí, ktoré na jeho zariadení nastali, sa vytvorí model chovania v určitom čase. Následne je tento model porovnávaný v rozdielnych časoch, prípadne s modelmi iných používateľov. Z tohto porovnania môžme získať dodatočné informácie o používateľovi a taktiež sme schopný detekovať anomálnu aktivitu používateľa. Informácie o anomáliách môžu pomôcť pri tvorbe bezpečnostného programu, ktorý sa stará o zamedzenie úniku cenných informácií z prostredia firemnej siete. Viac informácií o analýze používateľského chovania a o už existujúcich riešeniach v tejto oblasti sa nachádza v kapitole 2.

Kapitola 3 popisuje základné rozdelenie a popis metód pre analýzu používateľského chovania. Približuje výhody a nevýhody rozdielnych prístupov a dôkladnejšie popisuje triedu metód, ktoré je vhodné použiť pre riešený problém. V kapitole sú uvedené výhody a nevýhody jednotlivých metód pre použitie v rámci tejto práce.

Jedným z hlavných prvkov úspešnej dátovej analýzy je dostupnosť kvalitného a rozsiahleho zdroju dát. Štruktúre dostupných dát sa venuje kapitola 4, ktorá taktiež obsahuje návrh detekcie anomálnej aktivity a návrh používateľských a skupinových profilov.

Kapitola 5 popisuje implementáciu navrhnutých časti systému. Záver kapitoly prezentuje vizualizačnú aplikáciu, ktorá spája jednotlivé prvky systému a poskytuje intuitívny nástroj pre analýzu aktivity používateľov.

Záverečná kapitola 6 sa venuje testovaniu jednotlivých častí systému. Testovanie prebieha nad reálnymi dátami a výsledky testovania sú následne vyhodnotené z pohľadu správnosti detekcie anomálií. Kapitola taktiež obsahuje vyhodnotenie výsledkov pri tvorbe skupinových profilov. Záver kapitoly popisuje časovú náročnosť jednotlivých častí vytvorenej aplikácie.

Kapitola 2

Analýza chovania používateľov

Hlavnou úlohou analýzy používateľského chovania je tvorba informovaného pohľadu na akcie, ktoré používateľ vykonáva. Následne je možné z týchto informácií vyvodzovať pravidlá a vzory používateľského chovania. Pochopenie používateľských zámerov je zložitá, no nesmierne hodnotná úloha. Pomáha nám lepšie rozumieť používateľským potrebám a zlepšuje reakciu na zmeny týchto potrieb [26].

Medzi najznámejšie aplikácie patrí analýza používateľov pre potreby cieleného marketingu. Medzi hlavné zdroje informácií patrí analýza toho, na ktorých stránkach sa používateľ pohybuje alebo na ktorý druh reklám kliká najčastejšie. Aj na základe týchto informácií je následne vytvorený profil používateľa, podľa ktorého je pripravená nasledujúca reklama.

Existujú mnohé ďalšie využitia týchto informácií. Využitie, ktoré možno nie je na prvý pohľad úplne zrejmé, je detekcia používateľa s podozrivým chovaním. Jeden z prístupov ako detektovať takéhoto používateľa je správne modelovať chovanie bežných používateľov a následne vyhľadávať nezvyčajne správanie, ktoré môže byť spôsobené útočníkom. Tento prístup je možné uplatniť v mnohých oblastiach, od rozsiahlych sociálnych sieti až po malé systémy. Úspešnosť tohto prístupu spočíva v dostupnosti rozsiahlych dát, ktoré generuje bežný používateľ. Dáta generované škodlivým používateľom sú často veľmi malé v porovnaní s bežným používateľom. Útočník sa naviac snaží využiť slabé miesta systému a jeho činnosť sa často mení a vyvíja [4].

Príklady aplikácií v časti 2.1 približujú využitie analýzy požívateľského chovania. Jednotlivé príklady popisujú rôzne aplikácie modelov. Využívajú rozdielne techniky a riešia problémy, ktoré je možné spojiť s problémami riešenými touto prácou.

2.1 Existujúce aplikácie

Prvý príklad popisuje využitie metód analýzy v prostredí televízie a rádia. Rádio a televízia sú stále vysoko využívane služby a aktuálne kódovanie a prenos dokáže poskytnúť stovky rôznych kanálov. Spoločnosti sa preto zaujímajú o preferencie používateľov a hľadajú charakteristiky záujmu. Používateľov rozdeľujú do rôznych skupín, ku ktorým sa následne správajú rozdielne. Použitým algoritmom v takejto situácií je teda metóda zhlukovania, napríklad algoritmus *k-means* a *faktorová analýza* [26]. Bližšie budú tieto metódy popísané v kapitole 3.

Medzi konkrétne problémy, ktoré sa vyskytujú v tejto oblasti patrí napríklad aj analýza sledovaného obsahu, motivácie, lokalizácie a analýza času, v ktorom daná skupina využíva poskytované služby. Medzi informácie o používateľovi, ktoré sa pri analýze spracovávajú pat-

ria — vekové rozpätie, pohlavie, úroveň vzdelania, príjem a mnoho ďalších [26]. Tabuľka 2.1 približuje niektoré problémy a k nim navrhované algoritmy.

Aplikácia	Model dolovania dát
Analýza straty zákazníka	Clustering algorithm
Odporúčanie TV relácie	Recommendation algorithm
Analýza názoru verejnosti	Time series analysis/Clustering algorithm
Segmentácia používateľov	Kohonen neural network

Tabuľka 2.1: Problémy a súvisiace algoritmy v prostredí telekomunikácií [26].

Druhý príklad približuje analýzu *clickstreams*. Jedná sa o sekvenciu časovo označených udalostí, ktoré sú generované používateľom. Pre webové služby môžu tieto udalosti reprezentovať jednotlive *Hypertext Transfer Protocol* (*HTTP*) požiadavky. V prípade mobilných aplikácií to môžu byť udalosti, ako napríklad kliky tlačidiel, hlasový alebo textový vstup a iné. Automatický zber informácií oproti zastaralým metódam, ktoré napríklad zahrňovali dotazníky, prináša viacero výhod. Hlavná výhoda je v rozsahu informácií, ktoré máme k dispozícií. Naviac následná analýza môže odhaliť nové vzory chovania, ktoré by nemuseli byť viditeľné pri použití predpripravených otázok [24]. Medzi hlavné vlastnosti takéhoto algoritmu patrí:

- škálovateľnosť a správne chovanie na rozsiahlych a zašumených dátach;
- schopnosť zachytiť doteraz neznáme používateľské chovanie;
- intuitívna prezentácia detekovaného chovania.

Tento konkrétny príklad skúma dáta z oblastí sociálnych sieti. Medzi skúmané udalosti patrí začatie konverzácie, zdieľanie príspevku, alebo vytváranie a rušenie spojení medzi používateľmi [24]. Základom je analýza používateľa pomocou metódy *učenie bez učiteľa*. Používatelia sú mapovaný do grafu, kde miesta reprezentujú jednotlivých používateľov a hrany reprezentujú podobnosť ich chovania. Touto metódou vzniknú zhluky používateľov s podobným vzorom chovania. Táto podobnosť, ktorá je reprezentovaná hranami má mnoho vlastností. Vlastnosti, vďaka ktorým boli zhluky vytvorené sú ďalej analyzované a utlmované. Toto utlmenie spôsoby zvýraznenie menej spoločných vlastnosti, čo vytvorí ďalšiu úroveň zhlukovania. Takýmto spôsobom je vytvorené hierarchické zhlukovanie vlastností medzi jednotlivými používateľmi. Je taktiež vhodné vyzdvihnúť kvalitné spracovanie vizualizácie získaných informácií v tejto práci. Práca obsahuje intuitívne zobrazenie jednotlivých hierarchií [24].

Posledným príkladom je modelovanie používateľského chovania v online komunitách. Analýza v tejto oblasti pomáha monitorovať rôzne internetové komunity. Takéto komunity sú miestom kde používatelia najčastejšie komunikujú, pomáhajú si a zdieľajú obsah. Preto je dôležite tieto komunity udržovať. Cieľom práce je správne rozlíšiť normálne a anomálne chovanie používateľa. Na základe vlastností chovania používateľa je navrhnutá metóda pre popis anomálneho správania. Medzi monitorované vlastnosti patria:

- počet používateľov;
- počet príspevkov;
- správanie používateľov;
- zmena správania v závislosti na čase.

Prvou úlohou bolo zachytiť informácie o používateľovi a vytvoriť model používateľskej aktivity. Nasledovala analýza zmeny chovania používateľa vzhľadom k času. Na záver bol model vyhodnotený a boli zachytené dôsledky chovania používateľov na zdravie komunity. Vyhodnotenie prebiehalo pomocou porovnania zachytenej aktivity s charakteristikami danej role. Niektoré vlastnosti, ktoré boli vyhodnocované:

- In-degree Ratio — koncentrácia používateľov odpovedajúcich používateľovi v;
- Posts Replied Ratio — proporcia príspevkov používateľa v, podľa toho, koľko mala odpovedí;
- Thread Initiation Ratio — proporcia novo začatých vlákien používateľom v;
- Average Posts per Thread — priemerný počet príspevkov pre jednotlivé vlákna;
- Standard Deviation of Posts per Thread — štandardná odchýlka v počte príspevkov v jednotlivých vláknach, v ktorých bol používateľ v aktívny.

Na základe týchto vlastností boli vytvorené charakteristiky jednotlivých rolí [1]. Táto práca poskytuje náhľad na možností, ktoré by mohli byť použité aj v rámci tejto práce. Konkrétne ide napríklad o vyhodnotenie role používateľa v určitej komunite.

2.2 Spresnenie zadania a motivácia

Minulá kapitola poskytla náhľad do oblasti analýzy používateľského chovania. Táto sekcia poskytne stručný popis problému, ktorý sa snaží táto práca riešiť. Práca bude poskytovať nástroj pre analýzu používateľského chovania, v rámci firemnej infraštruktúry. Aplikácia sa nezameriava na konkrétnu oblasť a ani na konkrétne dáta. Cieľom aplikácie je spracovať poskytnuté dáta, ktoré opisujú chovanie jednotlivých používateľov. Toto spracovanie zahŕňa nasledujúce akcie:

- spracovanie formátu poskytnutých dát;
- odfiltrovanie a úprava poskytnutých dát;
- tvorba modelu pre popis chovania používateľa;
- tvorba modelu pre popis skupiny používateľov;
- aplikácia vhodných metód pre porovnanie vytvorených modelov;
- vyhľadanie anomálnej aktivity v používateľskom chovaní;
- vizualizácia získaných informácií.

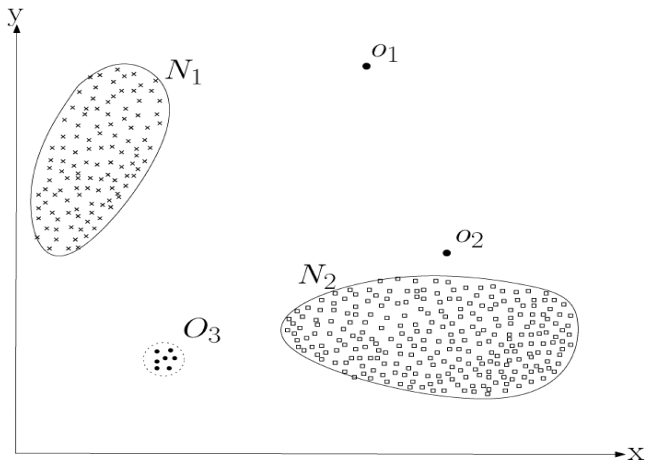
Výsledkom práce je systém, ktorý analyzuje používateľské chovanie jednotlivých používateľov v závislosti na čase vykonaných udalostí. Systém toto chovanie zachytí a prevedie do navrhovaného modelu. Jedným z výstupov budú zachytené intervaly, v ktorých systém vyhodnotil správanie, ktoré sa odlišuje od normálneho správania prekročením určitej hranice. Táto hranica môže byť definovaná na základe toho, či nám záleží na zachytení aj najmenších odlišností v chovaní. Nevýhodou vo veľkej citlivosti na zmenu je vysoká miera falošných pozitív, keďže chovanie používateľa v systéme je často veľmi rôznorodé a postupom času sa vyvíja. Preto je dôležité aplikovať na tieto výsledky ďalšie prostriedky, ktoré sa snažia pochopiť príčinám vzniku týchto anomálií. Ako príklad je možné uviesť mapovanie aktivity, kde na základe predošlého správania systém predikuje podobné správanie aj v budúcnosti. Získané podozrivé časové intervaly musia byť ďalej spracovávané a vyhodnocované, napríklad porovnaním s výsledkami iných používateľov. Používatelia, ktorých správanie bude vykazovať znaky podobnosti, budú označený ako členovia jednej skupiny a môžu tak vytvárať kontext, ktorý vysvetľuje získané rozdiely v správaní. Tým je možné vysvetliť chovanie podozrivého používateľa. Tento spôsob analýzy je bližšie priblížený v časti 5.3.2. V kapitole 4 bude bližšie popísaná štruktúra očakávaných vstupných dát, na základe ktorých prebieha analýza. V nasledujúcej sekcií sa nachádza základný prehľad druhov anomálií a taktiež popis možnosti, ako k týmto anomáliám pristupovať.

2.3 Anomálie a ich detekcia

Detekcia anomálií predstavuje riešenie problému vyhľadávania určitých vzorov v dátach, ktoré nezodpovedajú očakávanému chovaniu. Oblasť detekcie anomálií má široké využitie. Ide napríklad o podvody v oblasti kreditných kariet, poistenia, zdravotníctva, bezpečnosti kritických systémov a mnoho ďalších. Mnohé tieto oblasti kritický závisia na detekcií anomálií, ide napríklad o oblasť zdravotníctva, konkrétne detekcie príznakov choroby.

Obrázok 2.1 zobrazuje anomálie v 2D priestore. Bežným postupom pre vyhľadávanie anomálií je správne definovanie normálneho správania. Ak je toto definovanie presné, môžme pristupovať k dátam mimo tohto správania ako ku anomáliám. Tento postup má ale niekoľko problémov, s ktorými je nutné počítať [4]:

- samotná definícia regiónu, ktorý presne reprezentuje normálne chovanie používateľa je veľmi náročná. V mnohých prípadoch je hranica medzi normálnym a abnormálnym chovaním nejednoznačná;
- ak sú anomálie výsledkom škodlivého útoku, sú často maskované, aby ich algoritmus uvažoval ako bežné chovanie;
- presná definícia anomálie sa mení na základe prostredia, v ktorom je chovanie používateľov analyzované;
- bežné chovanie používateľov sa neustále vyvíja a preto je potrebné na tieto zmeny neustále reagovať;
- dáta často obsahujú šum, ktorého štruktúra sa môže podobať anomáliám. Takýto šum je náročné odhaliť;
- jeden z najdôležitejších problémov je nedostatok označených dát, ktoré sú použité pre trénovacie účely.



Obr. 2.1: Obrázok reprezentuje rozdiely medzi typmi anomálií. Nachádzajú sa tu dva normálne dátové regióny N_1 a N_2. Ich normálnosť plynie z toho, že utvárajú zhluky, ktoré obsahujú väčšinu dát. Body a regióny, ktoré sa nachádzajú mimo týchto regiónov, sú považované za anomálie (body o_1, o_2 a región O_3)[1].

2.3.1 Typy anomálií

Detekcia anomálií sa často zameriava len na jeden konkrétny typ anomálií. Medzi typy anomálií, ktoré môžeme detektovať v analyzovanom systéme patria:

- bodové anomálie;
- kontextové anomálie;
- kolektívne anomálie.

Bodové anomálie

Bodové anomálie sú špecifikované ako jednotlivé body, ktoré uvažujeme ako anomálie k zbytku dát. jedná sa o najjednoduchší typ anomálií a výskum takýchto anomálií je zatiaľ najrozšírenejší. V obrázku 2.1 je tento typ anomálií reprezentovaný bodmi o_1 a o_2. Taktiež body v regióne O_3 sú považované za bodové anomálie, keďže sa líšia od normálneho chovania [4].

Kontextové anomálie

Kontextové anomálie reprezentujú anomálie, ktoré sa objavujú len v prípade prítomnosti určitého kontextu. Každá dátová inštancia je definovaná dvoma typmi atribútov. Prvým sú *kontextové atribúty*, ktoré definujú kontext pre danú inštanciu. Kontext môže byť napríklad časový alebo priestorový [4]. Príkladom časového kontextu môže byť analýza spotrebovanej elektrickej energie. K vysokým hodnotám pristupujeme rozdielne na základe toho, v akej časti dňa sa tieto hodnoty namerali. V prípade priestorových dát môžeme považovať určitú výšku teploty za anomáliu, na základe toho v akej oblasti sa uskutočnilo meranie.

[1] https://dl.acm.org/citation.cfm?id=1541882

Druhým typom atribútov sú *behaviorálne atribúty*. Narozdiel od kontextových, tieto atribúty definujú konkrétne vlastnosti dátovej inštancie [4]. V súvislosti so spomenutými príkladmi z predošlého odstavca môžme tento krát hovoriť o hodnote spotrebovanej energii, alebo hodnote nameranej teploty. Samotná anomália je teda definovaná hodnotami behaviorálnych atribútov v kontexte danom kontextovými atribútmi. To, akú informáciu majú niesť tieto atribúty záleží na konkrétnom prostredí použitia.

Kolektívne anomálie

Posledným typom anomálií, ktoré budú uvedené sú *kolektívne anomálie*. Tieto anomálie sa vyskytujú v určitých skupinách. Hlavnou črtou týchto anomálií je to, že jednotlivé dátové inštancie nemusia byť považované za anomálie. Za anomálie sú považované až ich následné uskupenia [4]. Možným príkladom je sekvencia udalostí, ktoré sú typické (napríklad určitý pracovný postup). V prípade, že sú tieto udalosti vykonané v inom poradí, môžme skupinu takýchto udalosti považovať za anomáliu.

2.3.2 Typy detekcie anomálií

Spôsoby detekcie anomálií sa delia na niekoľko skupín. Jedným z dôvodov tohto rozdelenia je dostupnosť označených dát. Správne označené dáta je možné použiť pri trénovaní modelov. Medzi tri najhlavnejšie skupiny patria:

- Supervised Anomaly Detection;
- Semisupervised Anomaly Detection;
- Unsupervised Anomaly Detection.

Supervised Anomaly Detection je technika pri ktorej sú dostupné správne označené vstupné trénovacie dáta. Pre tvorbu modelu sú potrebne trénovacie dáta z oboch skupín, z normálnych dátových inštancií a aj zo skupiny anomálií. Model je teda schopný priamo rozhodovať, do ktorej skupiny daná inštancia patrí. Tento prístup ale so sebou nesie dve hlavné nevýhody. Prvou je dátová nevyváženosť tried, kedy trieda anomálií obsahuje zlomok dát oproti triede normálneho chovania. Druhým problémom je častá nedostupnosť správne označených trénovacích dát zo skupiny anomálií. Anomálie sa prirodzene môžu vyskytovať tam kde ich nečakáme a preto takéto dáta nie sú dostupné.

Častejšie používanou technikou je technika *Semisupervised Anomaly Detection*, pri ktorej sa predpokladajú trénovacie dáta len pre triedu normálnych dátových inštancií. Model sa vytvorí len pre túto triedu a následne je použitý pre detekciu inštancií, ktoré sú najviac rozdielne od tohto modelu. Vďaka tomu je možné túto metódu použiť v širšom rozsahu problémov [4]. Jedným z problémov môže byť prirodzená zmena správania bežného používateľa. Na túto zmenu musí daný model správne reagovať.

Posledná metóda *Unsupervised Anomaly Detection* vôbec nepracuje s trénovacími dátami. Táto technika predpokladá, že výskyt normálnych inštancií je omnoho častejší ako výskyt anomálií. V prípade, že počet normálnych inštancií výrazne neprevyšuje počet anomálnych inštancií, dochádza k výskytu falošných pozitív [4]. Výstupom týchto techník je skóre, ktoré určuje pravdepodobnosť, že daná dátová inštancia je anomália. Tento prístup je narozdiel od klasickej kategorizácie výhodnejší, pretože jednotlivé dátové inštancie môžme zoradiť podľa ich anomálnej úrovne. Táto práca sa zameriava prevažne na posledne menovaný druh detekcie, kde sú vstupné dáta analyzované bez bližšieho označenia, či sa jedná o normálne chovanie alebo sa jedná o chovanie anomálne.

2.4 Dolovanie dát

Dolovanie dát je proces odhaľovania zaujímavých vzorov a vzťahov v objemných dátach. Táto oblasť spája princípy štatistiky a strojového učenia. Medzi najdôležitejšie úlohy patrí predspracovanie vstupných dát. Toto predspracovanie odhaľuje prítomnosť nadbytočných a nedôležitých dát. Taktiež odstraňuje z týchto dát šum, transformuje a normalizuje vstupné dáta a extrahuje špecifické vlastnosti. Medzi hlavné kroky predspracovania vstupných dát patria:

- vyčistenie dát;
- dátová integrácia;
- transformácia dát;
- redukcia dát;
- diskretizácia dát.

Vyčistenie dát

Vstupné dáta sú často nekompletné, prípadne zašumené. K nekompletným dátam môžme pristupovať viacerými spôsobmi. Buď budeme daný úsek, kde nastala táto chyba, ignorovať alebo zvolíme prístup, kedy sa chýbajúca hodnota nahradí, prípadne dopočíta. Doplňovanie hodnôt ale môže tieto dáta skresliť. Medzi základné techniky pre doplňovanie chýbajúcich dát patrí napríklad použitie spriemerovania. Ďalším spôsobom môže byť využitie učiacich sa algoritmov, akými sú napríklad rozhodovacie stromy [6].

Vo viacerých prípadoch je ďalším faktorom, ktorý skresľuje výsledky analýzy, prítomnosť šumu a prítomnosť anomálnych hodnôt v dátach. Odstránenie anomálnych hodnôt prebieha na základe detekcie výskytu týchto hodnôt a ich následným odstránením. Pre tento účel je možné využiť niektorý zo zhlukovacích algoritmov, ktoré budú viac priblížené v časti 3.2. Pre odstránenie šumu zo vstupných dát je možné použiť regresné algoritmy. Pomocou lineárnej regresie je možné dopočítať hodnoty chýbajúcich vlastností, prípadne odstrániť šum [2] Medzi základné algoritmy v tejto oblasti patria algoritmy lineárnej alebo logistickej regresie.

Integrácia a transformácia dát

Dátová integrácia obsahuje postupy pre získanie dát z databázy. Môže ísť napríklad o synchronizáciu viacerých zdrojov dát. Dátová transformácie popisuje úpravu dát tak, aby boli v správnej forme pre následnú analýzu. Medzi metódy takejto úpravy patrí dátová normalizácia, pri ktorej sa hodnoty vlastnosti modelu transformujú do špecifického rozsahu, prípadne normalizácia na základe strednej hodnoty a štandardnej odchýlky. Do oblasti transformácie ďalej patrí agregácia a generalizácia dát. Pomocou agregácie upravujeme hodnoty z viacerých záznamov a spájame ich do jedného záznamu. Týmto spôsobom sa objem analyzovaných dát výrazne zmenší. Generalizáciou dát môžme nahradiť určitú postupnosť udalostí za jednu udalosť vyššej úrovne [2].

Redukcia a diskretizácia dát

Dátová redukcia má za úlohu znížiť objem analyzovaných dát, no pritom zanechať čo najviac hodnotných informácií pre následnú analýzu. Prvým krokom je redukcia počtu atribútov,

ktoré opisujú analyzovanú udalosť. Jednou z možností je využiť algoritmy redukcie dimenzionality, ide napríklad o algoritmus *Principal component analysis* (*PCA*), ktorého popis sa nachádza v časti 3.6 a je využívaný touto prácou. Ďalšou možnosťou redukcie dát je obmedzenie hodnôt jednotlivých atribútov. Takéto obmedzenie môže byť implementované pomocou zhlukovania hodnôt atribútov do skupín [6]. Ďalšou možnosťou je využitie histogramov, kde sa jednotlivé hodnoty mapujú do intervalov a tým sa redukuje ich rozsah hodnôt. Ako príklad diskretizácie spojitých dát je možné uviesť diskretizáciu na základe hodnoty atribútov, kde sa spojitý priestor hodnôt mapuje do intervalov. Tieto intervaly môžu mať rovnakú šírku, prípadne môžu byť definované na základe rovnakého počtu hodnôt, ktoré do nich spadajú [6].

2.5 Dimenzionalita dát

Pri tvorbe modelu používateľského chovania využívame informácie z viacerých zdrojov. Tieto informácie následne spájame a analyzujeme. Viacrozmerný priestor využívame napríklad v prípade, že popisujeme jednotlivých používateľov viacerými atribútmi. Každý atribút predstavuje jednu dimenziu. Ďalším príkladom môže byť práca s obrazom, kde každý pixel opäť predstavuje oddelenú dimenziu. Dimenzie môžu byť na sebe navzájom čiastočne závislé. V takom prípade môžme redukovať počet dimenzií, pripadne niektoré zanedbať a to v prípade, že nenesú užitočnú informáciu. Pri analýze anomálií sa vstupné dáta mapujú do jedného viacrozmerného priestoru.

Euklidovský priestor patrí medzi najčastejšie používané priestory. Metódy, ktoré sa pre detekciu v tomto priestore používajú sa delia na dve hlavné skupiny. Rozdiel medzi nimi je v uvažovaní pod-priestoroch pri definícií anomálií [27]. Táto práca pracuje s metódami, ktoré tieto pod-priestory neuvažujú.

