VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ
DEPARTMENT OF BIOMEDICAL ENGINEERING

ALGORITMY PRO DETEKCI ANOMÁLIÍ V DATECH Z KLINICKÝCH STUDIÍ A ZDRAVOTNICKÝCH REGISTRŮ
ALGORITHMS FOR ANOMALY DETECTION IN DATA FROM CLINICAL TRIALS AND HEALTH REGISTRIES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Maxim Bondarenko

VEDOUCÍ PRÁCE
SUPERVISOR
doc. Ing. Daniel Schwarz, Ph.D.

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Biomedicínské inženýrství a bioinformatika**
Ústav biomedicínského inženýrství

Student: Bc. Maxim Bondarenko ***ID:*** 192479
Ročník: 2 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Algoritmy pro detekci anomálií v datech z klinických studií a zdravotnických registrů

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši problematiky kvality dat ve zdravotnickém výzkumu s vazbou na existující informační systémy pro elektronický sběr dat (EDC systémy – Electronic Data Capture). 2) U vybraného informačního systému (CLADE-IS: Clinical Data Warehousing Information System) navrhněte rozšíření palety funkcí o automatické monitorování kvality dat detekcí anomálních záznamů: a) pomocí statistických metod, b) pomocí metod strojového učení (machine learning) a rozpoznávání vzorů (pattern recognition), a to včetně návrhu hodnocení úspěšnosti detekce. 3) Zabývejte se předzpracováním dat, a to konkrétně transformacemi datových záznamů s proměnnými různých datových typů na numerické vektory. 4) Pro navržené algoritmy realizujte softwarové řešení, přičemž využijte SQL databázi (PostgreSQL) a jeden ze skriptovacích jazyků (např. Python, PHP) nebo jiné vývojové prostředí (např. Matlab, R). Využijte data z již uzavřených zdravotnických registrů nebo neintervenčních klinických studií.

DOPORUČENÁ LITERATURA:

[1] Knepper D et al.: Statistical monitoring in clinical trials: best practices for detecting data anomalies suggestive of fabrication or misconduct, DOI: 10.1177/2168479016630576.

[2] Stephen L George & Marc Buyse: Data fraud in clinical trials, DOI: 10.4155/CLI.14.116.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 18.5.2018

Vedoucí práce: doc. Ing. Daniel Schwarz, Ph.D.
Konzultant:

prof. Ing. Ivo Provazník, Ph.D.
předseda oborové rady

Zveřejnění této práce je odloženo v souladu s ustanovením §47b odst. 4. zákona č. 111/1998 Sb. (Zákon o vysokých školách) v platném znění.

Důvodem odložení zveřejnění práce je ochrana právem chráněných zájmů společnosti Institut biostatistiky a analýz, s.r.o., 02784114, která za účelem zpracování této závěrečné práce poskytla autorovi své know-how a/nebo důvěrné informace vztahující se ke zpracovávané problematice.

ABSTRAKT

Daná diplomová práce se zabývá problematikou detekci anomálií v datech z klinických studií a zdravotnických registrů. Cílem práce je provedení literární rešerše problematiky kvality dat ve zdravotnickém výzkumu a realizace vlastního algoritmu detekce anomálních záznamů založeného na metodách strojového učení v reálných klinických datech z běžících nebo uzavřených klinických studií či registrů. V praktické části je popsán realizovaný algoritmus detekce, který se skládá z několika částí: import datového souboru z informačního systému, předzpracování a transformace importovaných datových záznamů s proměnnými různých datových typů na numerické vektory, využití známých statistických metod pro detekce outlierů a hodnoceni kvality a přesnosti algoritmu. Výsledkem zpracování algoritmu je vektor parametrů obsahujících anomálií, který má usnadnit práci správci dat. Tento algoritmus je navřen pro rozšíření palety funkcí informačního systému (CLADE-IS) o automatické monitorování kvality dat detekcí anomálních záznamů.

KLÍČOVÁ SLOVA

EDC systémy, klinické studie, outliery, mahalanobisová vzdálenost, euklidovská vzdálenost, kosinová podobnost, kvalita dat, strojové učení.

ABSTRACT

This master's thesis deals with the problems of anomalies detection in data from clinical trials and medical registries. The purpose of this work is to perform literary research about quality of data in clinical trials and to design a personal algorithm for detection of anomalous records based on machine learning methods in real clinical data from current or completed clinical trials or medical registries. In the practical part is described the implemented algorithm of detection, consists of several parts: import of data from information system, preprocessing and transformation of imported data records with variables of different data types into numerical vectors, using well known statistical methods for detection outliers and evaluation of the quality and accuracy of the algorithm. The result of creating the algorithm is vector of parameters containing anomalies, which has to make the work of data manager easier. This algorithm is designed for extension the palette of information system functions (CLADE-IS) on automatic monitoring the quality of data by detecting anomalous records.

KEYWORDS

EDC systems, clinical trials, outliers, mahalanobis distance, euclidean distance, cosine similarity, data quality, machine learning.