Modelovanie a detekcia anomálií vo viacrozmerných dátach prináša viacero úskalí. Jedným z nich je koncentrácia vzdialeností. Tento problém je definovaný ako „pomer variancie dĺžky rôzneho bodového vektoru $||X_d||$ s dĺžkou priemerného bodového vektoru $E[||X_d||]$, so zvyšujúcou sa dimenzionalitou dát konverguje k nule“. To ma za následok, že „proporcionálny rozdiel medzi vzdialenosťou najvzdialenejšieho bodu D_{max} a najbližšieho bodu D_{min} vymizne“[27]. Formálna definícia je reprezentovaná rovnicou 2.1.

$$\text{Ak: } \lim_{d\to\infty} var\left(\frac{||X_d||}{E[||X_d||]}\right) = 0$$
$$\text{Potom platí: } \frac{D_{max} - D_{min}}{D_{min}} \to 0 \tag{2.1}$$

To môže predstavovať problém pri získavaní ďalších informácií z analyzovaných dát alebo pri detekcií anomálií. Konkrétne ide o stratu numerického kontrastu hodnôt [27]. Aj vzhľadom k tejto skutočnosti sú v práci využívane algoritmy pre redukciu dimenzionality dát. Bližší popis vstupných dát bude predstavený v časti 4.1.

Kapitola 3

Metódy pre analýzu chovania používateľov

Táto kapitola poskytne stručný prehľad metód, ktoré sa používajú pre analýzu používateľského chovania. Jednotlivé sekcie taktiež poskytujú informácie o vhodnosti použitia daného algoritmu pre potreby tejto práce. Metódy, ktoré táto práca využíva vo svojej implementácií, budú popisované podrobnejšie. Jedná sa hlavne o modely zhlukovania, sekvenčné modely a modely faktorovej analýzy. Tabuľka 3.1 slúži ako prehľad ďalej popisovaných modelov.

Model	Reprezentovaný algoritmom
Classification algorithm model	Decision Tree
Clustering algorithm model	K-means
Association rule model	Association rules algorithm
Sequential pattern mining	Time series analysis
Neural network model	Recurrent Neural Networks and *LSTM*
Factor analysis model	Principal component analysis/Factor analysis

Tabuľka 3.1: Prehľad modelov pre analýzu používateľského chovania [26].

3.1 Klasifikačné modely

V rámci klasifikácie existuje viacero druhov algoritmov, ktoré sú aktuálne využívané. Medzi najznámejšie druhy patria:

- logistická regresia;
- *Support-vector machine* (*SVM*) klasifikátor;
- rozhodovacie stromy;
- neuronové siete;

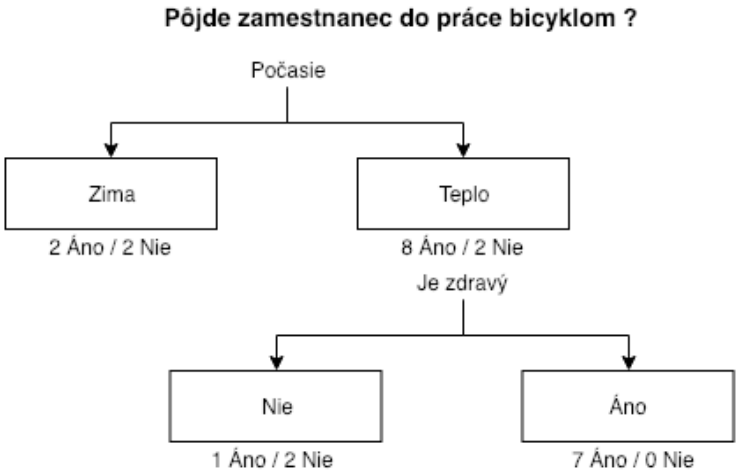
- algoritmy *K-nearest-neighbors* (*KNN*).

V tomto prípade je ako zástupca klasifikačných metód vybraný algoritmus rozhodovacích stromov. Tento algoritmus je použitý pre rozdeľovanie veľkých skupín dát. Každá úroveň stromu reprezentuje rozhodnutie, na základe ktorého sa vstupná skupina dát rozdelí. Každý uzol a list stromu reprezentuje triedu dát. Dáta v rovnakej triede majú určite spoločné vlastnosti.

Cieľom rozhodovacích stromov je dosiahnuť v každom liste stav, ktorý je jasne definovaný a nepotrebuje ďalšie podmienené delenie. Je dôležité určiť správnu mieru delenia tak, aby nedošlo k pretrénovaní. Používaná metóda pre rozhodnutie na akom mieste má dôjsť k deleniu je vylúčenie takých atribútov, ktoré majú čo najmenšiu koreláciu s cieľom [7]. Obrázok 3.1 približuje podobu rozhodovacieho stromu.

Pri stavbe stromu sa vyberie najviac významný atribút, ktorý rozdeľuje vstupné dáta. Tento atribút následne reprezentuje vrchol stromu. Strom je následne konštruovaný obdobne pre ďalšie úrovne. Novo prijaté dáta následne prejdú jednotlivými rozhodujúcimi vrstvami stromu a na jeho listoch im je pridelená trieda.

Tento druh klasifikácie je často používaný pre detekciu škodlivých aktivít v rámci analýzy sieťovej komunikácie. Je schopný spracovať veľké množstvo dát získaných v reálnom čase. V sieťovom prenose je taktiež veľké množstvo binárnych vlastností prenosu, či už ide o prítomnosť určitého protokolu alebo hodnotu konfigurácie. Tieto hodnoty poskytujú priame pravidlo pre daný uzol rozhodovacieho stromu. Hlavnou výhodou narozdiel od iných klasifikačných techník je to, že poskytuje širokú škálu pravidiel, ktoré sú ľahko pochopiteľné a integrovateľné do stávajúcich technológií [13]. Taktiež výsledná klasifikácia je ľahko interpretovateľná, keďže je jednoduché vystopovať dôvody daného prechodu stromu. Ďalšou výhodou algoritmu je nízka miera potreby predspracovania vstupných dát, akou je napríklad normalizácia.



Obr. 3.1: Príklad rozhodovacieho stromu. Každý uzol reprezentuje určitú vstupnú premennú, napríklad informácia o počasí. Jednotlivé listy reprezentujú všetky uvažované hodnoty týchto premenných. Pod listami sa nachádza pomer kladných a záporných odpovedí pri danom ohodnotení vstupných premenných.

Nevýhodami tohto prístupu je náchylnosť na pretrénovanie na trénovacích dátach. Aj vďaka tomu sa odporúča využitie redukcie dimenzionality pomocou algoritmu *PCA*. Tým

efektívne znížime počet redundantných vlastností, na základe ktorých sa buduje rozhodovací strom. Druhou nevýhodou je náchylnosť na zaujatosť klasifikácie jednej triedy v prípade, kedy pomer dostupných trénovacích dát pre jednotlivé triedy je nevyvážený [19]. Takýto nepomer je pri analýze anomálií v používateľskom chovaní veľmi častý.

3.2 Modely zhlukovania

Medzi najpoužívanejšie metódy zhlukovania patrí algoritmus *K-means*. Tento algoritmus patrí do skupiny algoritmov, ktoré pracujú metódou učenia bez učiteľa, teda pracujú nad neoznačenými dátami. Cieľom tohto algoritmu je nájsť zhluky vo vstupných dátach. Dátové inštancie sú zoskupované na základe podobnosti ich vybraných vlastností. Na obrázku 3.2 je možné vidieť zhlukovanie na základe vzdialenosti parametrov x_1 a x_2. Počet zhlukov definujeme vstupným parametrom. Tento parameter sa môže meniť v závislosti na vstupných dátach.

Algoritmus pozostáva z troch hlavných krokov.

1. Inicializácia nastavením K centrálnych bodov.
2. Rozdelením vstupnej množiny dát do K podmnožín, na základe vzdialenosti od centrálnych bodov.
3. Aktualizovaním pozície centrálnych bodov.

Iterovaním medzi druhým a tretím krokom dôjdeme k záveru, že algoritmus po určitom počte iterácií konverguje. Pre to, aby sme boli schopný správne posúdiť vzdialenosť medzi dvoma bodmi, potrebujeme správne definovať metriku vzdialenosti. Vzdialenosť je definovaná ako hodnota funkcie, ktorá prijíma dva objekty a vracia pozitívnu hodnotu vyjadrujúcu vzdialenosť. Formálne sa teda jedná o funkciu *Dist* s pozitívnymi reálnymi hodnotami, ktorá je definovaná nad kartézskym súčinom $X \times X$ nad množinou X [22]. Pre každé x, y, z z množiny X teda platí:

- axióm identity 3.1;

$$Dist(x, y) = 0 \Leftrightarrow x = y \tag{3.1}$$

- axióm trojuholníkovej nerovnosti 3.2;

$$Dist(x, y) + Dist(y, z) \geq Dist(x, z) \tag{3.2}$$

- axióm symetrie 3.3.

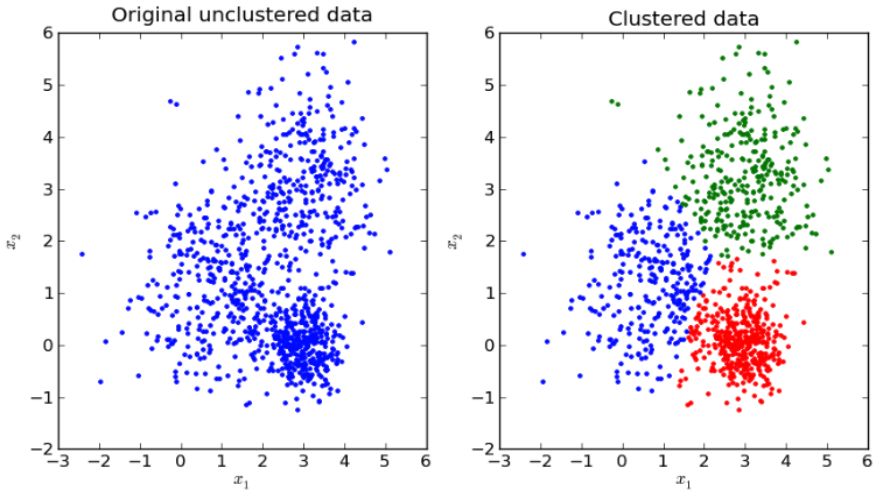
$$Dist(x, y) = Dist(y, x) \tag{3.3}$$

Existuje viacero prístupov pre vyjadrenie vzdialenosti medzi dvoma objektami. Následne budú uvedené niektoré najpoužívanejšie vyjadrenia vzdialenosti.

Euklidovská vzdialenosť

Vzdialenosť sa vypočíta na základe rozdielu druhých mocnín medzi koordinátami dvoch objektov [22]. Tento výpočet popisuje rovnica 3.4.

$$Dist_{XY} = \sqrt{\sum_{k=1}^{m} (X_{ik} - X_{jk})^2} \tag{3.4}$$



Obr. 3.2: Príklad použitia algoritmu *K-means* na neoznačených dátach. Po aplikácií algoritmu s nastaveným počtom ustúpení na tri, je možné vidieť súmerne rozdelenie vstupných dát na zhluky[1].

Manhattanovská vzdialenosť

Výpočet vzdialenosti prebieha na základe absolútneho rozdielu medzi koordinátami porovnávaných objektov [22]. Výpočet tejto vzdialenosti je reprezentovaný rovnicou 3.5.

$$Dist_{XY} = |X_{ik} - X_{jk}| \tag{3.5}$$

Minkowského vzdialenosť

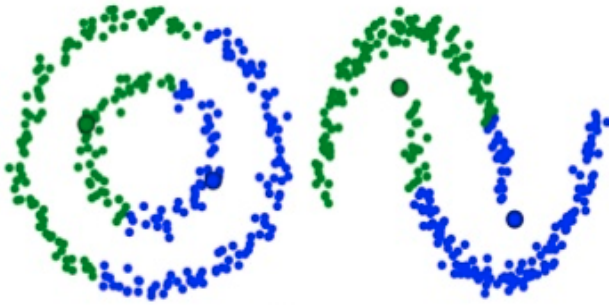
Vzdialenosť je generalizáciou definovanej metrickej vzdialenosti, ktorej výpočet je reprezentovaný rovnicou 3.6. V prípade p=2, táto vzdialenosť vyjadruje Euklidovskú vzdialenosť [22].

$$Dist_{XY} = \left(\sum_{k=1}^{d} |X_{ik} - X_{jk}|^{1/p} \right)^{p} \tag{3.6}$$

V našom prípade bude algoritmus *k-means* využitý pre potreby zhlukovania používateľov s rovnakým chovaním. Bližšie sa tomuto využitiu venuje časť 4.3, ktorá približuje tvorbu skupinového profilu.

Na obrázku 3.3 je možné vidieť problém pri zhlukovaní dát, ktoré nevytvárajú zhluk okolo jediného bodu. Pre riešenie aj takýchto zhlukov je možné modelovať dáta pomocou *Gaussian Mixture Models*, teda modelovanie pomocou skupiny niekoľkých funkcií, ktoré sú definované Gaussovov krivkou. V tomto prípade máme teda dva parametre, ktoré reprezentujú jednotlivé zhluky, je to stredový bod funkcie a jej štandardná odchýlka. Pre nájdenie týchto parametrov pre jednotlivé zhluky je možné využiť algoritmus *Expectation–Maximization* [20]. Nasledujúci výpočet popisuje kroky tohto algoritmu.

[1] https://mubaris.com/posts/kmeans-clustering/



Obr. 3.3: Ukážka typu vstupných dát, pri ktorých zhlukovanie pomocou algoritmu *K-means* zlyhalo. Dôvodom je, že tento algoritmus zhlukuje dáta, ktoré sa nachádzajú okolo určitého bodu a majú teda kruhovitú štruktúru[2].

1. Na začiatku sa vyberie niekoľko zhlukov, náhodne rozmiestnených na vstupných dátach. Parametre modelu pre každý zhluk sa taktiež určia náhodne, prípadne môžeme odhad zlepšiť tým, že vstupné dáta predom analyzujeme.

2. Na základe získaných modelov, algoritmus vypočíta aká je pravdepodobnosť, že jednotlivé dátové inštancie patria do daného modelu. Čím bližšie k stredu sa dátová inštancia nachádza, tým väčšia je pravdepodobnosť, že patrí do daného modelu.

3. Vypočítajú sa nové hodnoty parametrov pre každý model, ktorý reprezentuje zhluk dát. Nove hodnoty parametrov počítame ako váhovaný priemer pozícií, kde váha je reprezentovaná pravdepodobnosťou, že daná dátová inštancia patrí do modelu.

4. Algoritmus následne pracuje iteratívne podľa kroku 2 a 3, pokiaľ výpočet nových hodnôt parametrov nekonverguje.

Použitie tohto algoritmu je viac flexibilné na tvar zhlukov vo vstupných dátach. Je možné výrazne lepšie modelovať dáta, ktoré sú v tvare elíps [20].

3.3 Modely asociačných pravidiel

Získavanie asociačných pravidiel je rozšírený postup pri detekcií anomálií. Pomocou tohto postupu je možné získať dôležité vzťahy medzi veľkou množinou dátových objektov. Medzi najznámejšie príklady použitia patrí trhová analýza, ktorá hľadá zákonitosti správania zákazníkov v supermarketoch. Tieto zákonitosti následne ukazujú pravdepodobnosť, ktorý produkt si zákazník pravdepodobne kúpi na základe toho čo už ma kúpené [8].

Získané informácie z dát pomocou asociačných pravidiel môžu byť definované ako vzťah takých udalostí, ktoré sa často vyskytujú v analyzovaných dátach spoločne. Apriori Algorithm je jedným z algoritmov, ktorý sa často používa pre nájdenie takýchto pravidiel. Medzi hlavné komponenty algoritmu, ktoré budú následne v texte detailnejšie popísané, patria:

- podpora (support);
- spoľahlivosť (confidence);

[2] https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

- zdvih (lift).

V nasledujúcich vysvetleniach termínov budú jednotlivé odôvodnenia odkazovať na príklad využitia asociačných pravidiel z prostredia spomínaného príkladu správania sa zákazníkov v supermarkete.

Podpora

Hodnota reprezentujúca východziu popularitu daného objektu. Je vypočítaná na základe pomeru počtu transakcií, v ktorých sa daný objekt nachádza k celkovému počtu transakcií [10]. Predpokladajme, že chceme definovať túto hodnotu pre objekt B, platí teda rovnica 3.7.

$$\text{Podpora}(B) = \frac{\text{Počet transakcií obsahujúcich(B)}}{\text{Celkový počet transakcií}} \tag{3.7}$$

Spoľahlivosť

Hodnota odkazujúca sa na pravdepodobnosť, že objekt B je kúpený ak je kúpený aj objekt A. Túto hodnotu je možné vypočítať na základe pomeru počtu transakcií, v ktorých položky A aj B sú kúpené spoločne k počtu transakcií, v ktorých bola kúpená položka A [10]. Rovnica 3.8 presne popisuje túto definíciu.

$$\text{Spoľahlivosť}(A \rightarrow B) = \frac{\text{Počet transakcií obsahujúcich(B aj A)}}{\text{Počet transakcií obsahujúcich(A)}} \tag{3.8}$$

Zdvih

Táto hodnota popisuje koľko násobne sa zvýši predaj objektu B v prípade, že je predaný objekt A. Výsledok predstavuje pomer medzi dvoma predošlými parametrami a výpočet reprezentuje rovnica 3.9.

$$\text{Zdvih}(A \rightarrow B) = \frac{\text{Spoľahlivosť}(A \rightarrow B)}{\text{Podpora}(B)} \tag{3.9}$$

Samotný algoritmus je rozdelený na dva hlavné kroky. Prvým je generácia takých množín objektov, ktoré sa vyskytujú v transakciách spolu najčastejšie. Na začiatku sa vyberú všetky jednoprvkové podmnožiny a zahodia sa tie, ktoré majú nižšiu hodnotu *Podpora* ako je minimálna stanovená hranica. Z prvkov, ktoré ostali sú následne vybrané všetky dvojprvkové podmnožiny a opäť sa vyhodia prvky, ktoré nemajú hodnotu *Podpora* vyššiu ako stanovená hranica. Takto algoritmus pokračuje až nie je možné ďalej vyhodnocovať väčšie podmnožiny prvkov. Týmto postupom získame všetky podmnožiny vstupnej množiny objektov, ktoré sa často vyskytujú v transakciách.

V druhom kroku sa generujú asociačné pravidlá pomocou vypočítania hodnoty *Spoľahlivosť* pre každé pravidlo. Ak napríklad z predošlého kroku algoritmu vieme, že objekt A,B a C sa v transakciách často vyskytujú spolu, môžeme získať niekoľko pravidiel, ktoré popisujú vzťahy medzi týmito objektami. Rovnica 3.10 reprezentuje príklad vytvorenia takéhoto pravidla, teda pravdepodobnosť, že objekt C bude kúpený ak sú kúpené objekty A a B. V poslednom kroku sa vytvorené pravidlá zoradia na základe hodnoty *Zdvih* [10].

$$\text{Spoľahlivosť}(A \wedge B \rightarrow C) = \frac{\text{Podpora}(A \wedge B \wedge C)}{\text{Podpora}(A \wedge B)} \tag{3.10}$$

Modely asociačných pravidiel môžu byť užitočné pri budovaní, či už skupinových alebo individuálnych profilov používateľov. Pri budovaní individuálneho profilu môžeme takto získať užitočné vzory chovania, ktoré vedú k identifikácií záujmov používateľa. Môže sa napríklad jednať o analýzu skupiny udalostí, ktoré používateľ v určitú dobu vykonal. Jednou z oblastí využitia je oblasť odporúčacích algoritmov, kde identifikácia častých zoskupení vyhľadávaných objektov môže viesť k najpopulárnejším skupinám produktov, vyhľadávaných určitou skupinou ľudí [15].

V kontexte tejto práce môže byť tento model použitý pre definovanie skupinových profilov. Ako príklad, je možné vytvoriť pravidlá, popisujúce ktoré skupiny aplikácií sú najčastejšie využívané jednotlivými používateľmi. Tým môžeme získať informáciu o vzťahu medzi týmito používateľmi.

3.4 Sekvenčné modely

Jedná sa o analýzu dát, v sekvenčnej forme. Jednotlivé dáta sú teda na sebe závislé a v našom prípade je touto závislosťou čas. Hodnoty, ktoré sa objavujú pri analýze používateľského chovania sú v diskrétnej forme. Na základe týchto vlastností sú pre reprezentovanie tejto triedy použité algoritmy, ktoré analyzujú časové rady.

Medzi dve hlavné úlohy pri analýze časových radov patrí:

- identifikácia povahy javu, ktorý je reprezentovaný sekvenciou pozorovaní;
- predpoveď budúcich hodnôt časovej rady.

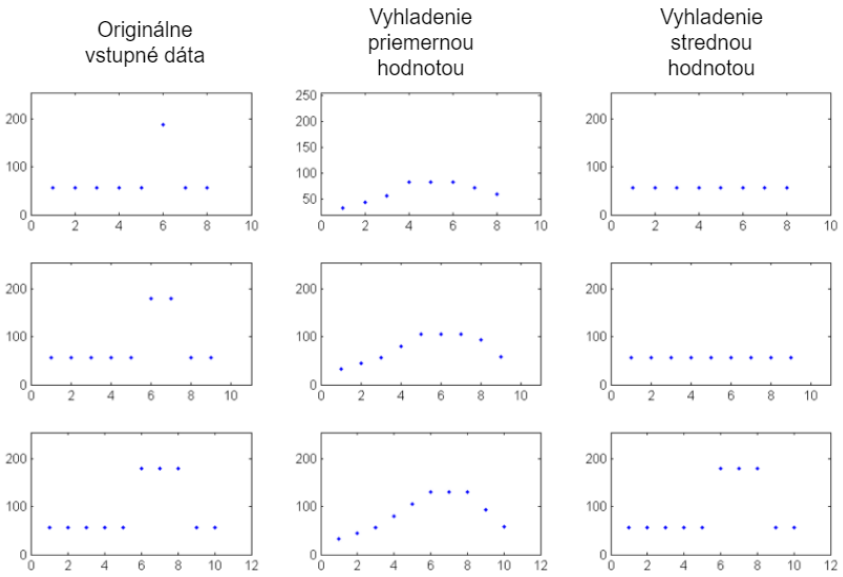
Oba ciele vyžadujú identifikáciu, prípadne formálny popis vzoru, ktorý reprezentuje časovú radu. Akonáhle získame vzor, na základe ktorého sú dáta vytvorené, môžme tento vzor využiť pre predpoveď budúceho vývoja časovej rady.

Väčšinu časových rád je možné popísať pomocou dvoch hlavných komponent. Prvou z nich je trend a reprezentuje základnú lineárnu alebo nelineárnu komponentu. Táto komponenta sa s postupom času mení a nemá periodický charakter. Druhá komponenta reprezentuje sezónnosť a má periodický charakter. Časové rady zložené z týchto dvoch komponent majú častý výskyt v reálnych dátach [23].

Základnými technikami pre analýzu trendu sú metódy vyhladzovania. Jedna z nich je metóda pohyblivého priemeru, ktorá nahradzuje hodnotu každého prvku časovej rady priemernou hodnotou okolitých prvkov. Počet prvkov, na základe ktorých sa počíta táto nová hodnota, sa nazýva veľkosť okna. V niektorých prípadoch môže byť namiesto funkcie priemeru použitá funkcia strednej hodnoty. Hlavná výhoda tohto prístupu je to, že výstupná časová rada je menej náchylná na odľahlé body. Nevýhoda nastáva v prípade väčšieho množstva odľahlých bodov v rámci okna. V takomto prípade vyhladzovanie zlyhá a výstup nie je vyhladený [23]. Tento stav je možné vidieť na obrázku 3.4.

Sezónnosť môže byť definovaná ako korelačná závislosť stupňa k medzi každým i a i-k prvkom časovej rady. Výpočet je realizovaný autokorelačnou funkciou, kde k reprezentuje opozdenie. Sezónnosť sa teda v časovej rade opakuje každých k prvkov. Rovnica 3.11 popisuje výpočet autokorelačnej funkcie pri posune k prvkov.

$$ACF(k) = \sum_{n=k}^{N-k} s(n)s(n-k) \tag{3.11}$$



Obr. 3.4: Obrázok reprezentuje porovnanie aplikácie filtru pracujúceho na základe výpočtu priemernej a strednej hodnoty. Z obrázku vyplávajú výhody a nevýhody filtrovania v prípadoch rôzneho počtu odľahlých bodov[3].

V dátach z reálneho sveta ale nie je jednoduché odhaliť matematický model, generujúci danú časovú radu. Pre popis skrytých vzorov a pre ďalšiu predikciu časovej rady bolo vyvinutých viacero algoritmov. Medzi najznámejšie patrí metóda *Autoregressive integrated moving average* (*ARIMA*) a jej modifikácie, alebo motóda *Seasonal Trend decomposition* (*STL*).