BONDARENKO, M. *Algoritmy pro detekci anomálií v datech z klinických studií a zdravotnických registrů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav Biomedicínské inženýrství a bioinformatika, 2018. 55 s., 13 s. příloh. Diplomová práce. Vedoucí práce: doc. Ing. Daniel Schwarz, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou závěrečnou práci na téma Algoritmy pro detekci anomálií v datech z klinických studií a zdravotnických registrů jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce doc. Ing. Daniel Schwarz, Ph.D. za vedení diplomové práce a také za jeho podporu, trpělivost, cenné rady, inspiraci a diskuze při vypracování této diplomové práce.

V Brně dne ....

(podpis autora)

OBSAH

ÚVOD

Poctivost a pravdivost jsou základními principy vědeckého výzkumu. Dodržování těchto zásad je nezbytné jak pro rozvoj vědy, tak pro veřejné vnímání vědeckých výsledků. Odchylky od těchto zásad mohou být považovány za vědecké pochybení nebo podvod. V oblasti klinického výzkumu může nedodržování těchto zásad vést k ohrožení lidského života.

Chybná data se stále častěji objevuj v klinických studiích v současností. Ve své diplomové práci se zaměřuji na metody detekce odlehlých hodnot nebo jinak anomálních záznamů v databázích s anonymizovanými daty z národních a nadnárodních neintervenčních klinických studií či zdravotnických registrů.

Cílem této diplomové práce je:

- připravit přehlednou rešerši o metodách detekce odlehlých hodnot,
- seznámit se z vybraným informačním systémem (CLADE-IS: Clinical Data Warehousing Information System),
- realizovat import dat z vybraného informačního systému pomocí SQL (PostgreSQL) příkazů,
- transformovat původní data ze systému pro elektronický sběr dat na data numerická a tj. pro statistickou analýzu,
- vyzkoušet známé algoritmy založené na jednorozměrných statistických metodách, a to na konkrétních souborech dat z běžících nebo uzavřených klinických studií či registrů,
- realizovat softwarové řešení či algoritmus pro rozšíření palety funkcí o automatické monitorování kvality dat detekcí anomálních záznamů u vybraného informačního systému,
- provést hodnocení úspěšnosti realizovaného algoritmu detekce.

V první části práce jsou stručně popsány fáze klinické studie a typy klinických studií, a takže přivedena informace o datech z běžících nebo uzavřených klinických studií či registrů a informačního systému (CLADE-IS), pomocí kterého tyto údaje byly nasbírány.

V další častí se popisuje rešeršní práci autora. Zde je rozepsána problematika kvality dat a zdrojů nekvalitních dat a taky uvedeny metody detekce anomálie v datech. Takže zde je popsána metoda hodnocení úspešnosti klasifikace.

Praktická část této diplomové práce zahrnuje popis realizovaného algoritmu a jejího hodnocení úspešnosti.

1 KLINICKÉ STUDIE

Klinické studium (klinický projekt nebo výzkum) může být definován jako zkoušky nebo projekty, které se provádějí buď pomocí zdravotnických pomůcek anebo klinického hodnocení léčivých přípravků.

Klinická studie (KS) se nejčastěji provádí v biomedicínském nebo zdravotním výzkumu, a v jiných příbuzných oborech, jako je například psychologie. V tomto případě se provádí výzkum vyhodnocující intervence. Obvykle se to provádí porovnáním dvou nebo více přístupů. Výzkumní pracovníci provádějící klinické zkoušky hledají různé výzkumné cíle v rámci různých výzkumných formátů. KS mohou zahrnovat otázky, které přímo nesouvisejí s terapií (například nákladová účinnost, metabolismus léků atd.), kromě těch, které přímo ovlivňují léčbu subjektů. Proces uvedení lék na trh je poměrně dlouhý a složitý. Současně jsou klinické studie v nejkritičtější, nákladnější a časově náročnější fázi procesu vývoje léků. Rozhodnutí o konečném schválení léčivého přípravku je ve většině případů založeno na datech KS [1]. Existuje celkem čtyři fáze KS [2], [3]:

0. fáze – preklinická fáze. Sponzor vyvíjí novou sloučeninu léčiva a provádí studie na zvířatech s cílem identifikovat potenciální nežádoucí účinky. Pak sponzor podává žádost o klinickou studii pro nové léčivo na základě výsledků počátečního testování a vypracuje plán zkoušek pro lidi.
1. fáze klinického studie. Tato fáze je navržena tak, aby poskytovala hlubší pochopení bezpečnosti léku včetně vedlejších účinků spojených s jeho dávkou. Je třeba tady poznamenat, že KS první fáze zahrnují stále častěji osoby se specifickými nemocemi – osoby, u kterých selhaly všechny konvenční terapie (např. karcinom kolorekta). Tyto studie mohou být označeny jako KS fáze I, které by měly být ve skutečnosti označeny jako smíšené KS fáze I / II nebo čisté fáze II. V této fázi se obvykle účastní 10 až 100 subjektů.
2. fáze. Studie fáze II hodnotí účinnost léku pro konkrétní terapeutické aplikaci u pacientů. Současně pokračuje hodnocení bezpečnosti léčivého přípravku. Hlavním cílem je získat předběžné údaje o tom, zda léčivo funguje pro lidi s určitou nemocí nebo v určitém stavu. Typický počet subjektů v teto fáze 100 až 1000.
3. fáze. Studie fáze III zahrnují poměrně velký počet pacientů (n> 1000) a jsou navrženy tak, aby shromažďovaly dostatečné informace o bezpečnosti i účinnosti léku a splňovaly požadavky úřadu kontroly potravin a léčiv pro adekvátní zhodnocení poměru prospěch/riziko, jakož i na přípravu informací pro označování léčiv. V teto fáze se především jedná o stanovení celkové účinnosti, bezpečnosti a porovnání efektivity nového léku se standardní léčbou. Fáze III a IV KS jsou navrženy tak, aby zvýšily přežití nebo kvalitu života subjektů trpících specifickou chorobou.