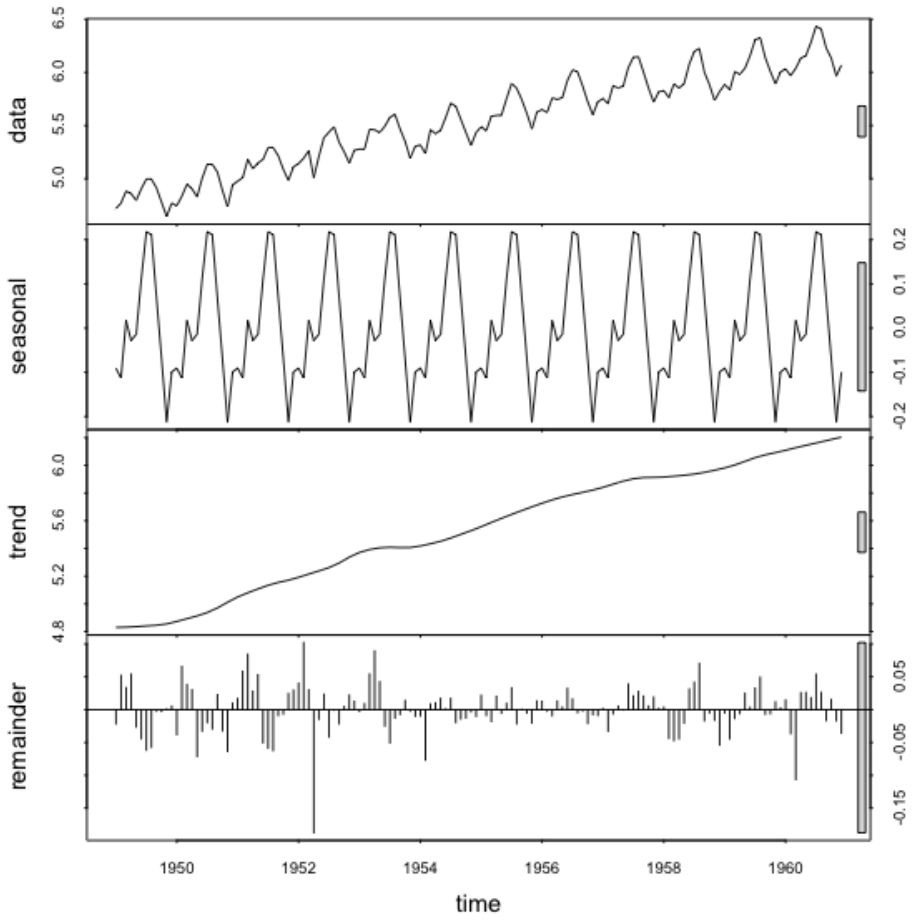
Pomocou *ARIMA* metodológie môžeme odhaľovať skryté vzory v analyzovaných dátach a umožňuje nám predpovedať ďalší vývoj. Technika sa skladá z dvoch hlavných procesov. Prvým je autoregresívny proces, ktorý popisuje závislosť prvku časovej rady na predchádzajúcich prvkoch. Každý prvok časovej rady je teda popísaný náhodnou chybou a lineárnou kombináciou predošlých prvkov. Pre správne vyhodnotenie tohto procesu musí byť dodržaná podmienka stacionarity [23]. Pre jej splnenie musí časová rada spĺňať tieto podmienky:

- hodnoty časovej rady majú konštantný priemer;
- hodnoty časovej rady majú konštantnú odchýlku, alebo strednú hodnotu;
- autokovariačné koeficienty časovej rady nezávisia na čase.

Druhým procesom je proces pohybujúceho sa priemeru. Nezávisle na autoregresívnom procese, každý prvok časovej rady môže byť ovplyvnený chybou v minulosti. Každý prvok časovej rady je teda popísaný náhodnou chybou a lineárnou kombináciou predošlých chýb [23].

Pre analýzu časových rád bola v tejto práci zvolená varianta techniky *STL*. Dáta nad ktorými bude algoritmus pre vyhľadávanie anomálneho správania nespĺňajú podmienku stacionarity. *STL* technika túto podmienku pre svoje správne fungovanie nevyžaduje. Ďalšia

[3] https://www.fit.vutbr.cz/study/courses/ZPO/private/lectures/zpo_sum_v_obraze.pdf



Obr. 3.5: Obrázok predstavuje výslednú dekompozíciu časovej rady po aplikovaní *STL*. Vstupná časová rada je dekomponovaná na sezónnu časť, ktorá sa opakuje v pravidelných intervaloch a na časť reprezentujúcu celkový trend. Posledná časť reprezentuje zbytok po dekompozícií, ktorý je následne možné použiť pre detekciu anomálií[4].

nevýhoda metódy *ARIMA* a jej rozšírení je jej slabá škálovateľnosť na rozsiahlych časových radách. Tieto algoritmy sú využívané hlavne pre prácu s mesačným, prípade niekoľko mesačným intervalom [25].

Základnou myšlienkou pre použitie tejto techniky pre vyhľadávanie anomálií v dátach reprezentujúcich používateľské chovanie je rozdelenie časovej rady na komponentu opisujúcu trend a sezónnosť. Chybu, ktorá reprezentuje rozdiel medzi predpovedanou a skutočnou časovou radou, môžeme reprezentovať ako anomáliu a to v prípade, kedy táto chyba presiahne určitý prah. Obrázok 3.5 približuje spracovanie časovej rady. Zvolená varianta *STL* techniky má názov *A Robust Seasonal-Trend Decomposition Algorithm for Long Time Series* (*RobustSTL*) [25] a rieši niektoré nedostatky základnej metódy, akou je napríklad schopnosť spracovávať časové rady so sezónnosťou obsahujúcou väčší počet hodnôt. Táto vlastnosť je v

[4] https://medium.com/wwblog/anomaly-detection-using-stl-76099c9fd5a7

kontexte tejto práce veľmi dôležitá, keďže jednotlivé dni obsahujú vysoký počet udalostí. *RobustSTL* je taktiež viac odolný proti šumu a časovým posunom v sezónnosti. Komponenta sezónnosti, nameraná v čase 1:00 môže korešpondovať s časom 1:30 v predošlom dni. Taktiež adaptácia na zmenu sezónnosti v čase je jednou z hlavných výhod tohto algoritmu. Väčšina algoritmov je taktiež náchylná na neočakávanú zmenu trendu a zvyšku po predikcií. Tieto zmeny sú ale hlavnou súčasťou pri analýzy časových rád pre detekciu anomálií [25]. Samotný algoritmus môžeme rozdeliť na štyri hlavné časti:

- odstránenie šumu z časovej rady použitím bilaterálneho filtru;
- extrahovanie trendu pomocou *Least absolute deviations* (*LAD*) regresie;
- výpočet sezónnosti aplikovaním nelokálneho sezónneho filtrovania;
- nastavenie extrahovaných komponentov;

Rovnica 3.12 popisuje dekompozíciu jedného bodu časovej rady, kde t reprezentuje trend, s reprezentuje sezónnosť a r reprezentuje komponentu, ktorá nie je modelovaná trendom ani sezónnosťou.

$$y_i = t_i + s_i + r_i, i = 1, 2, ...N \tag{3.12}$$

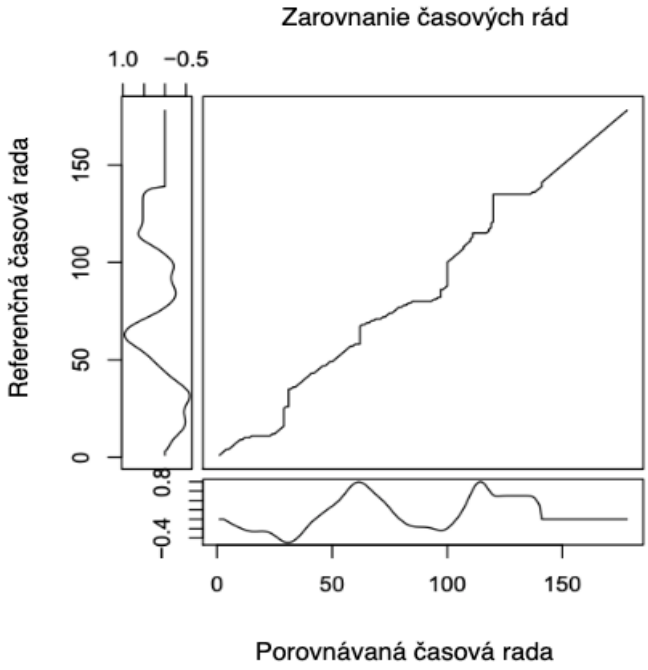
Analýza pomocou popisovaného algoritmu má dobré využitie pri analýze časových rád, ktoré popisujú používateľské chovanie. Pomocou tohto algoritmu môžeme sledovať anomálne správanie používateľa. Toto správanie sa odvíja od toho, ako sa správal v minulosti a je ho možné aplikovať napríklad na časovú os reprezentujúcu jeho aktivitu na pracovnom zariadení. Časť 5.3.2 sa bližšie venuje aplikácií tohto algoritmu.

Okrem analýzy časových rád je druhou využívanou operáciou porovnávanie jednotlivých časových rád. Jedným z používaných algoritmov je algoritmus *Dynamic time warping distance*, ktorý porovnáva dve časové rady. Porovnávané časové rady môžu byť voči sebe posunuté, prípadne môžu obsahovať rovnaké udalosti z rozdielnym odstupom. Algoritmus hľadá optimálny spôsob ako deformovať časovú os tak, aby našiel rovnaké výskyty udalosti v jednotlivých časových radách [18].

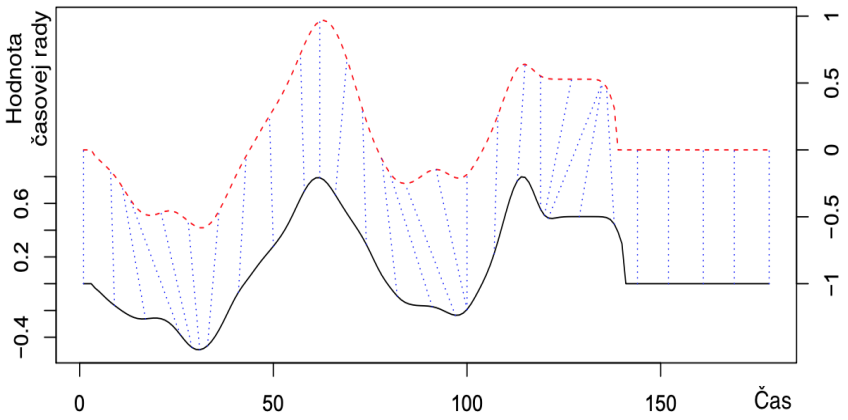
Samotný algoritmus sa skladá z dvoch hlavných krokov. V prvom kroku algoritmu sa vytvorí *Local cost matrix* (*LCM*) matica, ktorá má rozmery $N \times M$, kde N reprezentuje počet prvkov referenčnej časovej rady a kde M značí počet prvkov porovnávanej časovej rady. Takáto matica je osobitne vytvorená pre každú porovnávanú vzdialenosť medzi bodmi časových os. Obrázok 3.6 približuje tvorbu tejto matice. Predpokladajme, že x a y reprezentujú časové rady, potom pre každý element i,j matice LCM je vypočítaná norma medzi prvkami x_i a y_j. Tento výpočet je reprezentovaný v rovnicou 3.13, v ktorej p súvisí s normou l_p, ktorá bola použitá pre výpočet LCM. Index v slúži ako index časovej rady v prípade výpočtu s viacrozmernými časovými radami [18].

$$LCM(i,j) = \left(\sum_v |x_i^v - y_j^v|^p \right)^{1/p} \tag{3.13}$$

V druhom kroku sa pracuje so získanou maticou, ktorá popisuje optimálnu cestu pre porovnanie časových rád. Algoritmus v nej hľadá cestu, ktorá minimalizuje zarovnanie medzi danými radami. Na začiatku vychádza z počiatočnej pozície oboch rád *LCM(1,1)* a iteratíve postupuje k ich koncu *LCM(N,M)*. Pri postupe si vyberá cestu na základe najnižšej



Obr. 3.6: Vizuálna reprezentácia algoritmu pre nájdenie optimálnej cesty medzi dvoma časovými radami. Na ľavej strane, pozdĺž vertikálnej osy je naznačená referenčná časová rada. Na spodnej strane, pozdĺž horizontálnej osy je naznačená časová rada, ktorú porovnávame s radou referenčnou. Výsledná matica reprezentuje špecifickú LCM, získanú na základe porovnávaných časových rád[5].



Obr. 3.7: Mapovanie medzi dvoma časovými radami pomocou algoritmu *Dynamic time warping distance*. Modrá čiara ukazuje ako sú jednotlivé body prvej časovej rady mapované na druhú časovú radu[6].

[5] https://www.semanticscholar.org/paper/Comparing-Time-Series-Clustering-Algorithms-in-R-Sarda-Espinosa/ceabb44c8b3606decd791ae7da50e54401a0e9f5

ceny [18]. Týmto spôsobom určí dynamické zarovnanie porovnávaných rád, ktoré je možné vidieť na obrázku 3.7. V kontexte tejto práce je táto metóda použiteľná napríklad pri porovnávaní časových rád medzi jednotlivými používateľmi.

3.5 Modely neurónových sietí

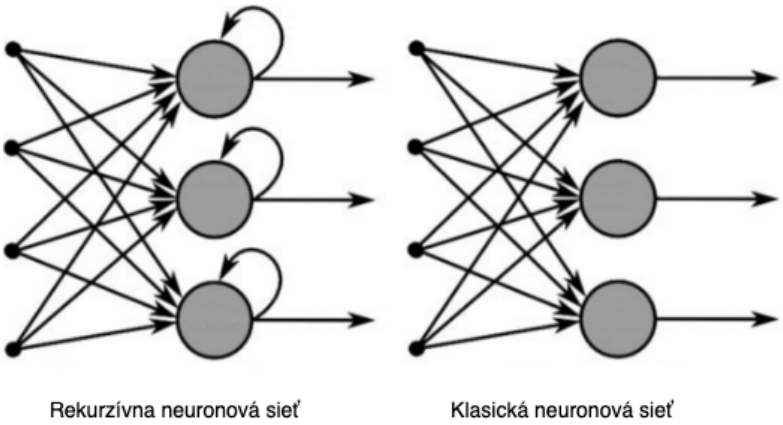
Vývoj v oblasti neurónových sieti získal v poslednom období veľkú pozornosť a obsahuje veľké množstvo metód, ktoré je možné využiť pre analýzu chovania používateľa. Rekurentné neuronové siete sú ďalším vývojovým stupňom v tejto oblasti. Základnou myšlienkou je využitie určitého typu spätnej väzby, ktorá do vstupu zakomponuje aj výsledok z predošlého kroku. Tento prístup rozširuje silu klasických neurónových sieti a umožňuje použitie na určitej sekvencií dát. Ako príklad je možné uviesť analýzu videa, kedy je možné využiť predošlé snímky pri analýze aktuálneho snímku [12]. Na obrázku 3.8 je znázornená popisovaná rekurzia. Z tohto modelu vyplýva, že takáto štruktúra neurónovej siete je schopná na vniesť do výstupu informáciu s niekoľkých predošlých výsledkov. Takáto vlastnosť pripomína určitý druh krátkodobej pamäte. Pri analýze reálnych sekvenčných dát je ale často krát potrebné využívať dlhodobú pamäť. Z tohoto dôvodu je tento algoritmus rozšírený o *Long Short Term Memory* (*LSTM*), teda o možnosť uloženia si určitej informácie do pamäte.

LSTM siete, majú rovnako ako rekurentné siete reťazovú štruktúru, kedy jednotlivé prechody sieťou obsahujú určitú závislosť. Táto štruktúra je znázornená zelenými blokmi na obrázku 3.9. Na obrázku vidíme určitú časť neurónovej siete. Sieť prijíma sekvenciu vstupu, ktorý je označený X_t. Môže sa napríklad jednať o spomínaný snímok videa. *LSTM* sieť obsahuje štyri vrstvy, kde každá má špecifickú úlohu a sú znázornené žltým obdĺžnikom. Prvá vrstva má za úlohu rozhodnúť, aká informácia bude vyhodená zo stavu uzlu. Ďalšie dve vrstvy rozhodujú o tom, ktorá informácia na vstupe bude uložená. Posledná vrstva rozhoduje o tom, aká informácia sa objaví na výstupe bunky [12]. Tento výstup je znovu zavedený na vstup pri ďalšom prechode sieťou. Tento hrubý popis siete prezentuje jednu z mnohých variant, ktoré je možné vytvoriť.

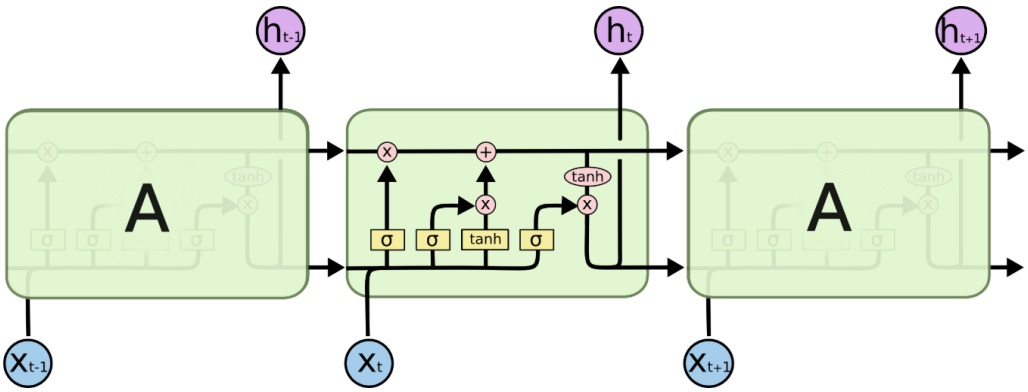
V kontexte detekcie anomálií v používateľských dátach je možné použiť viacvrstvovú *LSTM* sieť pre modelovanie dát, so štruktúrou časových rád. Konkrétne sa jedná o metódu učenia sa bez učiteľa, kedy dostupné dáta reprezentujú prevažne normálne chovanie. Po namodelovaní môžeme naučenú sieť použiť pri predikcií budúceho chovania. Rozdiel medzi predikovaným a reálnym chovaním systému následne indikuje anomálne správanie. Tento postup je možné použiť bez špecificky definovanej veľkosti okna a bez špecifického predspracovania vstupných dát [9].

V rámci detekcie anomálií boli v tejto práci využité sekvenčné algoritmy pre analýzu časových rád. Jedným z dôvodov je to, že výsledky týchto algoritmov sú v kontexte tejto práce lepšie interpretovateľné, než výsledky pri použití modelov neurónových sieti. Pre budúci vývoj práce je ale vhodné problém interpretácie ďalej riešiť a následne využiť výhody *LSTM* siete.

[6] https://www.semanticscholar.org/paper/Comparing-Time-Series-Clustering-Algorithms-in-R-Sarda-Espinosa/ceabb44c8b3606decd791ae7da50e54401a0e9f5



Obr. 3.8: Obrázok zobrazuje rozdiel medzi vnútornou štruktúrou rekurzívnej a klasickej neurónovej siete. Na obrázku je vidieť znázornenie spätnej väzby, ktorá reprezentuje funkciu krátkodobej pamäte[7].



Obr. 3.9: Štruktúra rozbalenej rekurzívnej slučky medzi jednotlivými prechodmi rekurzívnej neurónovej siete. Obrázok konkrétne zobrazuje *LSTM* sieť, obsahujúcu štyri vrstvy, znázornené žltými obdĺžnikmi. Ich úlohou je spravovať vnútorný stav bunky, pre dosiahnutie funkcie dlhodobej pamäte[8].

3.6 Modely faktorovej analýzy

Technika z oblasti faktorovej analýzy, ktorú táto práca využíva sa nazýva *Principal component analysis* (*PCA*). Je to jedna z najpoužívanejších techník v oblasti strojového učenia, pretože nám umožňuje odhaľovať skryté štruktúry vo vstupných dátach. Táto technika bude popísaná podrobnejšie než predošle a to z dôvodu, že je využívaná pri konkrétnej implementácií riešenia pre analýzu používateľského chovania. Jedná sa o bezparametrickú metódu pre extrakciu informácií z viacrozmerných dát. To je dosiahnuté redukciou dimenzionality dát. Táto redukcia prebieha na základe nájdenia najviac zmysluplnej bázy vo vstupných dátach. Inými slovami, pomocou *PCA* metódy hľadáme rozdielnu bázu, ktorá je lineárnou

[7] https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5
[8] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

kombináciou originálnej bázy a lepšie popisuje analyzované dáta [21]. Vlastnosťami, ktoré sú v dátach prítomné a ktoré spôsobujú ukrývanie zmysluplných dát sú:

- šum;
- rotácia;
- redundancia.

V prípade dvoj-dimenzionálnych dát, je identifikácia redundancie jednoduchá. Ako príklad je uvedená situácia na obrázku 3.10, kde vidíme tri prípady rozloženia dát. Na ľavej strane vidíme prípad kedy sú pre reprezentáciu jednotlivých dát potrebné obe dimenzie. Na pravej strane je zobrazený opačný prípad, v ktorom je jedna dimenzia nadbytočná a neposkytuje nám významnú informáciu. V dvoj-dimenzionálnom priestore môžeme takúto redundanciu detekovať pomocou úsečky, ktorú preložíme množinou bodov a vyhodnotíme ako správne daná úsečka leží na týchto bodoch. Pre generalizáciu tohto postupu v prípade viacrozmerných dát, využijeme kovariančnú maticu [21].

Kovariancia popisuje stupeň lineárneho vzťahu medzi dvoma premennými. Vysoká hodnota indikuje vysokú redundanciu. Takúto vysokú hodnotu môžeme pozorovať na obrázku 3.10c. Predpokladajme dva riadkové vektory a a b definované rovnicou 3.14.

$$\begin{aligned} a &= [a_1 a_2 ... a_n] \\ b &= [b_1 b_2 ... b_n] \end{aligned} \tag{3.14}$$

Pomocou týchto vektorov môžeme následne vypočítať kovarianciu. Tento výpočet je reprezentovaný rovnicou 3.15.

$$\sigma^2_{ab} \equiv \frac{1}{n-1} ab^T \tag{3.15}$$

Túto definíciu pre dva vektory môžeme následne rozšíriť pre ľubovolne množstvo vektorov. Predpokladajme maticu X, ktorá obsahuje m riadkových vektorov a je definovaná rovnicou 3.16.

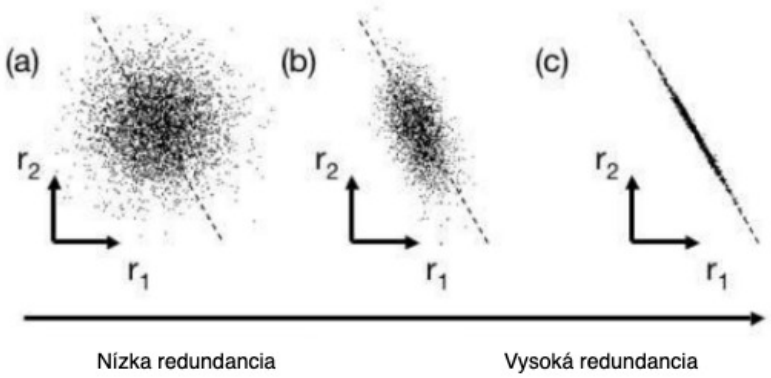
$$X = \begin{bmatrix} a \\ b \\ \vdots \\ c_m \end{bmatrix} \tag{3.16}$$

Po transponovaní získame maticu, ktorej každý riadok reprezentuje jednu inštanciu definovanú v n dimenzionálnom priestore. Stĺpce reprezentujú hodnoty v daných dimenziách. Samotnú kovariančnú maticu získame na základe rovnice 3.17.

$$C_X \equiv \frac{1}{n-1} XX^T \tag{3.17}$$

Výsledná matica má tieto vlastnosti:

- je symetrická;
- má rozmery m x m;
- jej prvky na diagonále reprezentujú varianciu dát v danej dimenzií;



Obr. 3.10: Obrázok popisuje tri prípady rozloženia dát, v ktorých skúmame redundanciu dimenzionality dát. Na ľavej strane vidíme prípad kde sú pre reprezentáciu jednotlivých dát potrebné obe dimenzie. Na pravej strane je zobrazený opačný prípad, v ktorom je jedna dimenzia nadbytočná a neposkytuje nám významnú informáciu[9].

- prvky mimo diagonály reprezentujú kovarianciu dát v daných dimenziách, kde veľké hodnoty znamenajú vysokú redundanciu.

Aby sme zredukovali redundanciu a utlmili šum, je potrebné dosiahnuť aby bola kovariančná matica diagonálna, teda aby mala pozitívne hodnoty len na diagonále. Existuje viacero spôsobov ako diagonalizovať kovariančnú maticu získanú z analyzovaných dát. Jedným z nich je metóda *PCA* [21].

Uvažujme namerané dáta X, kde každý stĺpec reprezentuje jednu nameranú inštanciu v niekoľko dimenzionálnom priestore. Následne nech Y je iná reprezentácia nameraných dát a táto reprezentácia je lineárne transformovaná maticou P. Rovnica 3.18 reprezentuje tento vzťah.

$$PX = Y \tag{3.18}$$

Metóda *PCA* predpokladá, že jednotlivé bázové vektory matice P sú ortonormálne, teda že všetky vektory sú navzájom ortogonálne a normované. Druhým predpokladom metódy je to, že najviac významné smery v dátach sú tie, ktoré majú najväčšiu varianciu. Na základe týchto predpokladov máme metódu, ktorá je schopná hodnotiť významnosť jednotlivých smerov, na základe veľkosti variancie dát v týchto smeroch. Úlohou metódy je teda nájsť ortonormálnu maticu P, ktorá transformuje analyzované dáta X do Y reprezentácie. Kovariančná matica novo získanej reprezentácie je diagonalizovaná a jednotlivé riadky matice P reprezentujú hlavné smery variancie dát [21]. Tieto smery sa nazývajú vlastné vektory. Vlastné čísla reprezentujú varianciu dát v smere vlastného vektoru.