4. fáze. Pro tuto fázi je specifické schválení a registraci léků úřadem pro kontrolu potravin a léčiv. Tato fáze představuje poslední KS po registraci léků. Zahrnuje postmarketingové sledovací studie, které především zkoumají dlouhodobou účinnost a toxicitu již uváděných na trh léků.

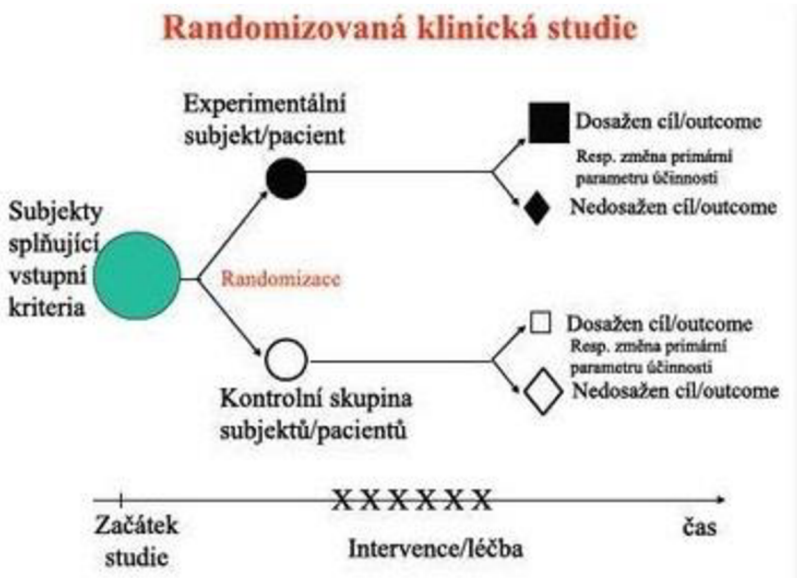
1.1 Typy klinických studií

Klinická studie je charakterizována léčbou s použitím farmakologických léků, nebo použitím jiného léčebného procesu (např. rehabilitace). KS se dělí na **studie s kontrolou** a **studie bez kontroly**:

- **Studie s kontrolou**

Studie s kontrolou vždycky má alespoň dvě ramena, tj. první je experimentální a druhé – kontrolní. Nejběžnějším a nejčastěji používaným typem tohoto výzkumu je randomizovaná klinická studie s minimálně jedním experimentálním ramenem a jedním kontrolním, což umožňuje paralelní průběh léčby ve dvou ramenech.

Randomizovaná klinická studie je studie, ve které pacienti jsou náhodně zařazování do skupin léčby a mají stejnou příležitost získat zkušební nebo kontrolní léčivý přípravek (referenční léčivo nebo placebo). Placebo je specifická látka, která má stejnou barvu a tvar jako testovaný lék. Ale když se pacient nemůže obejít bez léčby a placebo nemůže v tomto případě být použitelné, tak se srovnávají účinky nového léku se standardním se používajícím lékem. Randomizovaná studie může být **otevřená** (všichni účastníci (pacient i lékař) vědí, jaké léky pacient obdrží), **jednoduše zaslepená** (pacient neví, do kterého ramene studie je randomizován), **dvojitě zaslepená** (neví to ani pacient ani lékař) nebo **trojitě zaslepená** (ani statistik neví, které ze studií hodnotí, protože má pouze údaj o randomizaci typu A nebo B) [5].



Obr. 1.1: Schéma designu randomizované klinické studie.[5]

Kvůli tomu, že léčba probíhá v obou dvou paralelních směrech, tak každý pacient bude léčen pouze jednou z těchto randomizovaných terapií. Také se často používá zkřížený design, kdy pacient náhodně dostane po sobě obě dvě terapie.

Zkřížená studie (cross-over design) je studie, kdy každý pacient dostává oboje srovnávané léky obvykle v náhodném pořadí. Křížový design má hodně výhod, protože dává možnost ocenit personální odezvu pro každého pacienta na oba typy léčby a následně porovnat účinek v prvním a druhém případě, což umožňuje srovnat, jak se měnily účinky léčby v čase [6].