Ako už bolo uvedené, metódu *PCA* táto práca využíva napríklad pre redukciu dimenzionality používateľských dát. Štandardné použitie tejto metódy pozostáva zo šiestich krokov [14]:

1. Výpočet kovariančnej matice z originálnych d dimenzionálnych dát X.

2. Výpočet vlastných vektorov a vlastných čísel.

[9] https://arxiv.org/abs/1404.1100

3. Zoradenie vlastných čísel od najväčšieho po najmenšie.

4. Výber k najväčších vlastných čísel. Počet týchto čísel predstavuje počet dimenzií v novom podpriestore.

5. Konštrukcia projekčnej matice W, zloženej z vlastných čísel prislúchajúcim k vybraným vlastným číslam.

6. Transformácia originálnych dát X pomocou projekčnej matice W do nového podpriestoru.

Pri implementovaní a analýze možných riešení bola táto metóda používaná aj pre vizualizáciu vstupných dát. Konkrétne bola použitá pri analýze aktivity používateľa, kde jednotlivé dimenzie reprezentovali počet sekúnd aktivity v jednotlivých kategóriách používaných aplikácií. Tento viacrozmerný priestor bol zredukovaný na tri dimenzie, ktoré boli následne vizualizované priestorovým grafom. Dimenzie boli vybrane na základe najvyššej variancie dát v daných smeroch, konkrétne sa teda jednalo o tri hlavne bázové vektory matice P. Obrázok 3.11 poskytuje intuitívny príklad toho, ako táto metóda vyberie dve najvýznamnejšie smery variancie vstupných dát.

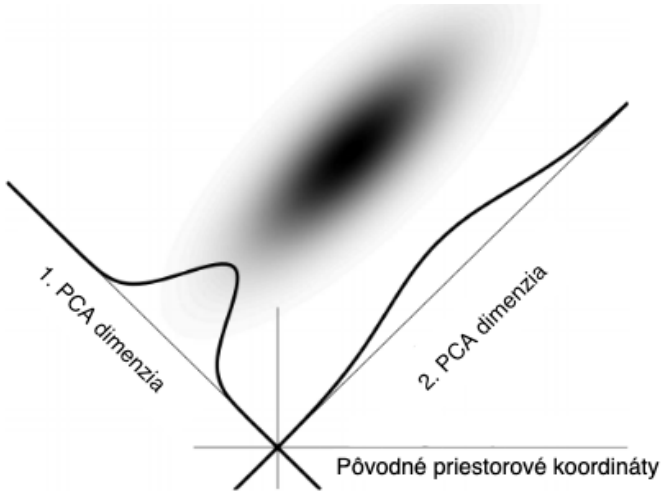
Technika *PCA* je často využívaná pri zhlukovaní dát, kedy sa v novo vytvorenom priestore využije niektorý zo zhlukovacích algoritmov, napríklad algoritmus *K-means*. Projekcia do priestoru, v ktorom sú dáta dobre rozdeliteľné funguje v prípade, že originálne dáta sú lineárne rozdeliteľné. Nelineárne rozdelenie dát bolo znázornené na obrázku 3.3, kde bol popísaný problém separácie dát pomocou niektorých zhlukovacích algoritmov. Pre prípady nelineárneho rozloženia dát vo viacrozmernom priestore je používaná metóda s názvom *Kernel Principal component analysis* (*KPCA*).

Základnou myšlienkou, akou algoritmus *KPCA* pracuje s nelineárnymi dátami je projekcia týchto dát do viacrozmerného priestoru. V tomto priestore sú analyzované dáta už lineárne separabilné. Táto projekcia sa nazýva kernelova funkcia, môžeme ju značiť Φ. Táto projekcia je vykonávaná na základe pridaní nelineárnych kombinácií originálnych d-dimenzionálnych priestorových komponent. Napríklad v prípade že x pozostáva z dvoch komponent, je x' výsledkom aplikovania kernelovej funkcie. Rovnica 3.19 definuje aplikáciu tejto funkcie [14].

$$\begin{gathered} x = [x_1 \quad x_2]^T \quad x \in R^d \\ x' = [x_1 \quad x_2 \quad x_1 x_2 \quad x_1^2 \quad ... \,]^T \quad x \in R^k \quad (k >> d) \end{gathered} \tag{3.19}$$

Existuje viacero druhov definície kernel funkcie. Jednou z nich je *Gaussian radius basis function* (*BRF*) a na základe tejto funkcie bude predstavený aplikácie *KPCA* metódy pre separáciu nelineárnych dát. Funkcia *BRF* obsahuje jeden volný parameter, ktorý je vhodné optimalizovať pre dosiahnutie dobrého výsledku [14].

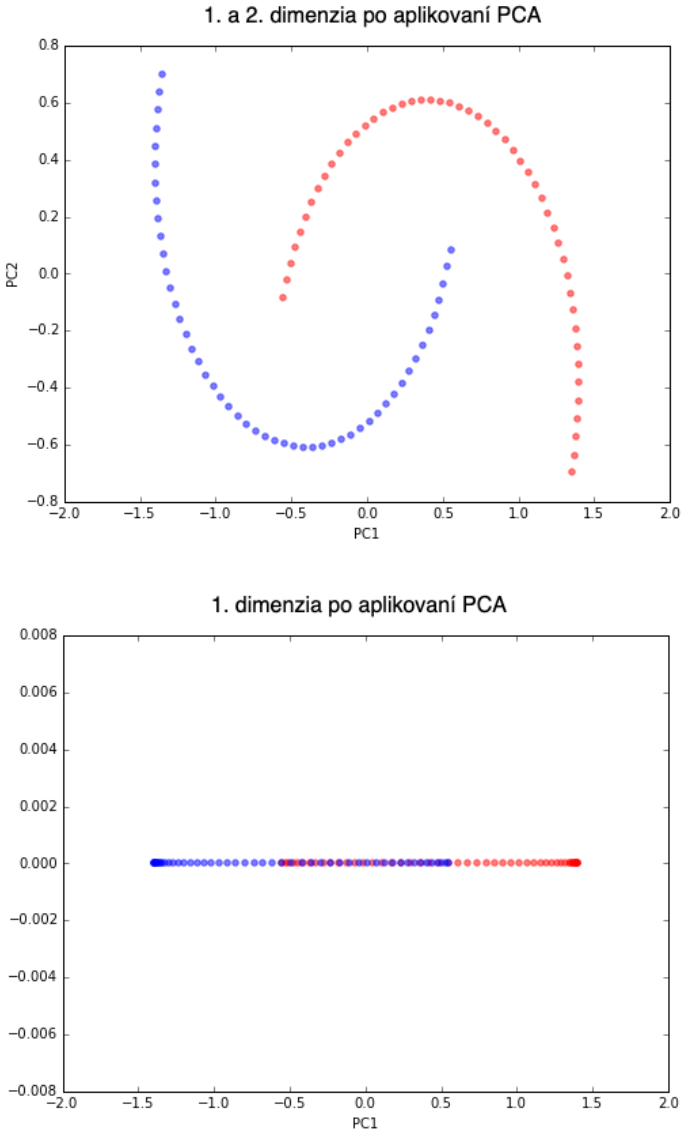
Opäť sa vrátime k nelineárnemu rozloženiu dát z časti 3.2 a to konkrétne druhá časť z obrázku 3.3. V tomto obrázku sú dáta rozdelené do dvoch skupín, ktoré nie sú linearne separabilné. Príklad je reprezentovaný dvojicou obrázkov, kde prvý 3.12 znázorňuje podobu dát po aplikovaní lineárnej *PCA* metódy. Z tejto ukážky je vidieť, že redukcia dimenzionality nám neuľahčila následnú segmentáciu dát. Druhý obrázok 3.13 ukazuje povahu dát po aplikovaní *KPCA*. Z obrázku je vidieť, že projekcia do rovnako dimenzionálneho priestoru nám poskytla lineárne separabilné dátové rozloženie. Následná redukcia dimenzionality dokázala



Obr. 3.11: Reprezentácia výberu bázy pomocou metódy *PCA*, ktorá popisuje smer najväčšej variancie analyzovaných dát[10].

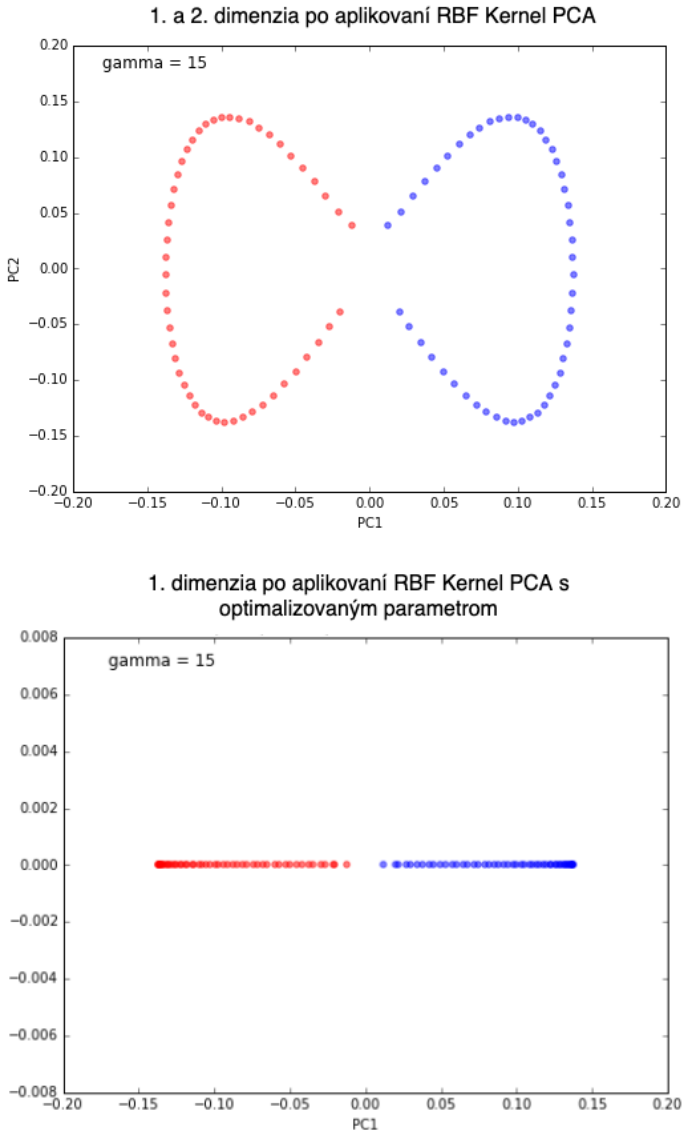
výsledne dáta projektovať to jednoducho separabilného jedno-dimenzionálneho priestoru. Je nutné dodať, že táto metóda nerozlišuje triedu (farbu) jednotlivých dátových bodov. Výsledné rozdelenie je získané po optimalizácií parametru *gamma*. Tento parameter sa viaže k použitej kernel funkcií.

[10] https://www.fit.vutbr.cz/study/courses/IKR/public/prednasky/03_extrakce_priznaku/extrakce_priznaku.pdf



Obr. 3.12: Reprezentácia spracovania nelineárne separabilných dát metódou *PCA* a následná redukcia dimenzionality na základe smeru najväčšej variancie dát. Z tejto ukážky je vidieť, že redukcia dimenzionality nám neuľahčila následnú segmentáciu dát[11].

[11] https://sebastianraschka.com/Articles/2014_kernel_pca.html
[12] https://sebastianraschka.com/Articles/2014_kernel_pca.html



Obr. 3.13: Reprezentácia spracovania nelineárne separabilných dát metódou *KPCA* s *BRF* funkciou. Z obrázku je vidieť, že narozdiel od predošlého príkladu s využitím metódy *PCA*, nám táto projekcia do rovnako dimenzionálneho priestoru poskytla lineárne separabilné dátové rozloženie. Následná redukcia dimenzionality dokázala výsledne dáta projektovať to jednoducho separabilného jedno-dimenzionálneho priestoru. Je nutné dodať, že táto metóda nerozlišuje triedu (farbu) jednotlivých dátových bodov. Výsledné rozdelenie je získané po optimalizácií parametru *gamma*[12].

Kapitola 4

Návrh modelu pre popis chovania používateľa

V tejto kapitole sa nachádza bližší popis dát, nad ktorými prebieha analýza používateľského správania. Následne je predstavený návrh modelu, ktorý popisuje používateľské správanie a taktiež model popisujúci skupinu používateľov. Model používateľského chovania je reprezentovaný profilom používateľa. Nad týmto profilom je následne vykonávaná detekcia anomálnej aktivity, ktorej návrh sa nachádza v časti 4.4. Záver kapitoly obsahuje návrh vizualizácie používateľského modelu z pohľadu analýzy dát a z pohľadu používateľa vytvorenej aplikácie.

4.1 Popis vstupných dát

Táto práca stavia na dostupnosti dát, ktoré popisujú používateľskú aktivitu v závislosti na čase. Jedná sa o zoznam udalostí, ktoré daný používateľ vykonal na svojom pracovnom zariadení. Taktiež sa predpokladá dostupnosť týchto dát od viacerých používateľov pre vytvorenie skupinových profilov.

Udalosťou rozumieme napríklad aktívne strávený počet sekúnd v aplikácií, prípadne na webovej stránke určitého typu. Združením týchto udalostí pre jedného užívateľa nám vznikne časová rada popisujúca aktivitu používateľa. Každá dátová inštancia tejto rady sa teda skladá z nasledujúcich údajov:

- identifikácia používateľa;
- časová značka;
- typ udalosti (môže ísť napríklad o identifikáciu aplikácie);
- počet sekúnd, počas ktorých bola táto akcia aktívna (v prípade udalosti popisujúcej aktívny čas).

Medzi typy udalostí, ktoré táto práca využíva patrí:

- aktívny čas v určitej aplikácií;
- aktívny čas na určitej webovej stránke;
- súborová operácia.

Poskytnuté informácie je následne nutné predspracovať tak, aby ich bolo možné použiť pri modelovaní používateľského chovania. Kapitola 5, ktorá popisuje spôsob implementácie, obsahuje bližší popis predspracovania vstupných dát.

Vstupné dáta teda popisujú používateľovu aktivitu v jednotlivých aplikáciách, prípadne na webových stránkach. Je zrejme, že rozličných aplikácií a webových stránok existuje veľké množstvo. V prípade, že v modely budeme uvažovať aktivitu v danej aplikácií ako nezávislú vlastnosť, tak každá takáto aplikácia predstavuje jednu dimenziu v používateľskom modely. Takýto prístup je preto príliš neefektívny a v prípade webových stránok sa táto nevýhoda ešte znásobuje. Naviac, ako už bolo spomenuté v časti 2.5, narastajúca dimenzionalita dát spôsobuje problémy pri detekcií anomálií vo vstupných dátach a taktiež pri zhlukovaní vstupných dát do skupín. Preto v prípade, že sa vo vstupných dátach nachádza veľké množstvo záznamov s rôznymi typmi udalostí o aktivite, je vhodné tieto vstupné dáta kategorizovať. Dataset, na ktorom boli testované výsledky navrhovaného modelu takúto kategorizáciu aplikácií a webových stránok obsahuje. Návrh aplikácie ale počíta aj so situáciou, kedy kategorizácia aplikácií a webových stránok nie je dostupná. V takomto prípade je možné očakávať horšie výsledky a to najmä v prípade tvorby skupinových profilov.

Práca taktiež uvažuje iné druhy dát, ktoré majú taktiež závislosť na čase. Ide napríklad o súborové operácie. Takéto Informácie poskytujú bližší kontext detekovanej anomálie. Tento kontext je následne zobrazený vo vyvinutej vizualizačnej aplikácií a poskytuje možnosť lepšieho pochopenia danej anomálie. Zobrazené súborové operácie slúžia taktiež na overenie, či daný používateľ nevyniesol citlivé dáta z firemného prostredia. Časť 4.5, popisuje návrh a použitie vizualizačnej aplikácie, ktorá je súčasťou tejto práce.

4.2 Profil používateľa

Model používateľa reprezentuje všetky informácie o používateľovi, ktoré máme k dispozícií. Tieto informácie sa rozdeľujú na dve kategórie, explicitné a implicitné. Explicitné informácie získame zo vstupných dát, bez nutnosti aplikovania dodatočných algoritmov. Základným spoločným rysom vstupných dát, ktoré táto práca analyzuje, je závislosť týchto dát na čase. Okrem základných informácií, akými sú napríklad identifikácia používateľa, je model chovania používateľa zložený z nasledujúcich troch hlavných komponent:

- agregovaná, jednodňová aktivita používateľa v jednotlivých kategorizovaných aplikáciách a na webových stránkach;
- agregovaná, jednohodinová celková aktivita používateľa;
- dodatočná aktivita (súborové operácie) pre poskytnutie bližšieho popisu v prípade detekovanej anomálie.

Prvá komponenta popisuje strávený čas v jednotlivých kategóriách aplikácií a webových stránok. Táto informácia popisuje chovanie používateľa a na základe tejto informácie je vytvorený skupinový profil, ktorého návrh sa nachádza v časti 4.3. Komponenta popisujúca celkovú aktivitu používateľa je využívaná pri detekcií anomálnej aktivity. Práca s touto informáciou je bližšie popisovaná v návrhu detekcie anomálií v časti 4.4. Posledná uvedená komponenta slúži pre používateľa vytvorenej aplikácie a poskytuje bližší kontext detekovanej anomálie.

Implicitné informácie získame zo vstupných dát pomocou dátovej analýzy. Ide napríklad o štatistickú analýzu explicitných dát. Táto analýza zahŕňa výpočet priemeru, variancie

alebo štandardnej odchýlky v dátach. Z tejto analýzy je možné získať štatistické informácie akými sú napríklad:

- priemerná doba strávená v aplikácií;
- priemerná pracovná doba používateľa;
- typy aplikácií s nízkou variáciou aktivity.

Tieto informácie prispievajú k lepšiemu popisu správania používateľa. Tieto informácie je ďalej možné využiť napríklad pri zhlukovaní používateľov do skupín, kde každá z týchto informácií môže predstavovať ďalšiu vlastnosť používateľa. Návrh používateľného modelu počíta taktiež s vizualizáciou tohto modelu. Samotný návrh vizualizácie na nachádza v časti 4.5.

4.3 Profil skupiny

Pri analýze chovania používateľov je dôležitou úlohou nájdenie používateľov s podobným chovaním. To je dosiahnuté porovnaním modelu analyzovaného používateľa s už analyzovanými modelmi. Je zrejmé, že takéto porovnanie by bolo pri vysokom počte používateľov časovo veľmi náročné. Z tohoto dôvodu je vhodné použiť metódy zhlukovania modelov. Pri návrhu modelu chovania je preto nutné myslieť na možnosti následného zhlukovania takéhoto modelu. Vytvorené zhluky budú následne reprezentovať chovanie skupiny používateľov. Chovanie skupiny môže prispieť k správnemu pochopeniu situácie v rámci analyzovaného prostredia. Jednotné skupinové chovanie môže viesť k správne definovanému kontextu pre definovanie kontextových anomálií. Ako príklad je možné uviesť situáciu, v ktorej používateľ nie je v bežný pracovný deň vôbec aktívny. V prípade, že užívateľ je zaradený do určitej skupiny používateľov a žiadny z nich nie aktívny, výsledná situácia nemusí byť vyhodnotená ako anomálna.

Skupinový profil v tejto práci bude vytvorený pomocou zhlukovacích algoritmov a to na základe dát popisujúcich aktivitu používateľa. Tieto dáta sú súčasťou modelu chovania používateľa, konkrétne sa jedná o dáta reprezentujúce aktivitu v kategorizovaných aplikáciách a kategorizovaných webových stránkach. Z týchto dát sa vytvorí model dňa, popisujúci aké kategórie aplikácií a webových stránok používateľ v daný deň využíva. Jeden deň je teda reprezentovaný ako bod vo viacrozmernom priestore, kde jednotlivé dimenzie reprezentujú čas strávený v danej aplikačnej alebo webovej kategórií.

Pred návrhom tvorby skupinových profilov, je potrebné správne analyzovať a predspracovať dostupné dáta. Týmto krokom zlepšíme výsledky pri aplikácií zhlukovacích algoritmov. V rámci predspracovania sa uvažuje o nasledujúcich operáciách:

- filtrácia aplikačných kategórií, vzťahujúcich sa k webovým prehliadačom;
- združenie aplikačných a webových kategórií s rovnakým účelom;
- filtrácia dní s nízkou celkovou aktivitou.

Cieľom filtrácie dimenzií, ktoré popisujú aktivitu v rámci webových prehliadačov je odstránenie redundantných dimenzií modelu. Táto filtrácia vychádza z faktu, že aktivita vo webovom prehliadači je plne popísaná súčtom aktivít v jednotlivých kategóriách webových stránok.

Nasledujúcou úpravou je združovanie aplikačných a webových kategórií s rovnakým účelom. Ide napríklad o prípad kategórie emailového klienta, kedy pre tento účel existuje ako desktopová tak aj webová aplikácia. Takáto upráva by mala byť vykonaná v závislosti na konkrétnych vstupných dátach. Touto úpravou sa napríklad zamedzí uvažovaniu o aktivite vo webovom a aplikačnom emailovom klientovi, ako o dvoch rozličných činnostiach. Každá takáto upráva taktiež zredukuje počet dimenzií modelu.

Posledná navrhovaná úprava vstupných dát pozostáva z odstránenia málo aktívnych dní zo skupinového modelu. Táto úprava vychádza z faktu, že dni, v ktorých je používateľ málo aktívny (jednotky až desiatky minút), nepopisujú v plnej miere chovanie používateľa. Takéto dni naopak vnášajú do modelu šum a návrh modelu počíta s ich filtrovaním.

Po predspracovaní dát, sú jednotlivé dni používateľov spojené do jedného skupinového modelu. Obrázok 4.1 vizuálne popisuje štruktúru tohto modelu.

	Aplikačné kategórie			Kategórie webových stránok		
	E-mailový klient	...	Hry	Správy	...	Sociálne siete
Používateľ 1, deň 1.	300		1200	2000		0
Používateľ 1, deň 2.	2000					
⋮						
Používateľ 1, deň n.	0					
Používateľ 2, deň 1.	0					
⋮						
Používateľ m, deň n.	500					

Obr. 4.1: Obrázok prezentuje štruktúru modelu skupinového profilu. Dostupné dni, popisujúce aktivity jednotlivých používateľov sú združené do riadkov matice. Stĺpce matice prezentujú jednotlivé kategórie aplikácií a webových stránok. Hodnoty matica reprezentujú počet sekúnd, kedy bol používateľ aktívny v danej kategórií.

Ako už bolo popisované v časti 2.5, so zvyšujúcou sa dimenzionalitou dát môžu nastať nepresnosti pri zhlukovaní dát. Vstupné dáta, ktoré uvažuje vyvíjaná aplikácia sú kategorizované a každá kategória predstavuje jednu dimenziu vo vytvorenom modely. Týchto kategórií ale môže byť ľubovolne množstvo. Návrh aplikácie taktiež počíta s prípadom vstupných dát, kedy dáta nie sú kategorizované a v tomto prípade každá aplikácia a webová stránka predstavuje samostatnú dimenziu. V konečnom dôsledku tak počet dimenzií modelu nie je dopredu známy a teda závisí na povahe vstupných dát. Z tohto dôvodu sú aplikované algoritmy pre redukciu dimenzionality.

Návrh spôsobu redukcie dát počíta s možnosťou výberu algoritmu. V prípade využitia algoritmu *PCA*, je možné spätne vyhodnotiť, ktoré dimenzie predstavovali najväčšiu varianciu v analyzovaných dátach. Metóda *KPCA* túto možnosť neumožňuje a to z dôvodu, že analyzované dáta mapuje do novovytvoreného priestoru. V tomto prípade je ale možné využiť vytvorené zhluky popisujúce chovanie používateľov a dodatočné spočítať charakte-

ristiky jednotlivých zhlukov. Týmto spôsobom vznikne popis zhluku na základe aktivity používateľov v jednotlivých zhlukoch.

	Používateľ 1	Používateľ 2	Používateľ 3	Používateľ 4
Používateľ 1	---	10	0	1
Používateľ 2	10	---	0	1
Používateľ 3	0	0	---	10
Používateľ 4	1	1	10	---

Obr. 4.2: Obrázok predstavuje vytvorenú maticu vzťahov medzi jednotlivými používateľskými modelmi. Matica teda vyjadruje podobnosť v správaní používateľov. Hodnoty matice vyjadrujú, koľko krát sa daná dvojica používateľov nachádzala v rovnakom zhluku. Na obrázku je ukážka vytvorenej matice a to v prípade, že bol zhlukovací algoritmus spustený desať krát. Vo výsledku sú vytvorené dva zhluky, kedy v prvom sa nachádzajú prvý dvaja používatelia a v druhom sa nachádza používateľ tri a štyri. Hodnoty 1 v matici predstavujú stav, kedy bol chybne inicializovaný počiatočný centrálny bod zhluku, výsledkom čoho boli vytvorené nesprávne skupiny používateľov.