- **Studie bez kontroly**

Jedná se o studii, ve kterých například buď chceme prokázat, že léčebná odezva se vyskytne u většího procenta případů, než je daná procentní hodnota (například 60%), anebo ukázat, že procento nežádoucích efektů je nižší než to předem stanovené procento. Tenhle efekt ale je však statisticky významný (větší či menší)!

Zde se jedná o studie bez terapeutického postupu, ale zároveň se používají diagnostické postupy včetně např. biopsie.

Existuje 3 základní designy těchto studií [6]:

1) **prospektivní studie,**
2) **retrospektivní studie,**
3) **průřezová studie.**

Prospektivní (kohortová) studie se provádí tak, že se účastníci rozdělí na dvě skupiny. První skupina sebou představuje pacienty se sledovaným rizikovým faktorem, druhá – pacienti bez rizikového faktoru. V obou skupinách se sleduje vliv rizikového faktoru na vznik nemoci v čase a zjišťuje se, kolik pacientů onemocnělo.

Naproti tomu **retrospektivní studie** zkoumá příčinu nemoci (tj. vycházíme z nemoci a jdeme směrem k příčině nemoci). V tomto typu studií sledujeme dvě skupiny pacientů (osoby s nemocí a bez nemoci) a určujeme rizikový faktor (příčinu nemoci).

Průřezová (cross-sectional) studie zkoumá objev forem a stádií onemocnění v populaci a taky sleduje podíl nemocných osob a počet osob s rizikovým faktorem. Tento typ výzkumu doplňuje dva předchozích (prospektivní a retrospektivní).

V závislosti na počtu výzkumných center, ve kterých je výzkum prováděn v souladu s jediným protokolem, jsou studie jednocentální a multicentrické. Pokud se studie provádí v několika zemích tak se nazývá mezinárodní [5].

V současné době dávají přednost tomu typu klinického výzkumu, který poskytuje nejspolehlivější údaje. V poslední době úloha klinických studií léků v souvislosti se zavedením principů klinického výzkumu se stará o zlepšování kvality zdravotní péče a propaguje rozhodování založené na důkazech. Hlavním z nich je přijetí specifických klinických řešení pro léčbu pacienta na základě přesně prokázaných vědeckých údajů, které lze získat pomocí dobře plánovaných kontrolovaných KS.

1.2 Systém elektronického sběru dat

Systém elektronického sběru dat (electronic data capture – EDC) je počítačový systém určený pro sběr klinických dat do elektronického formátu vhodného pro použití v KS. EDC nahrazuje tradiční metodiku sběru dat v papírové podobě, která zjednodušuje shromažďování údajů a zrychluje čas pro vstup na trh léků a zdravotnických prostředků. Řešení EDC jsou široce používána farmaceutickými společnostmi a klinickými výzkumnými organizacemi [7].

V dnešní době se data vkládají do informačního systému přes speciální elektronický záznam subjektů studie (Electronic Case Report Form eCRF). Tento formulář je nástrojem, který využívá sponzor klinického hodnocení ke shromažďování údajů od každého pacienta. Všechna data o každém pacientovi, který se zúčastnil klinického hodnocení, jsou uchovávána nebo dokumentována v CRF, včetně nežádoucích účinků. Z pohledu garanta KS je hlavním logistickým cílem klinického pokusu získat přesný CRF. Avšak, kvůli lidské a strojové chybě jsou údaje zadané v CRF málokdy zcela přesné nebo zcela čitelné. Pro odstranění těchto chyb garant najímá experty pro zajišťování kvality eCRF, aby ujistil, že CRF obsahuje správná data [7].

V této semestrální práci se používají data z CLADE-IS (Clinical Data Warehousing Information System) systému. CLADE-IS je informační systém pro skladování klinických dat patřící do skupiny nejmodernějších a progresivních systémů EDC, které byly vyvinuty na bázi web technologií. Je spojen s modelem databáze Entity -Attribute - Value. EAV model je datovým modelem, který je určen pro uložení datových struktur představujících sebou (obvykle velké) množství atributů, které jsou popsány určitými hodnotami. Model EAV je taky známý jako horizontální model databáze nebo otevřeného schématu a umožňuje kódování datových struktur nebo objektů s rozptýlenými charakteristikami. To je přesně ten případ datových objektů v databázích klinických výzkumů, kde počet parametrů nebo atributů použitých k popisu objektu je potenciálně velký, ale skutečný počet použitých atributů s přiřazenými hodnotami je relativně nízký. Konstrukce této platformy zahrnuje originální a komplexní online designer i generátor formulářů. Tyto komponenty umožňují definovat všechny potřebné subjekty sběru klinických dat, jako jsou rameno studia, fáze, formuláře, skupiny dotazů a jednotlivé otázky se všemi možnými typy dát. Efektivní schopnost nasazení CLADE_IS se dosahuje díky vlastnímu datovému modelu, který umožňuje nastavení množství uživatelských oprávnění, rolí a datového toku. Klasická konfigurace zahrnuje následující učitelské role: vyšetřovatel, správce webu, regionální koordinátor, správce dát, monitor, administrátor systémy.

Aktuální verze CLADE_IS záleží na open source objektově relačním databázovém systému. Má více než 15 let aktivního vývoje a osvědčenou architekturu, která získala dobrou reputaci za svou spolehlivost, integritu dát a kvalitu. CLADE-IS je ve skutečnosti spolehlivou platformou pro návrh a nasazení samostatné EDC systémů pro sběr a správu dat v klinických studiích různých typů a širokém spektru klinických oborů [8].

1.3 Data z reálné klinické praxe

Takzvaná Real World Evidence (RWE) data se v poslední době stala hitem na poli výzkumu ve farmaceutickém průmyslu a v segmentu life sciences. Nové technologie jako například elektronické zdravotní záznamy a nástroje pro data mining jsou nyní k dispozici a umožnují získat dříve neznámé informace a znalosti z oblasti zdravotnictví. Například, RWE může pomoct zjistit náklady na léčbu, její efektivitu (náklady, přínosy a rizika), ceny na léky, vedlejší účinky nebo dlouhodobé výsledky léčby.

Údaje získané mimo randomizovaná klinická studie, obsahující faktické informace o pacientech, se nazývají Real World Data (RWD). V oblasti klinického výzkumu se často používají pojmy „Real World Data“ a „Real World Evidence“ téměř bez rozdílu, ale nejsou to přesné synonyma. „Data“ znamená faktické informace a jsou surovinou, zatímco "Evidence" znamená, že organizace bude používat informaci, která má přivést k závěru. RWE jsou získávána ze zdrojů RWD, jako elektronické lékařské záznamy, laboratorní informační systémy, radiologické systémy a lékařské registry. Zdrojem dat RWD jsou také údaje shromážděné od pacientů z domácích a přenosných zařízeních pro monitorování, ze sociálních sítí a z mobilních aplikací. Všechny tyto údaje obsahují velké množství neprozkoumané informaci, která může mít obrovský dopad na klinický výzkum.

RWD má spoustu perspektivních oblastí použití, a to i ve stadiu klinických zkoušek léku. Také RWD může urychlit proces vytváření hypotéz s cílem informování o vývoji KS a umožnění identifikaci podskupin s vyšším poměrem «riziko/přínos» jako cílových skupin. RWD mohou také přispět k efektivnějšímu zařazení pacientů pro účast v KS. A nakonec, dřívější generace údajů o účinnosti léku může pomoct urychlit vstup na trh a také pomoct rozhodnout o jeho nákladech, což je velmi důležité v rámci státních programů, které poskytují obyvatelům řádnou lékařskou péči [9].

Kromě toho RWE poskytuje významné příležitosti ke zlepšení post marketingových aktivit, což zkracuje čas a zmenšuje náklady na provedení studií ve čtvrté fáze prostřednictvím účinnějších a včasně provedených metod sběru dat.

2 KVALITA DAT

Problém kvality dat je v současné době jedním z nejdůležitějších problémů, které je třeba řešit při provádění a vyhodnocování klinických studií. Klíčem pro úspěšné KS je získaní kvalitních dat, která podporují pravdivost závěrů po jejich statistickém vyhodnocení. Kvalita údajů v klinických studiích může být ovlivněna zároveň několika faktory:

- Chyby při zadávání do chorobopisů a nemocničních informačních systémů,
- Chyby při přepisování do EDC systémů,
- Falsifikace, arteficiální generování dat a podvodná data (data fraud).

Tato kapitola popisuje nejčastější zdroje nekvalitních dat a metody jejích detekce.

2.1 Zdroje nekvalitních dat

Podvody v datech mají největší negativní dopad na výsledek klinického výzkumu, neboť nejen porušují zákon, ale také ničí reputace všech, kteří se na výzkumu podíleli. Detekce podvodu je jedním z aspektů zajištění kvality údajů v klinických studiích. Součásti dobré klinické praxe je to, že zkušení garanti jsou povinní sledovat provádění klinických studií. Cílem sledování klinických studií je zajistit pohodlí pacientům, dodržování schválených protokolů, regulačních požadavků, přesnost a úplnost údajů. Sledování klinických studií se rozděluje do třech skupin: zkušební komise, monitorování na místě a centrální statistické sledování. Ale z toho vyplývá, že využívání těchto zdrojů je užitečné z hlediska jejich vlastního práva garantovat kvalitu zkušebních údajů a platnosti výsledků zkoušek. Kontrola, kterou provádí zkušební komise, je obzvlášť užitečná pro prevenci nebo odhalování chyb ve fázi návrhu a interpretaci výsledků. Kontrola na místě je taky obzvláště užitečná k prevenci nebo odhalení procedurálních chyb v průběhu testování v zúčastněných centrech (např. zda všichni pacienti nebo právně přijatelní zástupci podepsali informované souhlasy). Statistické sledování je obzvlášť užitečné při zjišťování datových chyb, ať už kvůli chybnému vybavení, nedbalosti, neschopnosti nebo podvodu [10].