Samotná tvorba skupinových profilov sa delí na dva kroky. V prvom kroku sa aplikuje vybraný zhlukovací algoritmus na novovzniknutý priestor, ktorý sme získali po aplikácií algoritmov redukcie dimenzionality. Tým vzniknú zhluky, reprezentujúce podobné správanie jednotlivých používateľov v jednotlivých dňoch. Pre zhlukovanie je možné použiť niektorý z algoritmov, popisovaných v časti 3.2. Časť 5.2.2 bližšie popisuje konkrétny vyber algoritmu pre tvorbu popisovaných zhlukov. Niektoré zhlukovacie algoritmy vyžadujú špecifikáciu počtu hľadaných zhlukov. Jedným z takýchto algoritmov je napríklad algoritmus *k-means*. Pre automatický odhad tohoto počtu je možné využiť validačnú metódu siluety[1]. Táto metóda pracuje s priemernou hodnotu šírky siluety pre každý zhluk a používa sa na analýzu vzdialenosti medzi jednotlivými zhlukmi. Silueta reprezentuje pomer podobnosti a odlišnosti od ostatných zhlukov. Algoritmus je teda potrebné spustiť niekoľko krát, vždy s iným počtom špecifikovaných zhlukov. Na záver sa vyberie ten počet zhlukov, ktorý reprezentuje najoptimálnejšiu separáciu vytvorených skupín.

V druhom kroku sa analyzujú vytvorené zhluky a pre každého používateľa sa vyberie ten zhluk, v ktorom sa nachádza naviac jemu priradených dní aktivity. Pre elimináciu nesprávnej inicializácie centrálnych bodov jednotlivých zhlukov sa algoritmus spustí niekoľko krát, vždy s náhodnou inicializáciou. Týmto spôsobom sa vytvorí matica reprezentujúca spoločné chovanie medzi používateľmi. Matica je symetrická a každý riadok a stĺpec reprezentuje jednotlivých používateľov. Hodnoty matice vyjadrujú počet, koľko krát bol daný používateľ v rovnakom zhluku s iným používateľom. Na základe najvyšších hodnôt je tak vyhodnotená skupina používateľov s podobným chovaním. Na obrázku 4.2 je vidieť, ako táto matica popisuje silu vzťahu medzi jednotlivými používateľmi. Týmto spôsobom sú porovnávané modely jednotlivých používateľov a na základe vyhodnotenia tejto matice sú vytvorené skupinové profily.

[1] https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Porovnávať jednotlivé používateľské modely, konkrétne ich aktivitu, je možné aj vizuálne, pomocou porovnania celkovej aktivity dvoch používateľov vzhľadom k časovej ose. Táto vizualizácia poskytuje porovnanie normálneho a anomálneho správanie medzi používateľmi a jej návrh sa nachádza v časti 4.5.2.

4.4 Detekcia anomálií

Na základe sekvenčnej povahy dostupných vstupných dát, sú pre analýzu anomálií v týchto dátach navrhnuté algoritmy, popisované v časti 3.4. Anomálie sú teda detekované na základe dekompozície časových rádov. Konkrétne sa jedná o sezónnu dekompozíciu udalostí reprezentujúcich používateľovu aktivitu. Konkrétny spôsob detekcie anomálií, ktorý bol využitý pre potreby tejto práce, popisuje časť 5.3.2, ktorá sa zaoberá implementáciou analýzy časových radov.

Za samotné anomálie sú považované v tejto práci časovo ohraničené úseky, kedy bola v nezvyčajný čas detekovaná používateľova aktivita. Nezvyčajný čas je čas, v ktorom používateľ nie je bežne aktívny na svojom pracovnom zariadení. Jedná sa teda napríklad o čas mimo pracovnú dobu alebo napríklad cez víkend. Algoritmus, ktorý odhaľuje takéto anomálne správanie musí byť odolný na validné no problematické typy situácií.

Prvou z nich je situácia, kedy je užívateľ aktívny nepravidelne. Ako príklad je možné uviesť situáciu, kedy je používateľ aktívny vo večerných hodinách, každý druhý týždeň. Takýto stav sa dá popísať ako viac násobná sezónnosť, kedy sa v používateľovej aktivite objavuje viac druhov sezónneho správania. Ďalšou problematickou vlastnosťou je fakt, že aj keď používateľ často vykonáva nejakú činnosť v pravidelných intervaloch, tak tieto intervaly vo väčšine prípadov nezačínajú a nekončia v rovnaký čas. Takýto stav je nazvaný ako fluktuácia používateľovej aktivity a na túto skutočnosť musí byť analýza pripravená.

Pri návrhu systému pre detekciu anomálneho správania je teda vhodné zhrnúť scenáre, na ktoré by mal systém reagovať. Nasledujúci zoznam udalostí, popisuje navrhnuté scenáre:

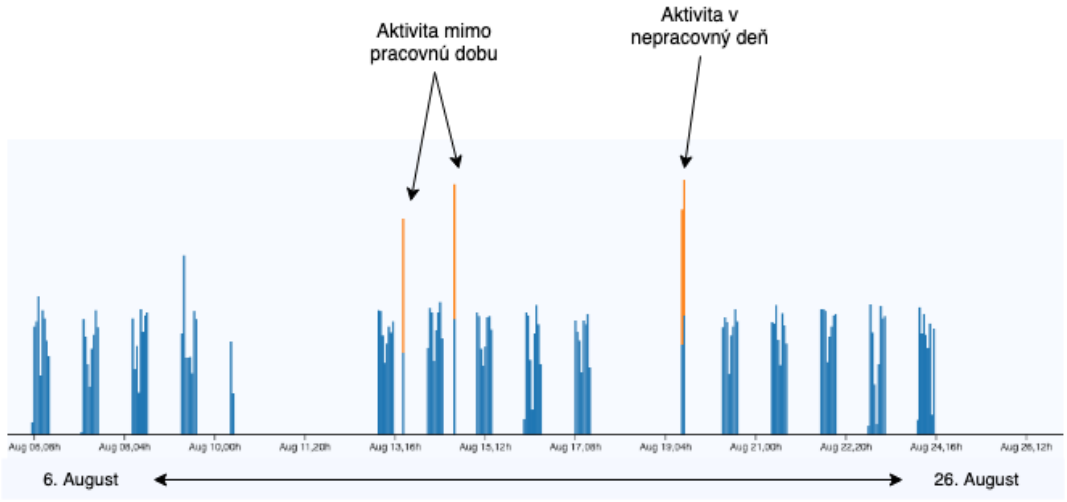
- anomálna aktivita mimo pracovnú dobu;
- anomálna aktivita v nepracovný deň;
- neanomálna aktivita po, prípadne pred bežnou aktivitou;
- neanomálna aktivita v nepracovnú dobu alebo deň - prispôsobenie sa periodicky opakujúcej sa aktivite.

Prvý scenár popisuje stav, kedy by mala byť zaznamenaná aktivita používateľa mimo jeho obvyklú pracovnú dobu. Konkrétne sa jedná napríklad o záznam aktivity vo večerných a nočných hodinách v aktivite používateľa s bežnou dennou pracovnou dochádzkou. Druhý scenár popisuje podobnú udalosť, no v tomto prípade sa jedná o aktivitu v deň, kedy používateľ v minulosti nepracoval. Oba scenáre a s nimi súvisiace anomálne aktivity zachycuje obrázok 4.3.

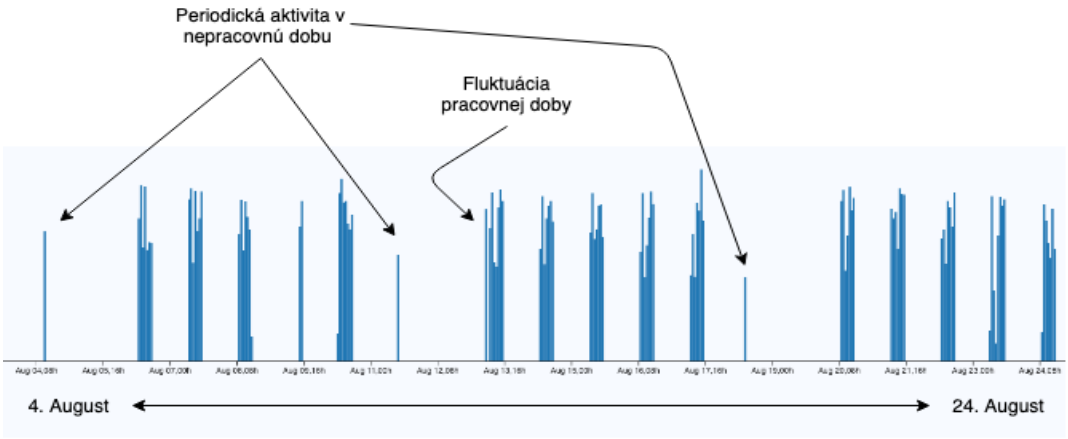
Ďalšie scenáre sa zameriavajú na filtráciu falošných pozitív. Takéto udalosti majú povahu anomálnej aktivity mimo pracovnú dobu, no na základe pravidelnej aktivity môžeme túto aktivitu označiť ako neanomálnu. Ide teda o aktivitu, ktorá sa vyskytuje v oblasti pred alebo po pracovnej dobe. Taktiež do tejto oblasti spadá fluktuácia používateľovej aktivity, kedy sa nejaká udalosť opakuje periodicky no v rozmedzí niekoľkých hodín. Posledným scenárom je prispôsobenie sa používateľovej aktivite, kde nie je táto aktivita pravidelne

každý týždeň. Ide napríklad o aktivitu vykonávanú každý druhý týždeň, prípadne raz do mesiaca. Výsledky posledných dvoch scenárov sú zobrazené na obrázku 4.4.

Pre dosiahnutie lepších výsledkov je vhodné, aby analýza časových radov prebiehala pre každý deň v týždni osobitne. Jedna časová rada sa teda rozdelí na sedem rád a nad týmito radami je následne vykonávaná analýza anomálneho správania. Takýto spôsob predspracovania dát je logickým krokom pre analýzu aktivity vo firemnom prostredí, kde sa často stretávame s určitou zaužívanou dochádzkou.



Obr. 4.3: Obrázok reprezentujúci aktivitu používateľa, agregovanú po jednej hodine. Aktivita zvýraznená oranžovou značkou reprezentuje možný scenár, detekcie anomálnej aktivity v prípade, že daná aktivita je zaznamenaná mimo pracovnú dobu alebo počas nepracovný deň.



Obr. 4.4: Obrázok reprezentujúci aktivitu používateľa, agregovanú po jednej hodine. Aktivita zvýraznená šípkou približuje možný scenár, kedy je zaznamenaná aktivita mimo pracovnú dobu alebo v nepracovný deň správne uvažovaná ako neanomálna aktivita.

Výsledné časové úseky, ktoré budu vyhodnotené ako anomálne, budú používateľa upozorňovať vo vizualizačnej aplikácií. Pre bližšie pochopenie a posúdenie vzniknutej anomálie bude môcť používateľ porovnať podozrivé časové úseky s ostatnými používateľmi. Títo

ostatní používatelia budú vybraný na základe vytvorených skupinových profilov. Týmto spôsobom je možné anomálne správanie vysvetliť na základe skupinovej aktivity. Ako príklad je možné uviesť udalosť nepravidelného školenia zamestnancov. V prípade, že sa toto školenie uskutoční v používateľom neobvyklí pracovný čas, bude tato anomálna udalosť detekovaná u väčšieho počtu zamestnancov. Takýto stav aplikácia vizualizuje a tým vyvráti podozrenie možnosti úniku dát z firemného prostredia.

Posledným prvkom, ktorý slúži pre bližšie pochopenie anomálnej činnosti je možnosť detailného nahliadnutia na aktivity, ktoré používateľ v daný čas vykonával. Jedná sa napríklad o informáciu o súborových operáciach. Pri analýze týchto udalostí je tak možné rýchlo vyhodnotiť závažnosť danej anomálie a poskytnúť detailné informácie v čo najrýchlejšom čase.

4.5 Návrh vizualizácie

Použitie správnych vizualizačných techník je nevyhnutné pre efektívnu analýzu používateľského chovania a následnú analýzu anomálneho chovania. Správne používateľské rozhranie zlepšuje uvedomenie si situácie, ktorá v systéme nastala. Hlavný cieľ využívania správnych vizualizačných nástrojov a techník je ich správna interakcia s nástrojmi pre efektívnu analýzu objemných dát. Vizualizácia taktiež pomáha efektívne rozlíšiť závažné anomálie od menej závažných, prípadne od falošných poplachov [16]. Vizualizácia použitá v rámci tejto práce sa delí na dva celky. Prvým je využitie vizualizácie pri analýze štruktúry dostupných dát. Táto vizualizácia je použitá pre lepšie pochopenie analyzovaných dát a pre správne určenie algoritmu, ktorý bude použitý v konečnej verzií systému. Druhá časť sa vzťahuje priamo na používateľské rozhranie vyvíjanej aplikácie. V rámci tejto práce sa jedná o vizualizačný nástroj, ktorý poskytuje bližší pohľad na aktivitu analyzovaných používateľov.

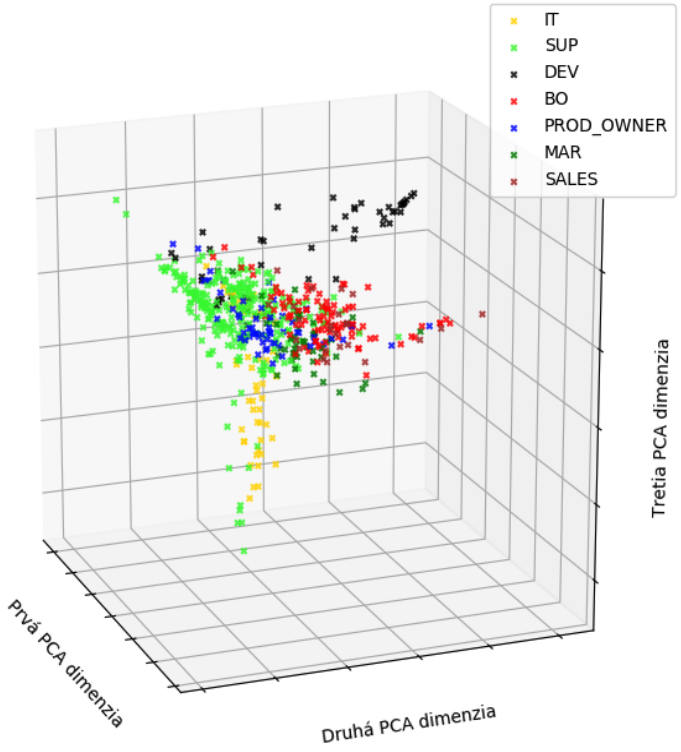
4.5.1 Návrh vizualizácie dát

Prieskum vstupných dát je prvým krokom pri analýze dostupných dát. Takýto prieskum zahŕňa analýzu variácie dát alebo analýzu korelácie jednotlivých vlastností modelu. Analyzovaný model je vo väčšine prípadov viacrozmerný. Takýto priestor je teda potrebné mapovať do dvojrozmerného alebo trojrozmerného priestoru. K tomuto účelu bol použitý algoritmus *PCA*, ktorý bol bližšie popisovaný v časti 3.6. Tento algoritmus využíva koreláciu dimenzií modelu. Výstupom je minimálny počet dimenzií, ktoré popisujú najvyššiu varianciu, teda informáciu o distribúcií vstupných dát [5].

Vizualizácia v rámci tejto práce bola využitá pri analýze používateľského modelu. Konkrétne sa jednalo o analýzu viacrozmerného priestoru, ktorý bol použitý pri tvorbe skupinového profilu. Redukcia dimenzionality umožnila posúdiť možnosti zhlukovania tohto priestoru pre potreby tvorby skupinových profilov. Algoritmus *PCA* poskytol okrem vizuálnej stránky aj informáciu o počte dimenzií, ktoré spolu obsahujú prevažné množstvo informácií obsiahnutých vo vstupných dátach. Obrázok 4.5 ukazuje príklad vizualizácie viacrozmerného modelu, ktorý bol použitý pre tvorbu skupinových profilov.

4.5.2 Návrh používateľského rozhrania

Cieľom vývoju tohto vizualizačného nástroja je poskytnúť používateľovi lepší prehľad a kontext o anomáliách, ktoré vyvíjaný systém odhalil. Konkrétne sa jedná o nasledujúce úlohy:



Obr. 4.5: Ukážka vizualizácie používateľských modelov pomocou metódy *PCA*. Táto vizualizácia bola využívaná pri návrhu a tvorbe skupinových profilov.

- umožniť filtrovať zobrazené informácie podľa časového intervalu;
- zobraziť informácie o konkrétnom analyzovanom používateľovi;
- zobrazovať aktivitu analyzovaného používateľa na časovej ose;
- zvýrazniť nájdené anomálne udalosti;
- možnosť porovnať aktivitu s inými používateľmi z rovnakého skupinového profilu;
- zobraziť detail časového úseku obsahujúci doplňujúce informácie, akými sú napríklad súborové operácie.

Nástroj, ktorý splňuje tieto požiadavky je možné využívať pre analýzu hromadných dát. Táto analýza bude prebiehať na základe interakcie s používateľom aplikácie. Z tohto dôvodu je potrebné, aby bolo ovládanie aplikácie intuitívne a prehľadné. V nasledujúcom texte bude bližšie popísaný návrh jednotlivých prvkov vizualizácie. Jednotlivé prvky vizualizácie sa delia na nasledujúce oblasti:

- filter zobrazených informácií;
- rozhranie pre porovnanie dvoch používateľských profilov;
- vizualizácia analyzovanej aktivity;

- detailný popis vybranej udalosti.

Jednotlivé vizualizačné prvky musia na zmeny vybraných vstupných dát náležite reagovať. Samotná zmena vstupných dát je vykonávaná na základe nastavenia filtru. Pomocou tohto filtra sa špecifikuje konkrétny používateľ a časový interval, pre ktorý sa vykonáva analýza časových radov. Pri každej zmene filtru sa vytvorí nová požiadavka na server, na základe ktorej sa vykoná analýza v novo zvolenom časovom intervale. Na základe tejto analýzy sú následne detekované anomálne udalosti.

Na základe výsledkov z tvorby skupinových profilov, musí byť používateľ schopný vybrať používateľa s podobným chovaním. Zoznam možných používateľov pre porovnanie, vychádza z výsledkov tvorby skupinových profilov. Týmto spôsobom je vybraný ďalší používateľ pre vzájomné porovnanie aktivít. Porovnanie jednotlivých používateľských modelov môže viesť k bližšiemu objasneniu detekovanej anomálnej udalosti.

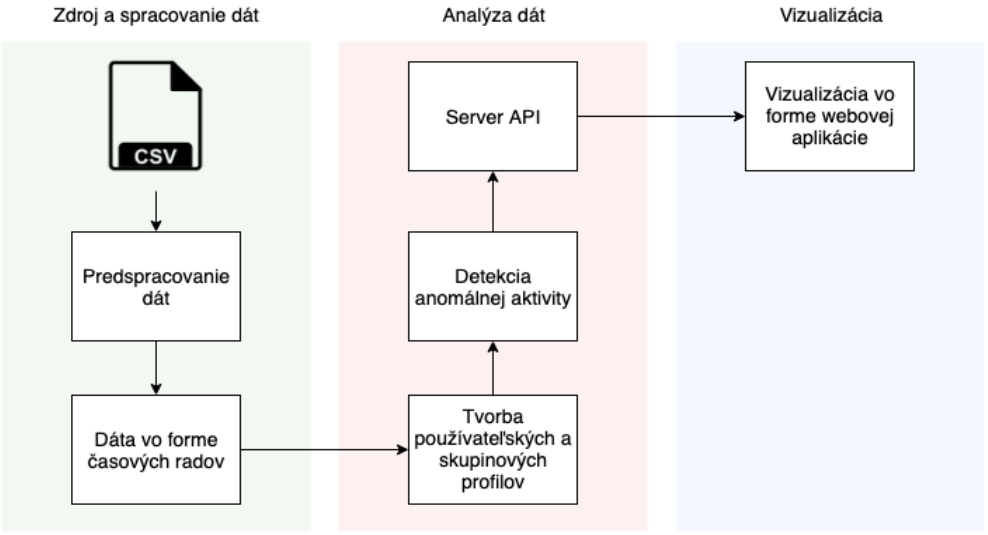
Analyzovaná aktivita používateľa je následne vizualizovaná vo forme časovej rady. Táto forma reprezentácie jasne popisuje časové úseky očakávanej a anomálnej aktivity. Očakávaná aktivita bude reflektovať počet sekúnd aktivity v danom časovom úseku. V prípade, že daný časový úsek je vyhodnotený ako anomálny, je na tomto mieste vytvorená značka, upozorňujúca na túto udalosť. Používateľ aplikácie využije túto reprezentáciu pre vyhodnotenie dôležitosti nájdenej anomálie. Ďalšou úlohou tohto prvku je porovnanie dvoch používateľských modelov. Na základe výberu používateľa z rovnakého skupinového profilu je aktivita oboch používateľov zobrazená v jednom grafe. Spolu s aktivitou sú zobrazené aj nájdené anomálne udalosti.

Jednotlivé nájdené anomálne udalosti je možné bližšie preskúmať v poslednej časti vizualizácie. Táto časť obsahuje detailný popis vybranej udalosti. Jedná sa hlavne o porovnanie aktivity medzi všetkými používateľmi rovnakého skupinového profilu. Aktivita jednotlivých používateľov je zobrazená v stĺpcovom grafe. Taktiež táto časť obsahuje ďalšie doplňujúce informácie o aktivite, akými sú napríklad vykonané súborové operácie. Ukážky jednotlivých častí používateľského rozhrania sa nachádzajú v časti 5.4. Práca taktiež obsahuje prílohu A, obsahujúcu podrobnejšiu ukážku aplikácie.

Kapitola 5

Implementácia

Súčasťou tejto práce je taktiež implementácia navrhnutého systému. Navrhnutý systém vytvára jednoducho ovládateľný nástroj pre detekciu a porovnávanie anomálnej aktivity medzi analyzovanými používateľmi. Systém sa skladá z niekoľkých častí a tieto časti sú zobrazené na obrázku 5.1.



Obr. 5.1: Popis stavebných blokov implementovaného systému pre detekciu anomálnej aktivity v používateľskom správaní.

V prvom bloku sú špecifikované prvky, popisujúce zdroj dát a spracovanie týchto dát. Primárnym zdrojom dát sú statické dáta vo formáte *Comma-separated values* (*CSV*), nachádzajúce sa v koreňovom adresári projektu. Aplikáciu je ale možné rozšíriť pre potreby jej integrácie do už existujúceho systému. V rámci tejto práce je ale riešenie so statickým zdrojom dát dostatočné.

V nasledujúcom texte tejto kapitoly sú popísane niektoré ďalšie dôležité časti systému. Jedná sa napríklad o spôsob implementácie vytvorenia skupinových profilov pomocou zhlukovacích algoritmov. V ďalšej časti je popísaná implementácia systému pre detekciu anomálnej aktivity. Na záver je predstavená vizualizačná aplikácia, pomocou ktorej je možné nájdené anomálne správanie bližšie analyzovať a porovnávať s ostatnými používateľmi.

5.1 Použité technológie

Pre implementáciu výsledného systému bol použitý jazyk Python[1] vo verzií 3.6. Tento jazyk je v oblasti analýzy a spracovania dát veľmi používaný[2]. Poskytuje mnoho metód, ktoré je možné využiť pre predspracovanie, zhlukovanie alebo pre interaktívnu analýzu dát. Medzi základné knižnice, ktoré boli pri analýze využité patria:

- numpy[3];
- pandas[4];
- matplotlib[5];
- scikit-learn[6].

Pre dekompozíciu časových rád boli využité niektoré algoritmy z menovaných knižníc. V konečnej implementácií bola ale nakoniec použitá implementácia algoritmu *RobustSTL*[7]. Pre potreby vizualizácie bola vytvorená webová aplikácia, pozostávajúca zo serverovej a klientskej časti. Na základe toho, že analýza dát je implementovaná v jazyku Python, bol tento jazyk vybraný aj pre aplikačné rozhranie vizualizácie. Konkrétne bola využitá knižnica Flask[8], ktorá spája webovú aplikáciu s časťou systému pre datovú analýzu. Webová aplikácia je tvorená s pomocou frameworku Angular[9] a jej implementácia bude bližšie popísaná v časti 5.4. Pre vývoj zobrazovacích prvkov bola použitá knižnica *D3.js*[10], ktorá umožňuje tvorbu interaktívnych grafov.

5.2 Tvorba skupinových profilov

Implementácia tvorby skupinových profilov vychádza z návrhu v časti 4.3. Proces vývoja prebiehal pomocou nástroja Jupyter notebooks[11], ktorý slúži pre interaktívnu prácu s jazykom Python nad analyzovanými dátami. Akonáhle boli výsledky analýzy dostatočné, implementácia bola presunutá z tohto nástroja do serverovej časti aplikácie. Týmto je možné iniciovať analýzu na základe používateľských požiadavok.

5.2.1 Predspracovanie vstupných dát

Vstupné dáta aplikácia očakáva vo formáte *CSV* a tieto dáta reprezentujú používateľskú aktivitu v jednotlivých kategóriách. Jedná sa teda o zoznám záznamov vo formáte tabuľky, kde jednotlivé stĺpce reprezentujú nasledujúce údaje:

- identifikácia používateľa;

[1] https://www.python.org/
[2] https://www.jetbrains.com/research/python-developers-survey-2018/
[3] https://www.numpy.org/
[4] https://pandas.pydata.org/
[5] https://matplotlib.org/
[6] https://scikit-learn.org/stable/
[7] https://github.com/LeeDoYup/RobustSTL
[8] http://flask.pocoo.org/
[9] https://angular.io/
[10] https://d3js.org/
[11] https://jupyter.org/

- časová značka;
- typ kategórie;
- počet aktívnych sekúnd.