Z průzkumu současných postupů sledování vyplývá, že převážná většina studií je sledována především prostřednictvím návštěv na místě s ověřením zdrojových dat. Toto ověření spočívá ve srovnání informací zaznamenaných ve formuláři s odpovídajícími zdrojovými dokumenty [11]. I když existuje obecná shoda, že je nutné provést některé monitorování na místě, stále více se zpochybňuje úloha ověřování zdrojových dat, zejména rozsáhlá ověření zdrojových dat [12]. Ověření zdrojových dat zjišťuje nesrovnalosti způsobené chybami přepisu ze zdrojové dokumentace do formuláře hlášení kazu, nikoli však chyby obsažené ve zdrojových dokumentech. Ověření zdrojových dat může být užitečné pro zajištění přesného zachycení primárního výsledku pokusu a některých klíčových bezpečnostních parametrů, avšak úplné (100%) ověření všech zdrojových dat je obzvlášť nákladově neúčinné [10],[13]. Nedávné pokyny od FDA a EMA jednoznačně upřednostňují využívání přístupů "kvalita od návrhu" a monitorování založených na rizicích místo tradičního monitorování technologie KS,

které se ukázaly jako nákladné i neúčinné [12]. Konkrétně lze vyčerpávající ověření zdrojových dat nahradit cílovými datovými audity, pokud jsou uvedeny.

Zdá se poněkud paradoxní, že statistická teorie, která je tak zásadní pro navrhování a analýzu klinických hodnocení, nebyla dosud použita k tomu, aby pomohla optimalizovat monitorovací aktivity. Nicméně potenciál statistiky je v odhalování podvodů v multicentrických studiích je pozorován více než deset let [12].

Problémy detekce anomálií nemají jednu definice a jsou často interpretovány různě v závislosti na typu dát a stanovených cílů [1,3,5]. Intuitivně, anomálie znamená něco, co nespadá do obecných pravidel a zákonů, které jsou platné pro předložené údaje. Taková definice potřebuje formální zdokonalení před vyřešením problému matematickými metodami.

V této práce bylo předpokládáno, že data mají charakteristikou reprezentaci, a tj. každý objekt x je daný určitým vektorem R^d. V klasické formulaci problému detekce anomálií je formulována takhle: v dané množině X pro každý prvek dáváme 0, jestliže tento objekt patří do skupiny normálních dat a 1 jestli tento objekt je abnormální. Takový úkol patří do třídy učení bez učitele, protože správné odpovědi ze strany vstupních dat nejsou k dispozici.

V podobném úkolu učení s učitelem je známá správná odpověď na některé části *Xtrain* vstupních dát, to znamená, že pro každý objekt $x \in Xtrain$ jsou známy štítky

$y(x) \in \{0,1\}$. Odsud plyne, že objekt je anomálií.

Úloha přidání hodnot {0,1} pro nová data *Xtest* je formálně úkolem binární klasifikace. Proto může být řešena pomocí jakýchkoli algoritmů strojového učení s učitelem. Nicméně, existuje jiná varianta, když všechny hodnoty $y(x)$, $x \in Xtrain$ jsou 0. Znamená to, že jsou uvedeny pouze normální data. V tomto případě algoritmy binární klasifikace budou dávat irrelevantní konstantní předpověď.

Téměř skoro všechny algoritmy detekce anomálií se redukují do určité funkce, která pro nějaký daný objekt vytváří určitý rating anomálií. Poté výstupní data se dělí do tříd anomálie a normální údaje. Toto rozdělení se provádí pomocí binarizace s určitou prahovou hodnotou, přičemž výběr prahu je zvláštní stupni řešení tohoto problému. Při absence předem známé informace je k dispozici pouze informace o jednorozměrném rozdělení vstupních hodnot, což je nedostatečné pro rozumné rozhodnutí. Ve většině případů je známy přibližný podíl anomálií v datech. V takových případech je jako prahová hodnota zvolen vhodný kvantil.

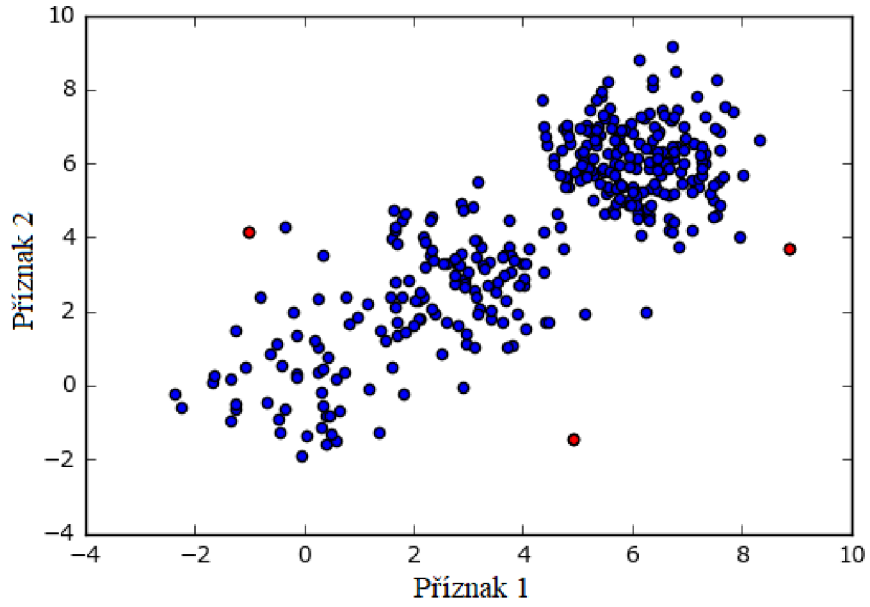
3 METODY DETEKCE ANOMÁLIE V DATECH

Statistické sledování KS používá několik základních postupů založených na povaze shromážděných údajů. Předběžná úprava pro jakoukoli studii zahrnuje:

- čištění dat, které spočívá v odstranění šumu a špatných údajů;
- komprese dat včetně stanovení minimální příznakového prostoru a reprezentativního souboru dat založeného na metodách redukce a transformace;
- kombinování dat umožňující snížit množství dat, přičemž se udržuje původní informace pomocí heuristických algoritmů.

V KS se obvykle provádí pouze čištění s důrazem na odstranění outlierů.

Je důležité si uvědomit, že extrémní význam a anomální hodnota jsou odlišné pojmy. Například v malém vzorku: *[1, 39, 2, 1, 101, 2, 1, 100, 1, 3, 101, 1, 3, 100, 101, 100, 100]*, hodnota *39* může být považována za anomálii, i když to není maximum nebo minimum. Tady je určitě nutné poznamenat, že anomálie jsou zpravidla charakterizovány nejen extrémními hodnotami jednotlivých příznaků (Obr. 3.1).



Obr. 3.1: Příklad výskytu outlierů při porovnání dvou příznaků mezi sebou.

Na obrázku výše červenou barvou jsou označeny outliery se vyskytující při porovnání dvou příznaků.

Odlehlé hodnoty (outliery) jsou hodnoty, které se výrazně liší od ostatních hodnot v shromážděném datovém souboru. Mohou být způsobeny, chybami při měření, nesprávným záznamem dat, chybou měřicího přístroje či laboranta, technika atd. Odstraněním outlierů z datového souboru se může dosáhnout přesnějších výsledků.

Pro zajištění správného pochopení statistických údajů je třeba vypočítat a odhadnout outliery.

Hodnoty x_i se označují jako outliery, pokud jsou splněné následující podmínky:

$$X_i < Q1 - 1.5 \cdot IQR, \tag{3.1}$$

nebo

$$X_i > Q3 + 1.5 \cdot IQR, \tag{3.2}$$

kde $Q1$ je první kvartil, $Q3$ je třetí kvartil, 1.5 je outlierový koeficient a IQR je interkvartilové rozpětí (tedy $Q3$ - $Q1$).

Detekci outlierů v datech je možné provádět následujícími způsoby:

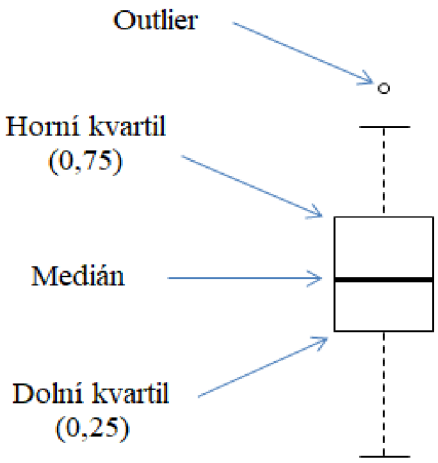
- Vizualizaci dat,
- Jednorozměrnou statistickou metodou
- Vícerozměrnou statistickou metodou.

Kromě toho pro zlepšení kvality detekce anomálie se často používá princip Ensemble learning [14]. Základní myšlenkou této metody je, že jeden algoritmus je dobře, ale několik algoritmů ve skupině je mnohem lepší. Což znamená, že výsledek daný několika metody je přesnější než výsledek daný jednou metodou.