Dáta sú následne transformované do matice, ktorá bola popisovaná v časti 4.3. Následne boli na maticu aplikované ďalšie úpravy, ako napríklad združenie určitých súvisiacich kategórií. Ide napríklad o prípad kategórie emailového klienta, kedy pre tento účel existuje desktopová aplikácia, ale taktiež aj webová aplikácia. Tieto úpravy sú ale dátovo špecifické a ich využitie zavisí od štruktúry použitých dát. Je preto možné tieto úpravy vynechať. Taktiež bola vynechaná aplikačná kategória webových prehliadačov, keďže túto kategóriu zastupujú konkrétne webové kategórie.

5.2.2 Redukcia dimenzionality a zhlukovanie dát

V priebehu implementácie boli použité viaceré metódy redukcie dimenzionality, popisované v časti 3.6. Dôvody použitia týchto metód boli popisované v návrhu, konkrétne v časti 4.3. Výber konkrétnej metódy bol zvolený na základe vyhodnotení výsledkov nad testovacími dátami. Implementácia počíta s dvoma druhmi metód. Prvou je metóda *PCA* a druhou je metóda *KPCA* so sigmoid kernelom. Konečnú metódu je ale možné jednoducho v implementácií zameniť. V prípade využitia metódy *KPCA* je potrebné špecifikovať parameter γ. Hodnota tohto parametru je nastavená na predvolenú hodnotu v implementácií scikit-learn knižnice a jej hodnota je definovaná rovnicou 5.1.

$$\gamma = \frac{1}{\text{Počet dimenzií modelu}} \tag{5.1}$$

Ďalším krokom v predspracovaní dát je vynechanie tých dát reprezentujúcich dni, v ktorých bola zaznamenaná nízka aktivita. Takéto dni nepredstavujú validný popis aktivity používateľa a môžu do modelu zanášať šum.

Implementovaný spôsob tvorby skupinových profilov pracuje na základe aplikácie zhlukovacích algoritmov v priestore, kde každý bod odpovedá jednodňovej aktivite niektorého používateľa. Pre tvorbu zhlukov bol využitý algoritmus *k-means* popisovaný v časti 3.2, vďaka jeho dobrej efektivite a dobrým výsledkom nad testovacími dátami. Pre určenie počtu počiatočných centrálnych bodov, bola využitá validačná metóda siluet.

V prvom kroku sa aplikuje algoritmus *k-means* na novovzniknutý priestor, ktorý sme získali po aplikácií algoritmov redukcie dimenzionality. V druhom kroku sa analyzujú vytvorené zhluky a pre každého používateľa sa vyberie ten zhluk, v ktorom sa nachádza naviac jemu priradených dní aktivity. Pre elimináciu nesprávnej inicializácie zhlukov sa algoritmus *k-means* spustí desať krát, vždy s náhodnou inicializáciou. Týmto spôsobom sa vytvorí matica reprezentujúca spoločné chovanie medzi používateľmi.

5.3 Detekcia anomálií

Detekcia anomálií je v kontexte tejto práce definovaná ako detekcia časových úsekov, ktoré boli vyhodnotené ako anomálne. Anomálnym úsekom sa myslí čas, kedy používateľ vykonával aktivitu mimo jemu obvyklú dobu. V prípade, že sa na používateľovu aktivitu pozeráme ako na úsek periodicky opakujúcej sa aktivity, môžme využiť algoritmy pre analýzu a dekompozíciu časovej rady. Nasledujúca časť kapitoly popisuje implementáciu detekcie anomálnej aktivity z návrhu v časti 4.4.

5.3.1 Predspracovanie vstupných dát

Rovnako ako pri tvorbe skupinových profilov, sú očakávané dáta vo formáte *CSV*. Tento krát dáta reprezentujú celkovú aktivitu vo všetkých aplikáciách a táto aktivita je združená na hodinové intervaly. Jednotlivé stĺpce reprezentujú tieto informácie:

- identifikácia používateľa;
- časová značka reprezentujúca špecifickú hodinu;
- počet aktívnych sekúnd.

Ako už bolo popisované v návrhu, dáta sú následne rozdelené na sedem časových rád, kde každá reprezentuje aktivitu v danom dni v týždni. Vstupné dáta nemusia poskytovať informáciu pre každú hodinu na časovej ose, preto je nutné tieto hodnoty doplniť hodnotou nula. Keďže sa v tejto analýze neberie ohľad na presnú intenzitu aktivity v danú hodinu, môžeme vstupné dáta previesť na binárne hodnoty. Binárne hodnoty vyjadrujú informáciu, či používateľ bol alebo nebol v danú hodinu aktívny.

5.3.2 Analýza časových radov

Samotná analýza a detekcia anomálií následne prebieha za pomoci algoritmu *RobustSTL*. Tento algoritmus rozdeľuje vstupnú časovú radu a hľadá v tejto rade sezónnu aktivitu. Algoritmus príma niekoľko parametrov medzi ktorými sú:

- počet prvkov definujúcich periódu sezóny;
- regularizačné parametre pre extrakciu trendu;
- počet minulých sezón, ktoré berieme do úvahy pri extrakcií sezónnosti;
- šírka susedstva, ktoré uvažujeme pri extrakcií sezónnosti;
- parametre pre potlačenie šumu;
- parametre pre extrakciu sezónnosti.

Tento algoritmus následne spustíme pre každú zo siedmich časových rád. Jednotlivé parametre boli nastavené pre potreby analýzy používateľského chovania v rámci štandardnej firemnej dochádzky. Ide napríklad o parameter definujúci dĺžku periódy sezóny. Táto perióda bola nastavená na hodnotu 24, teda jeden deň. Keďže je analyzovaná aktivita používateľa prevedená do binárnej podoby, tak je informácia o trende časovej rady eliminovaná. Z tohto dôvodu je parameter pre extrahovanie trendu nastavený tak, aby neovplyvňoval výslednú analýzu. Parameter určujúci počet minulých sezón je nastavený na 4. Táto hodnota pokrýva najväčšiu uváženú periódu, ktorou je jedna udalosť za mesiac. Je potrebné si pamätať, že dáta analyzujeme pre každý deň v týždni zvlášť. Optimálna hodnota nastavenia parameteru šírky susedstva je hodnota 1 alebo 2. Táto hodnota hovorí o koľko časových úsekov môže nájdená sezónnosť fluktuovať bez toho, aby bola aktivita označená za anomálnu. Ide napríklad o stav kedy je zamestnanec aktívny vždy od deviatej hodiny, no v jeden deň bude aktívny už o siedmej. To či bude na túto udalosť systém reagovať, záleží na nastavení tohto parametru. Parameter šumu v našom prípade nebol využitý, keďže analyzované dáta

sú binárne. Posledný parameter bol experimentálne overený a jeho hodnota bola nastavená v závislosti na testovaných dátach.

Výsledky analýzy poskytujú informácie o extrahovanej sezónnosti a o zvyšku aktivity, ktorá nebola popísaná sezónnosťou. Tento zvyšok predstavuje časové obdobie, ktoré reprezentuje chýbajúcu alebo prebytočnú aktivitu. V rámci tejto práce je pre nás dôležitá hlavne prebytočná aktivita v čase, kedy to je pre používateľa neobvyklé. Takáto aktivita signalizuje podozrivé správanie používateľa, kedy je napríklad možné detekovať únik dôležitých informácií z firemného prostredia. Nájdené udalosti sú následne podstúpené pre vizualizáciu a posúdenie používateľom aplikácie.

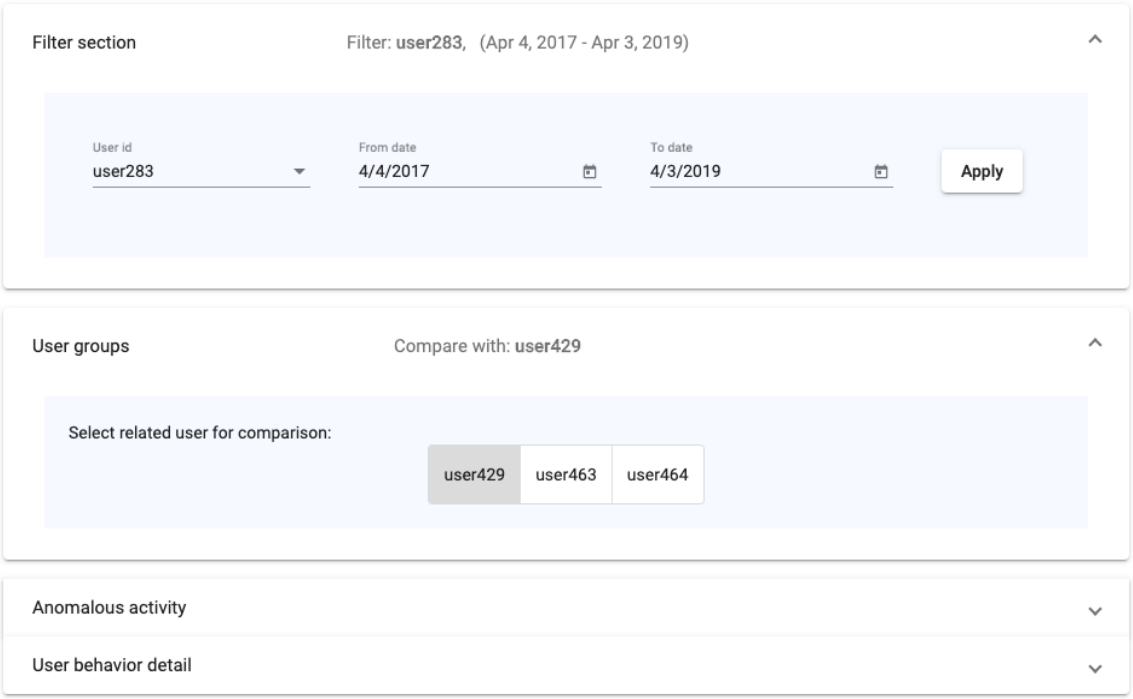
5.4 Vizualizácia

Vizualizácia časť nástroja pre analýzu používateľského chovania je realizovaná v podobe webovej aplikácie. Toto riešenie prináša širokú dostupnosť na rôznych druhoch zariadení. Implementácia vizualizácie vychádza z návrhu v časti 4.5. Aplikácia je teda rozdelená do štyroch celkov, kde jednotlivé časti je možné minimalizovať pre dosiahnutie maximálnej plochy určenej k zobrazeniu výsledkov. Implementácia taktiež rešpektuje zásady responzívneho dizajnu, aby ju bolo možné využiť na zariadeniach akými sú napríklad tablety. Dôležitou vlastnosťou vizualizácie je rýchla adaptácia zobrazených informácií na požiadavky používateľa.

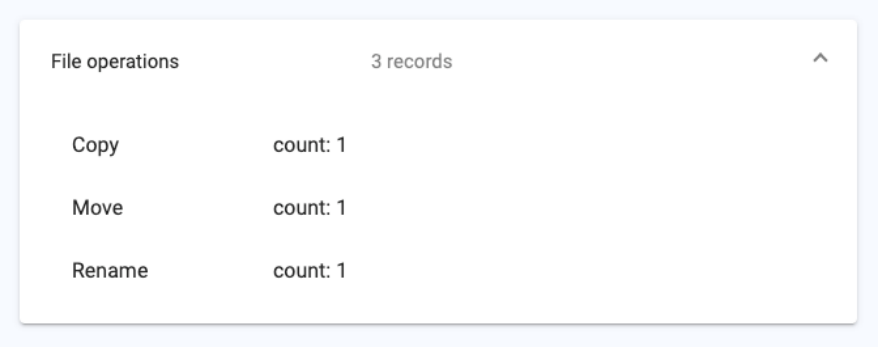
Pri prvotnom načítaní aplikácie sa inicializuje proces vytvárania skupinových profilov z celého rozsahu dostupných používateľských dát. Po tomto procese je používateľovi aplikácie umožnený výber špecifického používateľa a špecifický časový interval na základe ktorého bude prebiehať analýza anomálií v jeho chovaní. Po potvrdení zadaného výberu je na server vyslaná požiadavka pre analýzu aktivity vybraného používateľa.

Druhá časť aplikácie obsahuje možnosti výberu používateľa pre porovnanie. Tento používateľ je z rovnakého skupinového profilu, z akého je aj aktuálne vybraný používateľ. Obrázok 5.2 zobrazuje popisované časti aplikácie spolu s ďalšími časťami, ktoré sú minimalizované, no budú popisované v nasledujúcom texte.

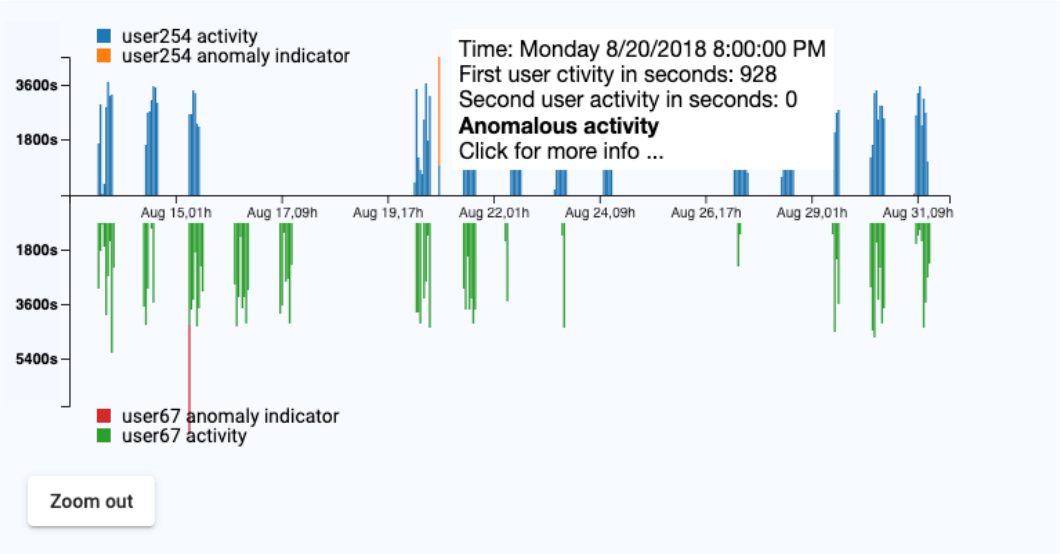
Jadrom vizualizačnej aplikácie je časová os zobrazujúca aktivitu používateľa v hodinových intervaloch. Časová os reprezentuje celkovú aktivitu, ktorú špecifický používateľ za daný časový interval vykonal. Rozsah zobrazených udalostí je možné interaktívne meniť za pomoci priblíženia alebo oddialenia časovej osy. Aplikácia podporuje zobrazenie dvoch časových rád, pre rýchle porovnanie aktivít oboch vybraných používateľov. Anomálne udalosti sú na tejto časovej ose zvýraznené a poskytujú detailnejší náhľad na aktivitu, ktorú používateľ v danom čase vykonal. Na obrázku 5.4 je možné vidieť ukážku aktivity zobrazenej na časovej osy, ktorá zobrazuje niektoré nájdene anomálne úseky. Anomálne úseky sú zvýraznené rozdielnou farbou a majú vždy jednotnú veľkosť. Po kliknutí na akúkoľvek oblasť grafu sa zobrazí detail daného úseku. Tento detail obsahuje porovnanie aktivity analyzovaného používateľa s aktivitou všetkých používateľov z rovnakého skupinového profilu. Toto porovnanie je implementované vo forme stĺpcového grafu, ktorý je zobrazený na obrázku 5.5. V detaile aktivity sa taktiež nachádza zoznam súborových operácií, ktoré boli vykonané v daný časový interval. Ukážka tohto informačného prvku sa nachádza na obrázku 5.3. Tento zoznam udalostí poskytuje pre používateľa tohto nástroja detailnejší kontext anomálnej udalosti. Táto časť môže byť do budúcna ďalej rozšírená inými informáciami o aktivite analyzovaného používateľa.



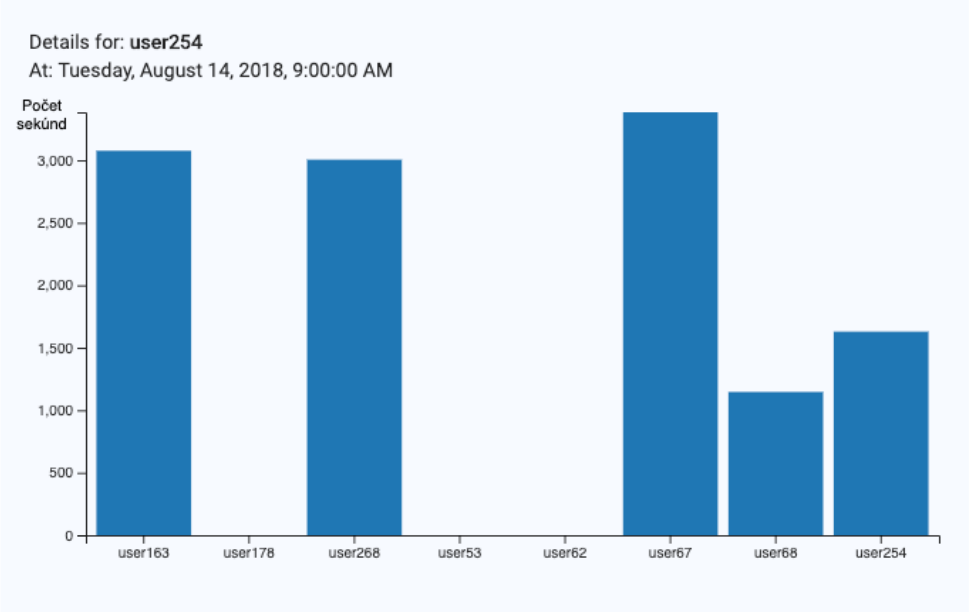
Obr. 5.2: Hlavný panel vizualizačnej aplikácie zobrazujúci sekciu filtrovania a sekciu výberu používateľa pre porovnanie. Po vybraní používateľa a časového intervalu v prvej sekcií je vytvorený zoznam používateľov z rovnakého skupinového profilu. Výber používateľa pre porovnanie ovplyvní ďalšie sekcie, ktoré porovnávajú aktivitu na základe analýzy časových radov.



Obr. 5.3: Obrázok predstavuje vizualizačný prvok pre analýzu aktivity v špecifickú hodinu. Na obrázku sa nachádza zoznam súborových operácií, ktoré boli vykonané v určitý časový interval.



Obr. 5.4: Obrázok približuje časť z vizualizačnej aplikácie. Jedná sa o vizualizáciu podozrivej aktivity a porovnanie aktivity s iným používateľom. Jednotlivé stĺpce v modrej a zelenej farbe reprezentujú aktivitu používateľov v špecifických hodinách. Oranžová a červená farba reprezentuje časovú udalosť, ktorá bola označená ako anomálna. Po kliknutí na akúkoľvek časť grafu sa zobrazí detail aktivity v danej hodine.



Obr. 5.5: Obrázok predstavuje vizualizačný prvok pre analýzu aktivity v špecifickú hodinu. Na obrázku sa nachádza graf porovnávajúci aktivitu analyzovaného používateľa s aktivitou ostatných používateľov z rovnakého skupinového profilu.

Kapitola 6

Testovanie

Testovanie vyvinutého systému pre analýzu používateľského chovania môžeme rozdeliť na dve oblasti. Prvou je overenie správnej tvorby skupinových profilov. Táto analýza združuje používateľov z podobným chovaním. Testovanie prebieha na základe pracovnej pozície, ktorú analyzovaný používatelia zastávajú. Druhá časť sa zameria na testovanie metódy pre detekciu anomálneho správania. Konkrétne sa jedná o anomálnu aktivitu, ktorá je vykonávaná v používateľovi neobvyklí čas. V nasledujúcej sekcií budú predstavené testovacie dáta.

6.1 Testovacie dáta

Testovacie dáta obsahujú aktivitu používateľov z prostredia stredne veľkej firmy s približne päťdesiatimi zamestnancami. Testovacie dáta sú anonymizované a zobrazujú aktivitu z oblasti kategorizovaných aplikácií a webových stránok. Konkrétne sa jedná o dvadsaťdeväť aplikačných kategórií medzi ktoré patrí napríklad:

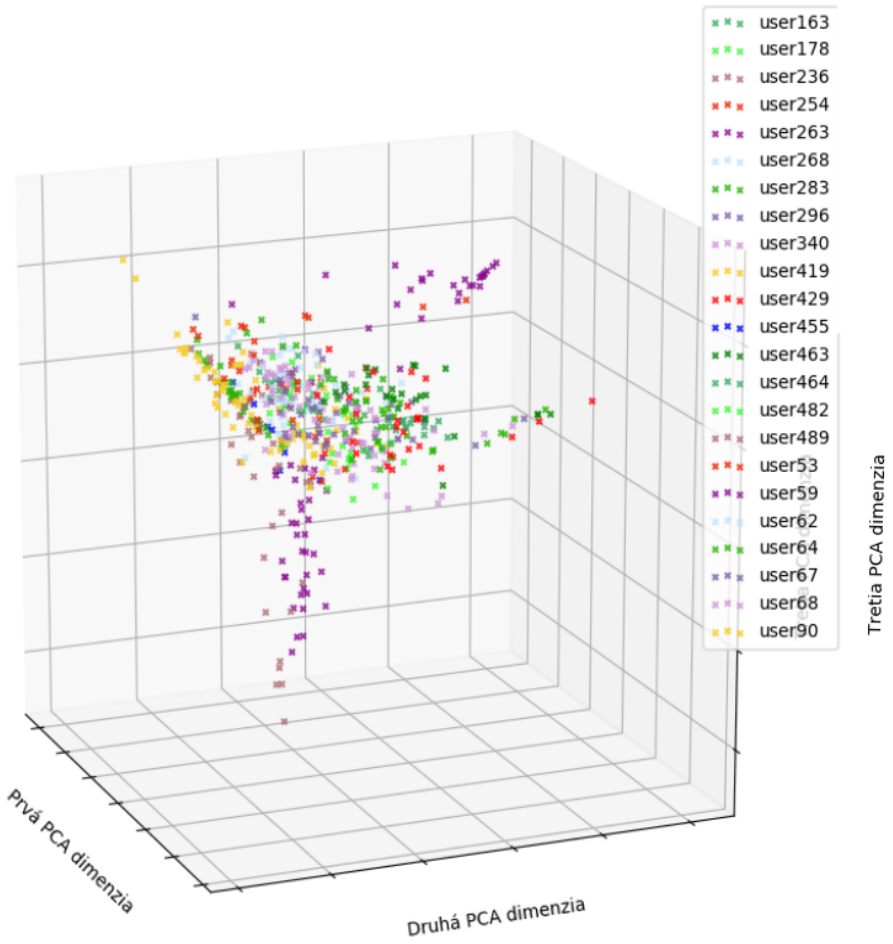
- emailový klient;
- biznis software;
- komunikačné aplikácie.

Webové kategórie obsahujú dvadsaťpäť kategórií, ktoré sú podobne štrukturované ako kategórie aplikačné. Celkovo dáta zaznamenávajú aktivitu dvadsiatich troch zamestnancov z celkovo siedmich oddelení. Práve informácia o priradenom pracovnom oddelení bude braná ako referenčný znak podobného správania v rámci skupiny používateľov. Na základe rovnakých dát boli pripravené vstupné dáta do oboch častí systému. Prvá časť obsahuje jednodenne agregovanú aktivitu pre tvorbu používateľských profilov. Druhá časť obsahuje hodinovo agregovanú celkovú aktivitu pre konkrétneho používateľa. Cela množina dát popisuje aktivitu používateľov v časovom rozmedzí desiatich týždňov.

6.2 Tvorba skupinových profilov

Prvým krokom pri tvorbe skupinových profilov je využitie niektorého z algoritmov pre redukciu dimenzionality. Keďže algoritmus nie je stavaný na určitú štruktúru vstupných dát, nie je ani známa dimenzionalita vstupných dát. Z toho dôvodu v tomto prípade využijeme redukciu na polovičný počet dimenzií, pre ukážku chovania algoritmu.

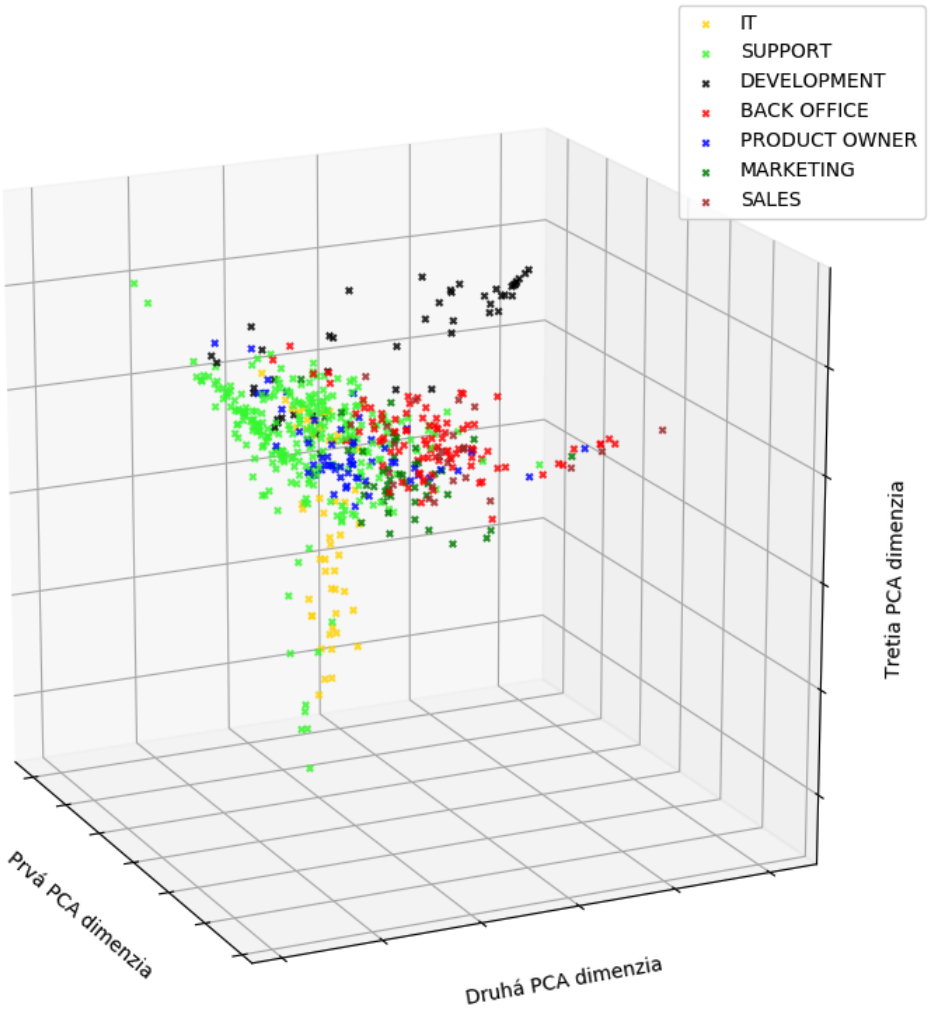
Redukcia dimenzionality taktiež umožňuje čiastočne vizualizovať vstupné dáta. Obrázok 6.1 prezentuje prvé tri dimenzie s najväčšou variáciou. Taktiež ďalšie ukážky reprezentujúce aktivitu používateľa budú vizualizované na základe prvých troch dimenzií. Je ale potrebné si uvedomiť, že daný model obsahuje omnoho viac dimenzií. Na obrázku je vidieť, že niektoré inštancie dát utvárajú skupiny. Na základe týchto zhlukov bude vytvorený skupinový profil.



Obr. 6.1: Obrázok zobrazuje tri najvýraznejšie dimenzie dátového modelu aktivity používateľov. Každý dátový bod reprezentuje aktivitu používateľa v jeden konkrétny deň. Na obrázku je vidieť, že body rovnakej farby utvárajú zhluky a tým reprezentujú určité stabilné chovanie používateľa. Nad týmto modelom bude následne prebiehať zhlukovanie pre detekciu používateľov s rovnakým chovaním.

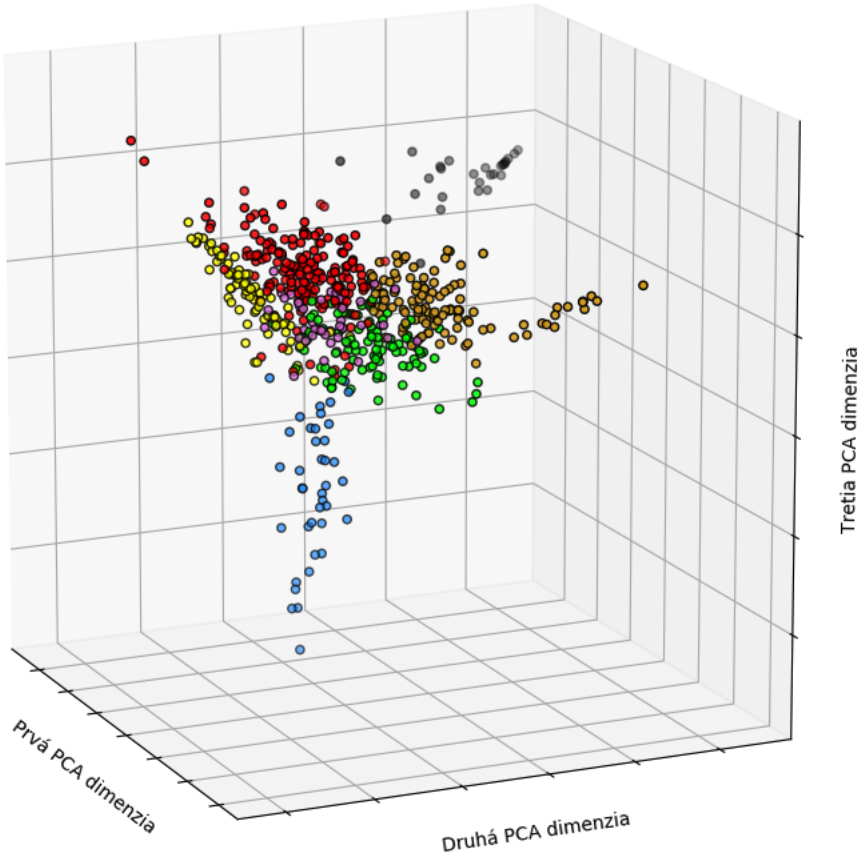
V našom konkrétnom prípade testovacích dát je dostupné referenčné ohodnotenie skupín používateľov s podobným správaním. Toto ohodnotenie je vo forme pracovného oddelenia, do ktorého jednotlivý používatelia patria. Obrázok 6.2 prezentuje reálne priradenie pracovného oddelenia jednotlivým používateľom. Na obrázku sú vidieť referenčné zhluky, ktoré hľadáme pomocou zhlukovacích algoritmov.

Pre segmentáciu dátových inštancií je potrebné na začiatku určiť počet zhlukov, ktoré budú vytvorené na základe algoritmu *k-means*. V tomto prípade bolo pomocou metódy



Obr. 6.2: Na obrázku sú zvýraznené rovnakou farbou dátové inštancie používateľov, ktorý patria do rovnakého pracovného oddelenia. Jedná sa o referenčnú segmentáciu a k takémuto ohodnoteniu sa snažíme priblížiť pomocou zhlukovacích algoritmov, ktoré pracujú vo viacrozmernom priestore.

siluet vypočítané, že optimálny počet zhlukov je sedem. Následne bol každej inštancií priradený špecifický zhluk do ktorého patrí. Na obrázku 6.3 je zobrazený výsledok segmentácie jednotlivých dátových inštancií. Porovnaním s referenčným vzorom z obrázku 6.2 vidíme, že segmentácia sa podobá v mnohých oblastiach. Jeden z rozdielov je vidieť v prípade, že sa zameriame na zelenú skupinu dát z referenčného vzoru (support). V tomto prípade bola segmentácia tohto zhluku rozdelená na dve skupiny reprezentované žltým a červeným zhlukom. Je pravdepodobné, že aj napriek tomu, že daný používatelia zastavajú rovnakú pozíciu, ich aktivita sa rozdeľuje do dvoch skupín. Obe skupiny ale zahŕňajú hlavne používateľov z rovnakého pracovného oddelenia. Výsledok segmentácie je zhrnutý v tabuľke 6.1. Na výsledkoch segmentácie je vidieť, že niektoré zhluky obsahujú používateľov z viacerých pracovných oblastí. Nie je presne jasné do akej mieri je toto správanie v realite podobné, no je nutné dodať, že oddelenie podpory je v tomto prípade z prostredia technickej podpory.



Obr. 6.3: Segmentácia jednotlivých dátových inštancií pre vytvorenie skupinových profilov. Na obrázku je vidieť, že segmentácia dosiahla podobné výsledky ako referenčný vzor z obrázku 6.2.

Aj preto sa v niektorých dňoch prekrýva s aktivitou iných pozícií. Ďalšiu zaujímavú situáciu tvorí oddelenie vývoja (development), kde používateľ v zhluku päť je správne oddelený od iných používateľov. Ďalší používateľ z tohto oddelenia je ale chybne priradený do zhluku s technickou podporou. Táto situácia nastala z toho dôvodu, že v daný interval, v ktorom bola uskutočnená analýza, nebola aktivita používateľa na základe vybraných techník rozlíšiteľná od používateľov technickej podpory. Po bližšej analýze konkrétneho používateľa bolo zistené, že daný používateľ, aj keď je priradený do oddelenia vývoja, nezastáva pracovnú pozíciu vývojára. Mimo iné, obsahovala aktivita daného zamestnanca tvorbu dokumentácie a tvorbu testovacích scenárov. Jeho aktivita na pracovnom zariadení je teda aj reálne podobnejšia s pozíciami oddelenia technickej podpory. Podobný scenár sa nachádza aj pri zhluku jedna, kedy ale dané pracovné pozície úzko súvisia a tento zhluk je teda považovaný za správny.

Pre ohodnotenie presnosti vytvorených zhlukov využijeme metódu harmonického priemeru značenú ako *F-measure*. Táto metóda podporuje rozdielnu váhu chyby v prípade falošných pozitív a v prípade pravdivých negatív [11]. Metóda v kontexte ohodnocovania vytvorených zhlukov pracuje so všetkými možnými dvojicami prvkov vo vstupnej množine prvkov. V našom prípade sú to jednotlivý používatelia. Pre definovanie tejto metódy je potrebné zadefinovať nasledujúce prvky metódy:

UserId	Cluster	Department
user455	0	SUPPORT
user236	0	SUPPORT
user64	0	SUPPORT
user419	0	SUPPORT
user283	1	BACK OFFICE
user429	1	SALES
user463	1	BACK OFFICE
user464	1	BACK OFFICE
user59	2	IT
user489	2	SUPPORT
user263	3	DEVELOPMENT
user340	4	MARKETING
user482	4	MARKETING

UserId	Cluster	Department
user67	5	SUPPORT
user62	5	SUPPORT
user53	5	DEVELOPMENT
user163	5	IT
user268	5	SUPPORT
user254	5	SUPPORT
user178	5	SUPPORT
user68	5	SUPPORT
user296	6	PRODUCT OWNER
user90	6	PRODUCT OWNER

Tabuľka 6.1: Vytvorené zhluky na základe aktivity jednotlivých používateľov.

- celkový počet možných dvojíc N;
- pravdivé pozitíva TP - stav, kedy sú dvaja používatelia s podobným chovaním priradený do rovnakého zhluku;
- pravdivé negatíva TN - stav, kedy sú dvaja používatelia s podobným chovaním priradený do odlišného zhluku;
- falošné pozitíva FP - stav, kedy sú dvaja používatelia s rozdielnym chovaním priradený do rovnakého zhluku;
- falošné negatíva FN - stav, kedy sú dvaja používatelia s rozdielnym chovaním priradený do odlišného zhluku.

Následne sú všetky možné dvojice používateľov, ktorých je N, rozdelený do uvedených prvkov metódy. Výsledok tohto rozdelenia je zobrazený v tabuľke 6.2.

	Rovnaký zhluk	Rozdielny zhluk
Dvojica používateľov s rovnakým chovaním	TP=29	FN=36
Dvojica používateľov s rozdielnym chovaním	FP=14	TN=174

Tabuľka 6.2: Vyhodnotenie stavov, medzi všetkými dvojicami používateľov na základe vytvorených zhlukov z tabuľky 6.1.

Následne je z týchto hodnôt vypočítaná presnosť P a senzitivita R. Definícia týchto hodnôt je zobrazená v rovnici 6.2 a 6.1. Tieto hodnoty budú následne použité pre výpočet hodnoty *F-measure*, podľa rovnice 6.3. Taktiež bude definovaná hodnota *Accuracy*, ktorá

	Pôvodné referenčné rozdelenie	Upravené referenčné rozdelenie
F-measure	0.536	0.724
Accuracy	81%	91%

Tabuľka 6.3: Tabuľka obsahuje vyhodnotenie presnosti vytvorených skupinových profilov. Výsledky pre pôvodné referenčné rozdelenie uvažujú rozdelenie zobrazené na obrázku 6.2. Hodnoty v poslednom stĺpci tabuľky uvažujú upravené referenčné rozdelenie, kde oddelenie podpory obsahuje dve rozdielne skupiny používateľov. Toto rozdelenie viac zodpovedá reálnemu rozdeleniu pracovných pozícií a je ho možné vidieť na obrázku 6.2 ako dva podobne veľké zhluky jednej farby.

vyjadruje prestnosť správne priradených zhlukov, nezávisle na počte nesprávne priradených zhlukov. Táto hodnota je definovaná rovnicou 6.4.

$$P = \frac{TP}{TP + FP} \tag{6.1}$$

$$R = \frac{TP}{TP + FN} \tag{6.2}$$

$$F\text{-}measure = \frac{2PR}{P + R} \tag{6.3}$$

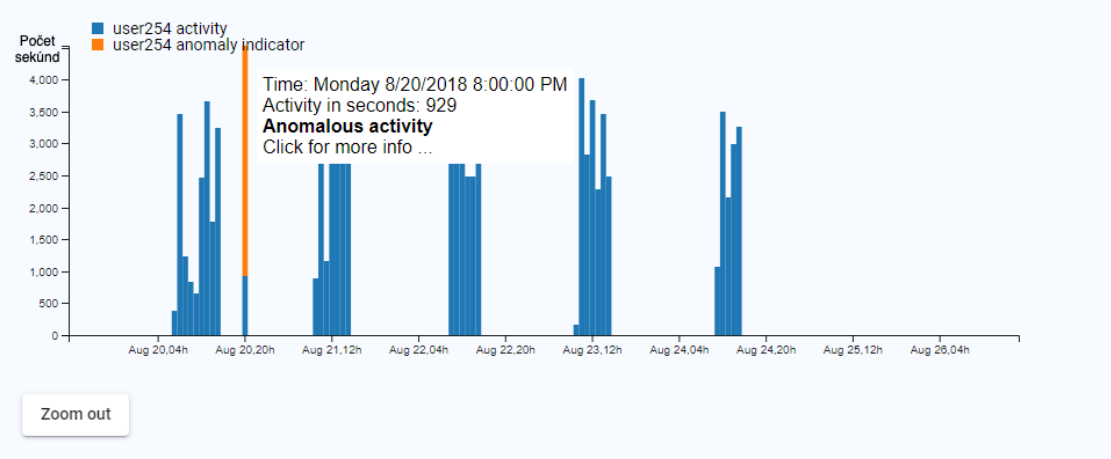
$$Accuracy = \frac{TP + TN}{N} \tag{6.4}$$

Tabuľka 6.3 porovnáva získanú *F-measure* hodnotu v prípade skupiny používateľov s referenčným ohodnotením podľa obrázku 6.1 a skupiny s upraveným referenčným ohodnotením. Toto nové ohodnotenie spočíva v bližšom rozdelení oddelenia podpory na dve menšie oddelenia, na základe ich reálnych odlišností. Tým je docielené spresnenie referenčného rozdelenia používateľov a táto zmena je reflektovaná aj do vyhodnotenia presnosti algoritmu. Vyhodnotenie presnosti zhlukovania bude použité pre porovnanie presnosti aktuálneho spôsobu vytvárania skupinových profilov s možnou implementáciou nadväzujúcou na túto prácu.

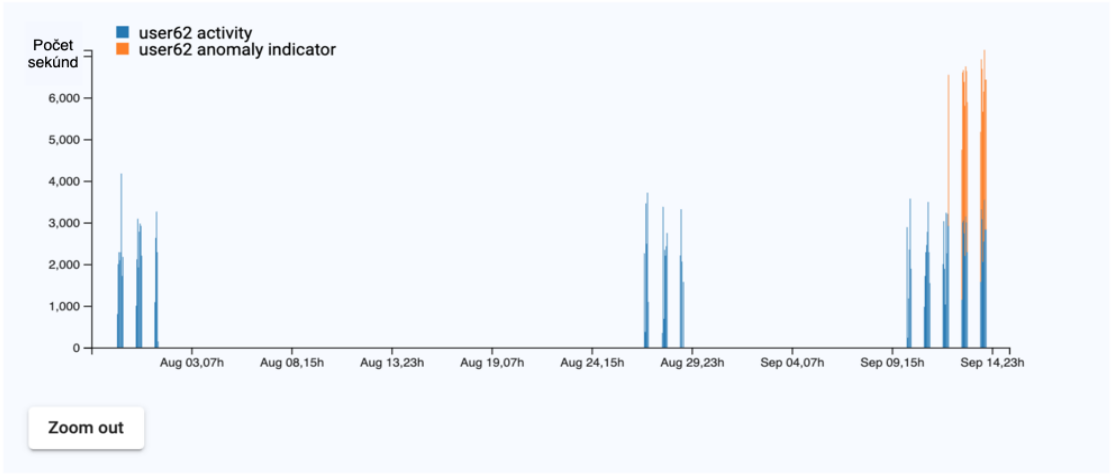
6.3 Detekcia anomálneho správania

Pri detekcií anomálneho správania budeme vychádzať z predpripravených scenárov, na ktoré má detekcia reagovať, a ktorých návrh bol predstavený v časti 4.4. Jedná sa teda o správnu detekciu aktivity používateľa v jemu neobvyklých hodinách. Pre testovacie účely bol vybraný časový interval v rozmedzí piatich týždňov. Keďže neexistujú referenčné výsledky pre detekciu anomálnej aktivity, bude vyhodnotené, či detekované anomálne udalostí boli validné. Ďalšou úlohou testovania je odhalenie chybných reakcií na vzniknuté správanie.

Celkovo bolo analyzovaných dvadsaťtri používateľov. Pri detekcii anomálií v zadanom intervale, bolo nájdených niekoľko anomálnych udalostí. Každá z týchto udalostí predstavovala validnu anomálnu aktivitu v nepracovný deň a vo večerných hodinách. Na základe

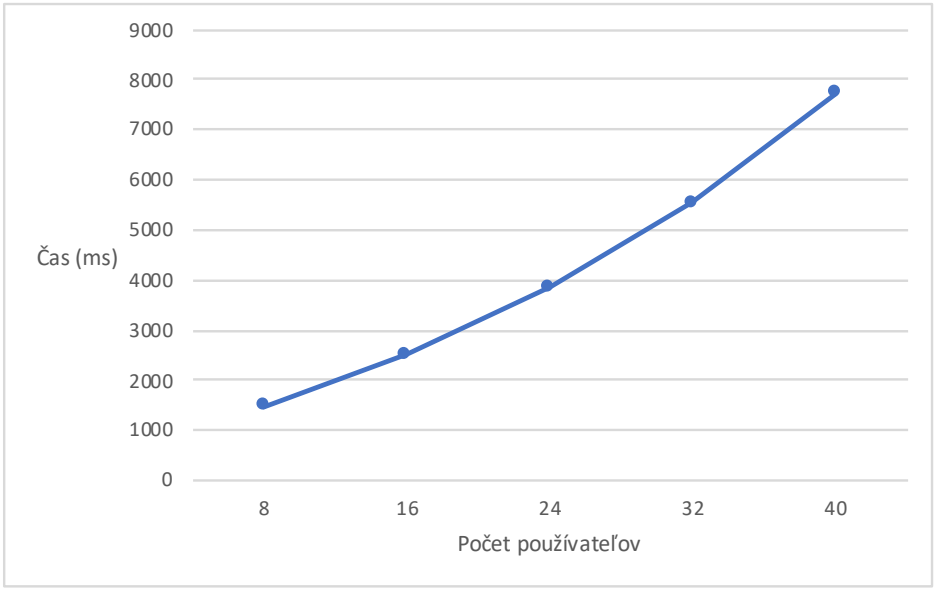


Obr. 6.4: Obrázok reprezentujúci správne detekovanú anomálnu udalosť v reálnych dátach. Jedná sa o anomálnu aktivitu vo večerných hodinách.



Obr. 6.5: Obrázok reprezentujúci chybu pri detekovaní anomálnej aktivity. Chyba vychádza z vlastnosti nastavenia algoritmu analýzy časových rád. Udalosti boli označené z dôvodu, že používateľ nevykonal žiadnu aktivitu dlhšie ako jeden mesiac. Tým sa stratila náväznosť na používateľovu bežnú pracovnú dochádzku a systém upozornil na neočakávanú aktivitu.

výsledkov algoritmu a na základe detailu anomálnej udalosti, ktorý popisuje vykonané súborové operácie, bolo možné vyhľadať detailnejšie informácie o tejto aktivite. Pri analýze používateľa s pravidelnou bežnou pracovnou dochádzkov, algoritmus nevykázal takmer žiadne falošné pozitíva. Falošné pozitíva nastali u dvoch typov prípadov. Prvým je analýza používateľa s riedkou aktivitou. V tomto prípade môže ísť napríklad o externého pracovníka u ktorého nie je možné adaptovať systém na pravidelnú aktivitu. Druhý prípad je zobrazený na obrázku 6.5. Táto chybná interpretácia vychádza z nastavenia algoritmu pre analýzu časových rád. Udalosti boli označené z dôvodu, že používateľ nevykonal žiadnu aktivitu dlhšie ako jeden mesiac. Z tohto dôvodu sa stratila náväznosť na používateľovu bežnú pracovnú dochádzku. Riešenie takéhoto stavu môže spočívať v prenastavení počiatočného bodu, od



Obr. 6.6: Obrázok popisuje vyhodnotenie časovej náročnosti tvorby skupinových profilov v závislosti na počte používateľov.

ktorého prebieha analýza na bod, od ktorého sa predpokladá pravidelná dochádzka používateľa.

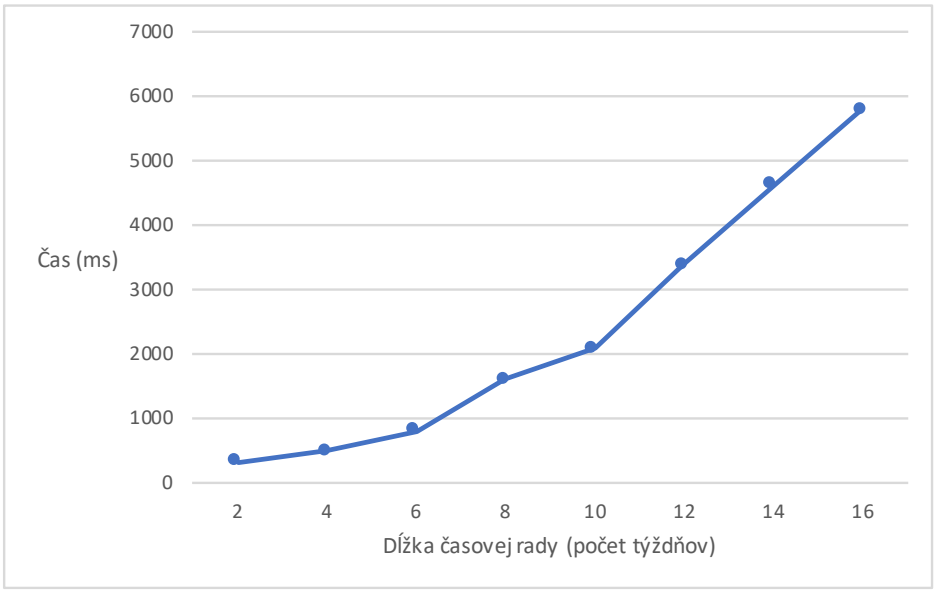
6.4 Časová náročnosť

Okrem testovania algoritmov na správnosť tvorby skupinových profilov a na správnu detekciu anomálií, je tiež potrebné vyhodnotiť časovú náročnosť týchto úloh. Táto náročnosť sa počíta vzhľadom k počtu používateľov a k dĺžke analyzovanej časovej rady. Je to z dôvodu, že návrh aplikácie počíta s použitím vytvorenej aplikácie v rôzne veľkých firemných prostrediach. Jednotlivé výsledky testov reprezentujú priemernú časovú dobu výpočtu z troch meraní.

Časová náročnosť tvorby skupinových profilov bude vyhodnotená vzhľadom k počtu používateľov. Pri vyhodnotení časovej náročnosti sa použili modely používateľov, obsahujúce aktivitu z rozmedzia štyroch mesiacov. Obrázok 6.6 zobrazuje nameranú priemernú časovú náročnosť pri rozdielnom počte používateľov.

Pri testovaní časovej náročnosti detekcie anomálií bola detekcia uskutočnená na jednom používateľovi so štandardnou pracovnou dobou. Výsledný čas je vyhodnotení vzhľadom k meniacej sa dĺžke analyzovanej časovej rady. Obrázok 6.7 zobrazuje priemerný čas výpočtu pri rôznej dĺžke časovej rady.

Výsledky časovej analýzy ukazujú, že aplikáciu je možné používať pri relatívne veľkom množstve používateľov a pre dostatočne dlhé časové obdobie tak, aby odpovedala na používateľské požiadavky v rozumnom čase (jednotky až desiatky sekúnd). Táto vlastnosť je dôležitá predovšetkým preto, že dobrá interakcia s aplikáciou bola jednou z požiadavok pri návrhu.



Obr. 6.7: Obrázok popisuje vyhodnotenie časovej náročnosti detekcie anomálnej aktivity pre konkrétneho používateľa v závislosti na dĺžke analyzovanej časovej rady.

Kapitola 7

Záver

Práca vo svojom úvode rozoberá problematiku analýzy používateľského chovania a obsahuje motiváciu a dôvody pre vznik vyvíjanej aplikácie. Následne popisuje aktuálny stav v tejto oblasti a poskytuje základný prehľad už existujúcich aplikácií, ktoré čiastočne ovplyvnili návrh vyvíjanej aplikácie.

Jednou z hlavných častí práce je široký prehľad metód a modelov, ktoré je možné použiť pri dátovej analýze používateľskej aktivity. Následne boli niektoré z popisovaných metód vybrané a použité pri analýze používateľskej aktivity. Pre potreby správnej analýzy bolo potrebné správne zadefinovať štruktúru vstupných dát. Pre dosiahnutie lepších výsledkov bolo nutné tieto dáta vhodne predspracovať.

Pred samotnou implementáciou riešenia bol uskutočnený návrh používateľských a skupinových profilov. Následne bol vytvorený návrh pre porovnanie a zhlukovanie týchto modelov a spôsob detekcie anomálnej aktivity. Ná základe vytvorených návrhov bol vyvinutý nástroj pre analýzu anomálnej aktivity používateľa. Tento nástroj správne funguje pre detekciu určitých typov anomálnej aktivity. Jedná sa hlavne o odlišnosti v správaní na základe predošlej aktivity. Používateľ tohto nástroja môže jednoducho odhaliť anomálnu aktivitu používateľa a následne ju porovnať s inými špecificky vybranými používateľmi. Tento výber určuje aplikácia na základe podobnosti chovania, kde túto podobnosť modeluje skupinový profil.

Záverečná kapitola obsahuje testovanie nad reálnymi dátami. V rámci tohto testovania je overená správnosť vytvorených skupinových profilov a správnosť detekcie anomálnej aktivity pri rôznych scenároch. Následne bola vyhodnotená presnosť vytvorených zhlukov a taktiež boli uvedené bližšie informácie k detekovaným anomáliám. Záver kapitoly obsahuje vyhodnotenie časovej náročnosti jednotlivých častí analýzy používateľského chovania.

Celkovo aplikácia splňuje požiadavky, ktoré boli na ňu pri návrhu kladené a je ju možné použiť pre analýzu reálnych dát. Funkčnosť aplikácie je po získaní lepšie označených dát do budúcna možné rozšíriť, využitím neurónových sieti. Ide napríklad o metódu *LSTM*, pomocou ktorej je možné vykonávať detekciu anomálneho správania s využitím vlastností dlhodobej pamäte [9].

Výsledky práce je možné využiť dvoma spôsobmi. Prvý spôsob je využitie vlastnosti, že vytvorená webová aplikácie je vo forme modulu. Takáto forma dovoľuje jej jednoduché začlenenie do už existujúcej webovej aplikácie. Druhý spôsob spočíva vo využití vytvorených metód pre analýzu používateľského chovania pomocou serverového API. To umožňuje využiť výsledky analýzy anomálnych udalostí a informácie o podobnosti chovania jednotlivých používateľov v už existujúcom riešení pre únik citlivých informácií z firemného prostredia.

Zoznam termínov a skratiek

ARIMA *Autoregressive integrated moving average*. 19, 20

BRF *Gaussian radius basis function*. 28, 30

CSV *Comma-separated values*. 42, 43, 44

D3.js knižnica pre tvorbu interaktívnych vizualizácií na základe vstupných dát. 43

HTTP *Hypertext Transfer Protocol*. 6

KNN *K-nearest-neighbors*. 13

KPCA *Kernel Principal component analysis*. 28, 30, 35, 44

LAD *Least absolute deviations*. 22

LCM *Local cost matrix*. 22, 23

LSTM *Long Short Term Memory*. 13, 24, 25, 58

PCA *Principal component analysis*. 11, 14, 24, 26, 27, 28, 29, 30, 35, 39, 44

RobustSTL *A Robust Seasonal-Trend Decomposition Algorithm for Long Time Series*. 21, 43, 45

SIEM *Security Information and Event Management*. 3

STL *Seasonal Trend decomposition*. 19, 20, 21

SVM *Support-vector machine*. 13

clickstreams záznamy o udalostiach, ktoré boli vyvolané vstupom od používateľa. 6

faktorová analýza je štatická metóda, slúži na vysvetlenie rozptylu premenných. 5

k-means zhlukovací iteračný algoritmus. 5, 16, 36, 44, 50

učenie bez učiteľa metóda učenia na základe štruktúry vstupných dát, nevykonáva trénovaciu fázu. 6

Literatúra

[1] Angeletou, S.; Rowe, M.; Alani, H.: Modelling and Analysis of User Behaviour in Online Communities. In *The Semantic Web – ISWC 2011*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ISBN 978-3-642-25073-6, s. 35–50.
URL https://link.springer.com/chapter/10.1007/978-3-642-25073-6_3

[2] Anshu Bhardwaj: Data Preprocessing Techniques for Data Mining. Indian Agricultural Statistics Research Institute, [Online; navštívené 10.1.2019].
URL http://iasri.res.in/ebook/win_school_aa/notes/Data_Preprocessing.pdf

[3] Cerullo, G.; Formicola, V.; Iamiglio, P.; aj.: Critical Infrastructure Protection: having SIEM technology cope with network heterogeneity. *Computing Research Repository (CoRR)*, ročník abs/1404.7563, 2014.
URL http://arxiv.org/abs/1404.7563

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

[5] Derksen, L.: Visualising high-dimensional datasets using PCA and t-SNE in Python. Towards Data Science, 2016, [Online; navštívené 7.5.2019].
URL https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b

[6] Dr. Zdravko Markov: Data Preprocessing. Central Connecticut State University, 2019, [Online; navštívené 7.4.2019].
URL http://www.cs.ccsu.edu/~markov/ccsu_courses/DataMining-3.html

[7] Fokin, D.; Hagrot, J.: *Constructing decision trees for user behavior prediction in the online consumer market*. Bakalárska práca, KTH Royal Institute of Technology, Stockholm, Sweden, 2016.
URL http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A927446&dswid=1299

[8] Hyeok, K.; Cholyong, J.; Unhyok, R.: Implementation of Association Rule Mining for Network Intrusion Detection. *Computing Research Repository (CoRR)*, ročník abs/1601.05335, 2016, withdrawn.
URL http://arxiv.org/abs/1601.05335

[9] Malhotra, P.; Vig, L.; Shroff, G.; aj.: Long short term memory networks for anomaly detection in time series. In *ESANN 2015 proceedings, European Symposium on*

Artificial Neural Networks, Computational Intelligence and Machine Learning, Presses universitaires de Louvain, 2015, ISBN 978-287587014-8, str. 89.
URL https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2015-56.pdf

[10] Malik, U.: Association Rule Mining via Apriori Algorithm in Python. Blog post, StackAbuse, 2018, [Online; navštívené 15.3.2019].
URL https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/

[11] Manning, C. D.; Raghavan, P.; Schütze, H.: *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008, ISBN 0521865719, 9780521865715.

[12] Olah, C.: Understanding LSTM Networks. Blog post, 2015, [Online; navštívené 19.4.2019].
URL http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[13] Rai, K.; Devi, M. S.; Guleria, A.: Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*, ročník 7, č. 4, 2016: str. 2828, ISSN 0975-0290.
URL http://www.ijana.in/papers/V7I4-7.pdf

[14] Raschka, S.: Kernel tricks and nonlinear dimensionality reduction via RBF kernel PCA. September 2014, [Online; navštívené 27.3.2019].
URL https://sebastianraschka.com/Articles/2014_kernel_pca.html

[15] Rawat, R.: *User behaviour modelling in a multi-dimensional environment for personalization and recommendation*. Dizertačná práca, Queensland University of Technology, 2010.
URL https://eprints.qut.edu.au/48135/

[16] Renée E. Etoty and Robert F. Erbacher: A Survey of Visualization Tools Assessed for Anomaly-Based Intrusion Detection Analysis. Computational and Information Sciences Directorate, ARL, 2014.
URL https://apps.dtic.mil/docs/citations/ADA601590

[17] Richards, K.: User behavior analytics leads the security analytics charge. [Online; navštívené 30.12.2018].
URL https://searchsecurity.techtarget.com/feature/User-behavior-analytics-leads-the-security-analytics-charge

[18] Sardá-Espinosa, A.: Comparing time-series clustering algorithms in R using the dtwclust package. *Vienna: R Development Core Team*, 2017.
URL https://www.semanticscholar.org/paper/Comparing-Time-Series-Clustering-Algorithms-in-R-Sarda-Espinosa/ceabb44c8b3606decd791ae7da50e54401a0e9f5

[19] Seif, G.: A Guide to Decision Trees for Machine Learning and Data Science. Blog post, Towards Data Science, 2018, [Online; navštívené 15.3.2019].
URL https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956

[20] Seif, G.: The 5 Clustering Algorithms Data Scientists Need to Know. Towards Data Science, 2018, [Online; navštívené 10.1.2019].
URL https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

[21] Shlens, J.: A Tutorial on Principal Component Analysis. *Computing Research Repository (CoRR)*, ročník abs/1404.1100, 2014.
URL http://arxiv.org/abs/1404.1100

[22] Singh, A.; Yadav, A.; Rana, A.: K-means with Three different Distance Metrics. *International Journal of Computer Applications*, ročník 67, 04 2013: s. 13–17, doi:10.5120/11430-6785.
URL https://www.ijcaonline.org/archives/volume67/number10/11430-6785

[23] StatSoft, I.: How To Identify Patterns in Time Series Data: Time Series Analysis. Electronic Statistics Textbook. Tulsa, 2013, [Online; navštívené 10.1.2019].
URL http://www.statsoft.com/Textbook/Time-Series-Analysis#index

[24] Wang, G.; Zhang, X.; Tang, S.; aj.: Unsupervised Clickstream Clustering for User Behavior Analysis. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2016, ISBN 978-1-4503-3362-7, s. 225–236, doi:10.1145/2858036.2858107.

[25] Wen, Q.; Gao, J.; Song, X.; aj.: RobustSTL: A Robust Seasonal-Trend Decomposition Algorithm for Long Time Series. *Computing Research Repository (CoRR)*, ročník abs/1812.01767, 2018.
URL http://arxiv.org/abs/1812.01767

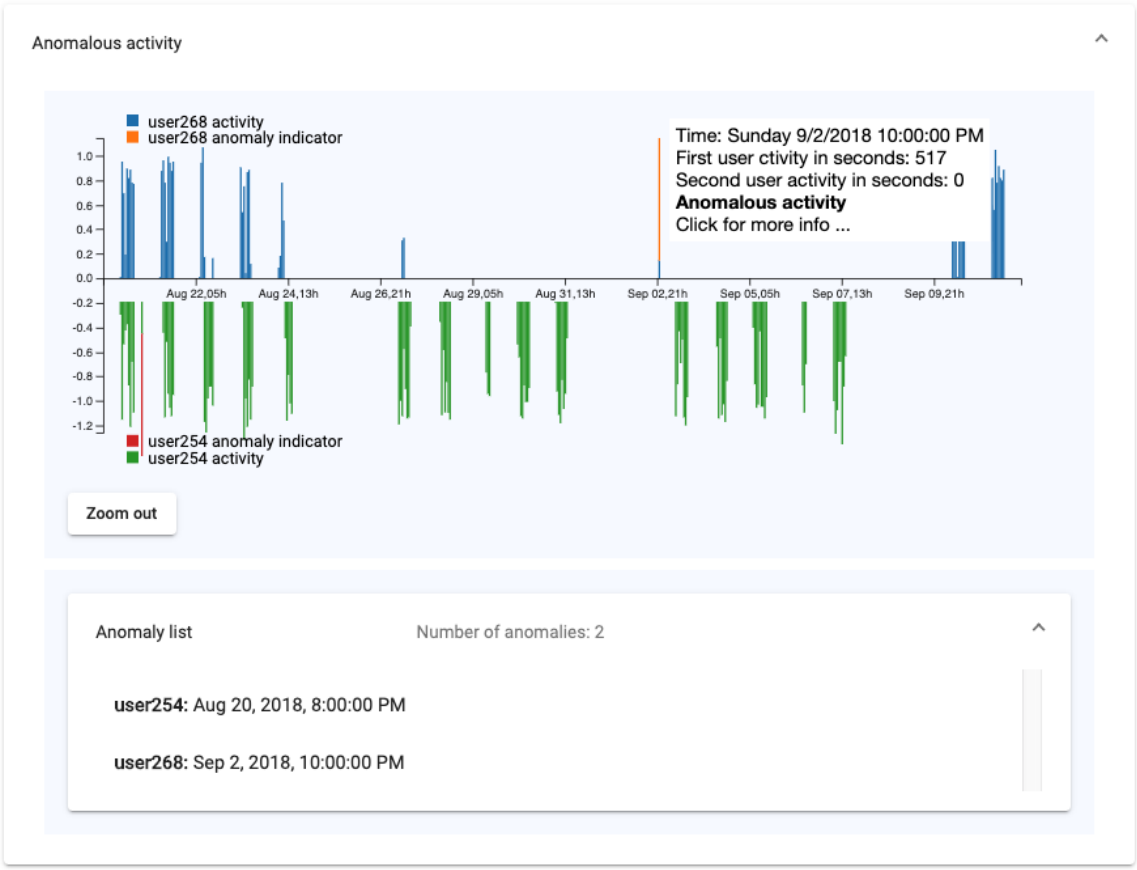
[26] Zhang, M.; Wang, Y.; Chai, J.: Review of User Behavior Analysis Based on Big Data: Method and Application. In *Conference: 2015 International Conference on Advances in Mechanical Engineering and Industrial Informatics*, Atlantis Press, 01 2015, ISBN 978-94-62520-69-1, doi:10.2991/ameii-15.2015.17.

[27] Zimek, A.; Schubert, E.; Kriegel, H.-P.: A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, ročník 5, č. 5, 2012: s. 363–387, doi:10.1002/sam.11161.
URL https://onlinelibrary.wiley.com/doi/10.1002/sam.11161

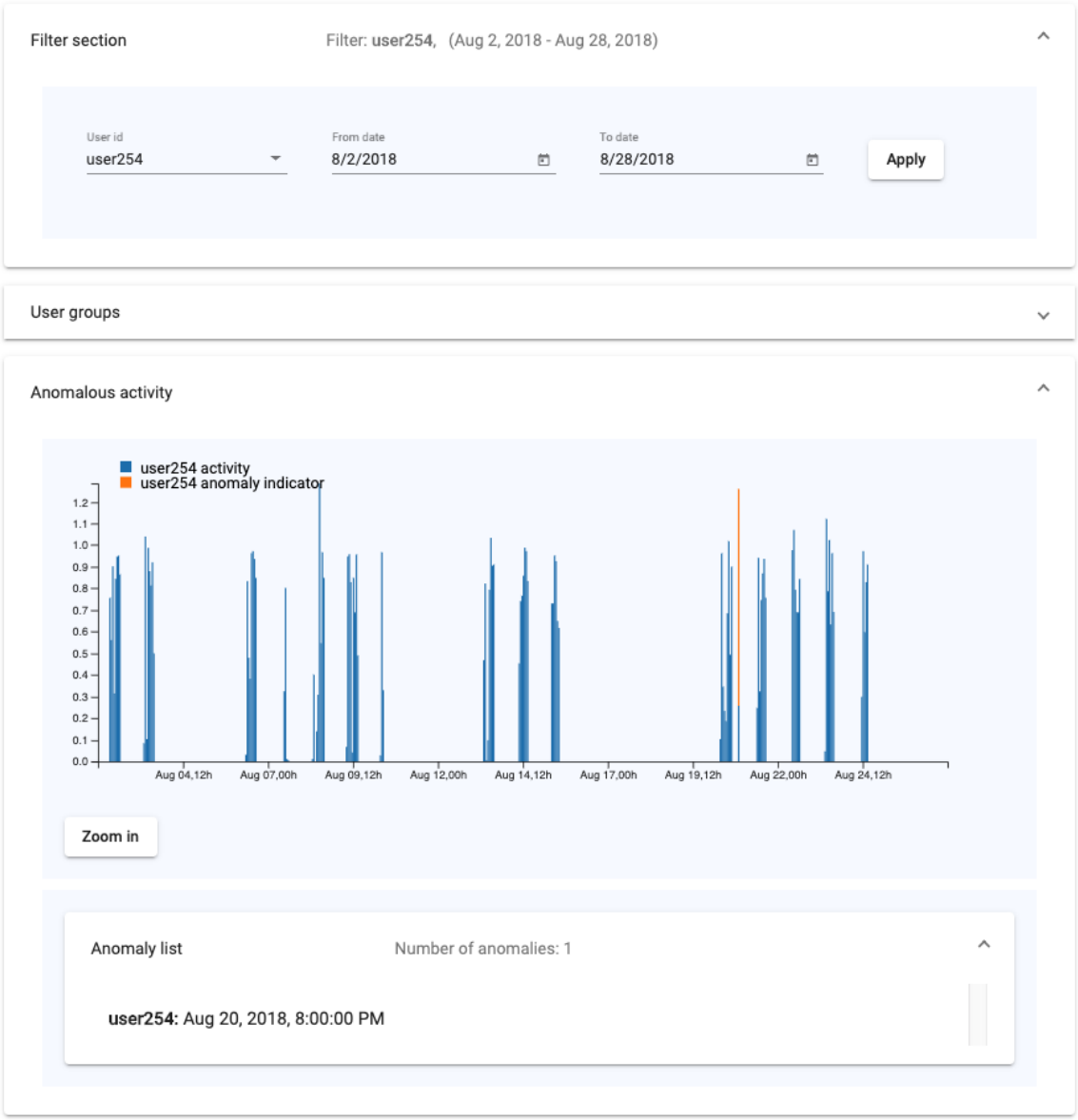
Príloha A

Webová aplikácia

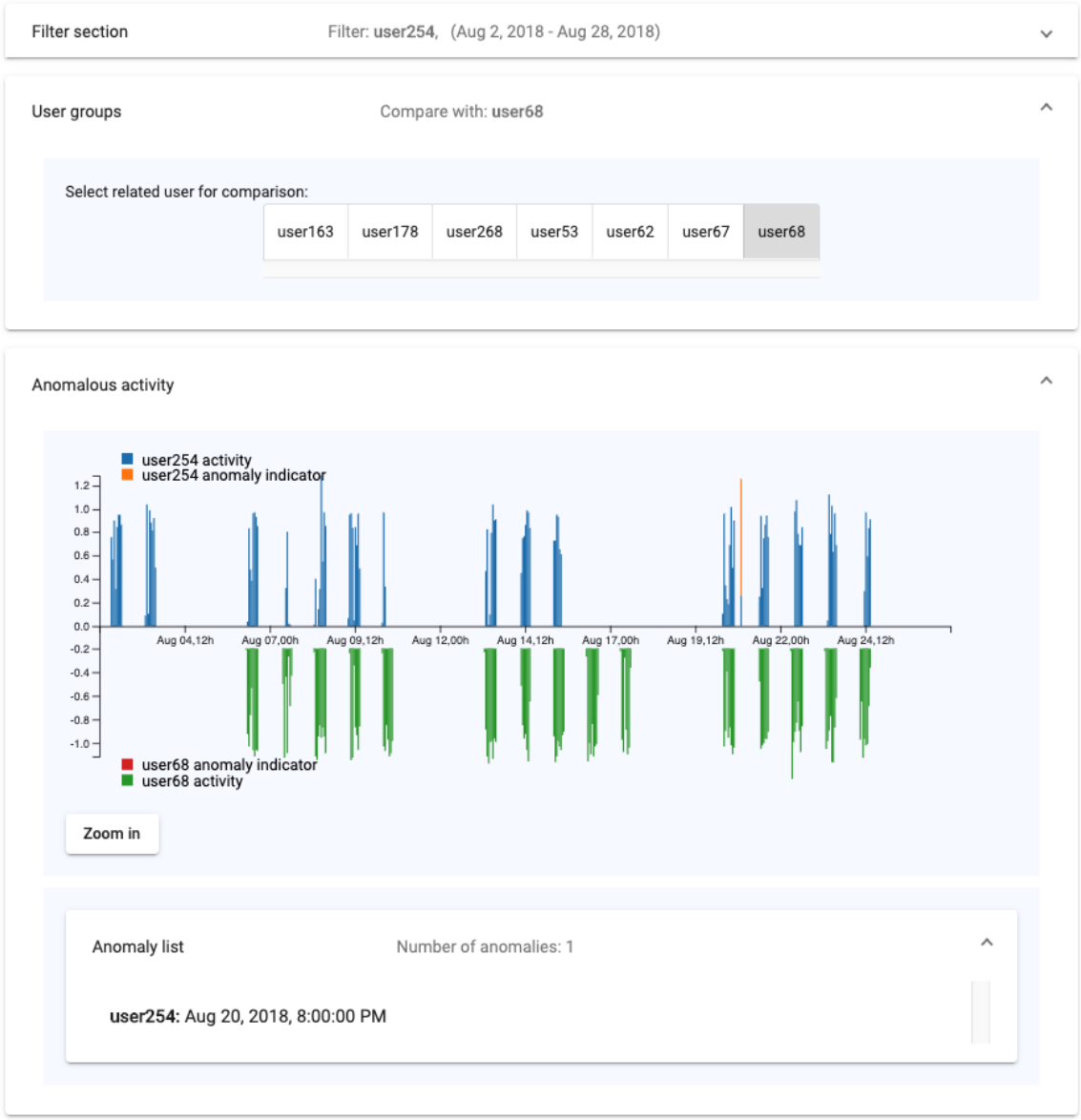
Táto príloha obsahuje ukážky z vytvoreného vizualizačného nástroju pre analýzu anomálií v používateľskom chovaní. Nástroj je implementovaný vo forme webovej aplikácie.



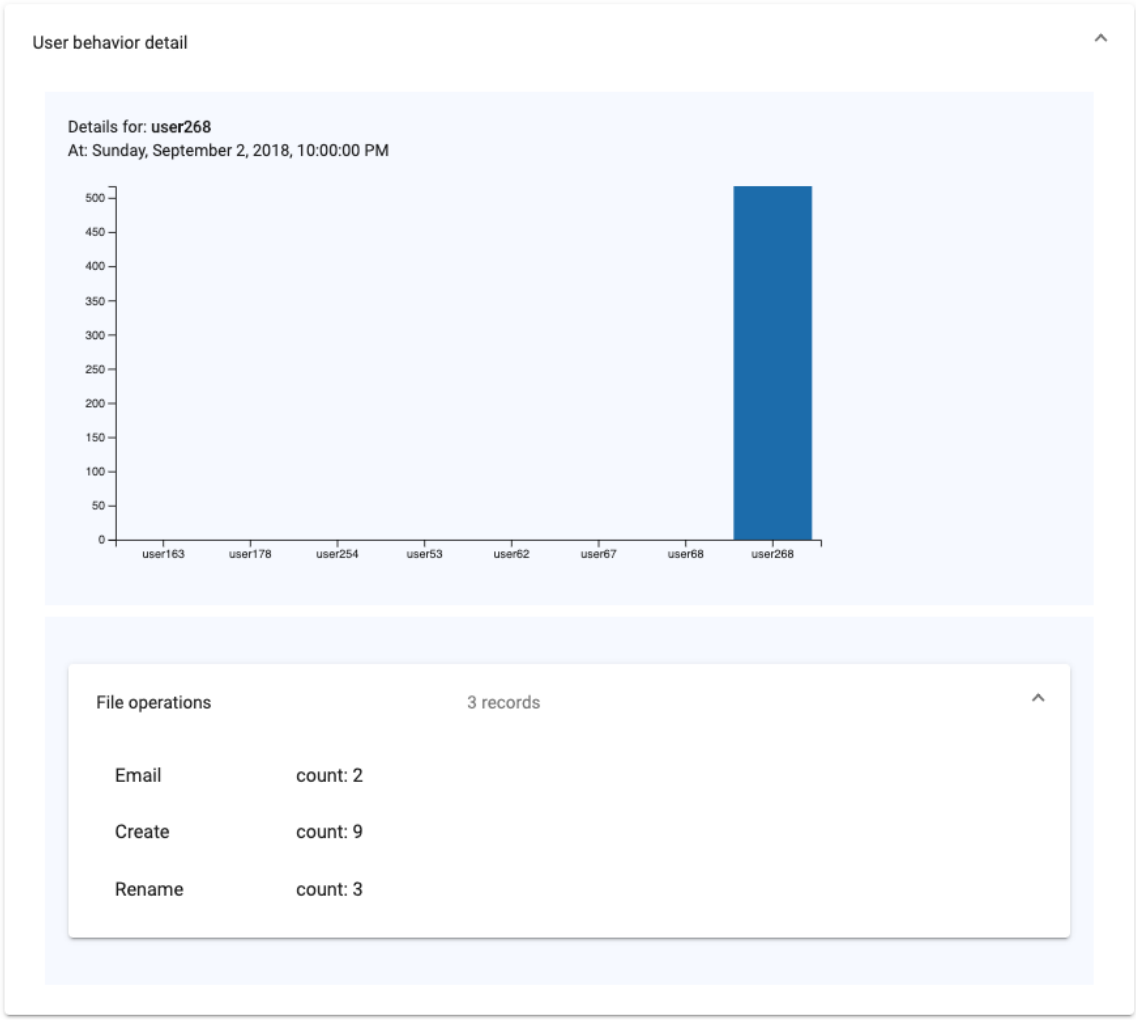
Obr. A.1: Ukážka vizualizačného prvku, pomocou ktorého je možne preskúmať normálne a anomálne správanie používateľov. Prvok taktiež obsahuje zoznam detekovaných anomálií v aktivite jednotlivých používateľov.



Obr. A.2: Ukážka vybraných vizualizačných prvkov, pomocou ktorých je možné špecifikovať analyzovaného používateľa a časový interval danej analýzy. Po špecifikovaní týchto informácií je zobrazená aktivita používateľa, spolu s detekovanou anomálnou aktivitou.



Obr. A.3: Ukážka vizualizačného prvku, pomocou ktorého je možne porovnávať aktivitu medzi jednotlivými vybranými používateľmi. Vybrať používateľa pre porovnanie je možné v druhom okne aplikácie. Všetci ponúknutý používatelia pre výber sú z rovnakého skupinového profilu.



Obr. A.4: Ukážka detailu aktivity vybraného používateľa. Detail aktivity je špecifický pre vybranú hodinu a skladá sa z grafu, obsahujúceho aktivitu všetkých používateľov z rovnakého skupinového profilu a zo zoznamu súborových operácií, ktoré boli v danú hodinu vykonané.