3.1 Vizualizace dat

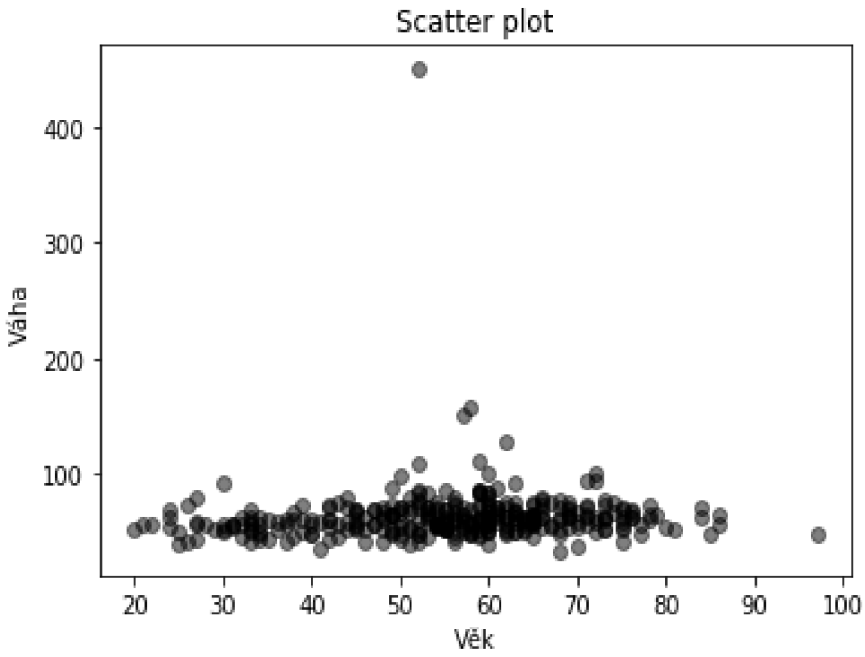
Před zpracováním datového souboru by měly být stanoveny potenciální outliery. To lze snadno zjistit pomocí vizualizace dat či statistickými metodami detekce outlierů. Pokud jsou hodnoty datové sady vykresleny na grafu, odlehlé hodnoty se nacházejí daleko od většiny ostatních hodnot. Pokud například většina hodnot spadá na přímku, outliery leží po obou stranách takové přímky. Pro vizualizaci dat lze použít krabicové grafy či bodové grafy:

Krabicový graf (boxplot) je graf používaný v popisné statistice, který kompaktně zobrazuje jednorozměrné rozdělení pravděpodobnosti (Obr. 3.2). Tento typ grafu ukazuje medián (nebo pokud je třeba, průměr), dolní a horní kvartil, minimální a maximální hodnoty vzorku a outliery. Několik takových krabic může být nakresleno vedle sebe pro vizuální porovnání jedné distribuce s druhou. Mohou být umístěny jak horizontálně, tak vertikálně. Vzdálenosti mezi různými částmi krabice umožňují určit stupeň rozptýlení (disperze) i asymetrie dat a taky identifikovat outliery. Outliery jsou mezi vnějšími a vnitřními hradbami, tj. v intervalu ($x_{0,75}$ + 1,5q, ∞) nebo v intervalu (-∞, $x_{0,25}$ − 1,5q), kde q je interkvartilové rozpětí.



Obr. 3.2: Krabicový graf (boxplot).

Bodový graf (scatterplot) je matematický diagram znázorňující hodnoty dvou proměnných ve formě bodů v Kartézském systému (Obr. 3.3). Na bodovém grafu každá základní jednotka datové sady odpovídá bodu, jehož poloha se rovná hodnotám dvou parametrů. Pokud se předpokládá, že jeden z parametrů je závislý na druhém, tak hodnoty nezávislého parametru se zobrazují na vodorovné ose a hodnoty závislého parametru – na svislé ose. Bodový graf se používá k prokázání přítomnosti či nepřítomnosti korelace mezi dvěma proměnnými.

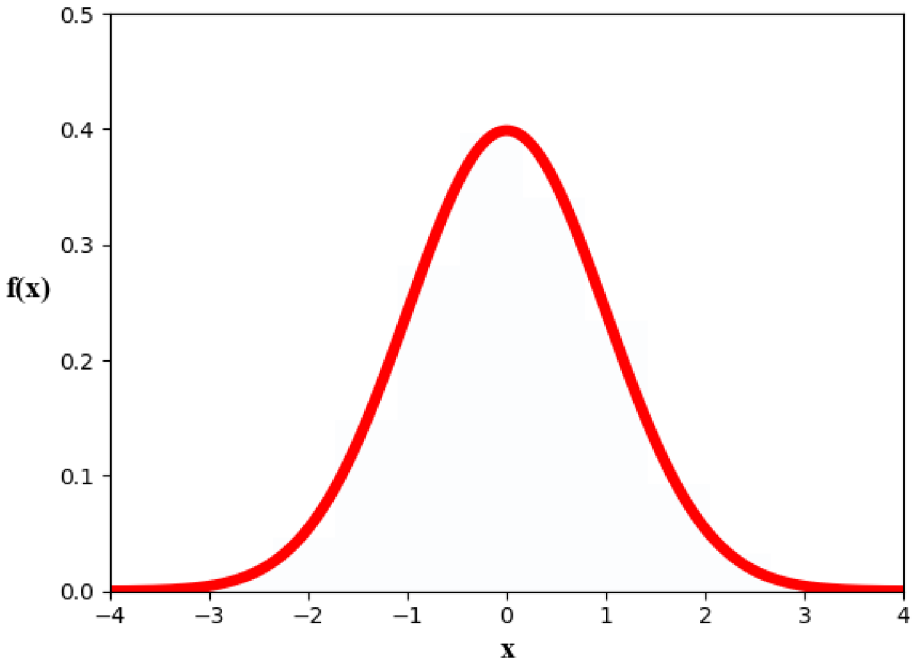


Obr. 3.3: Vlastní příklad bodového grafu.

Na obrázku 3.3 je pěkně vidět přiklad výskytu odlehlé hodnoty. Zde je vidět, že jedna z hodnot váhy je anomálně velká.

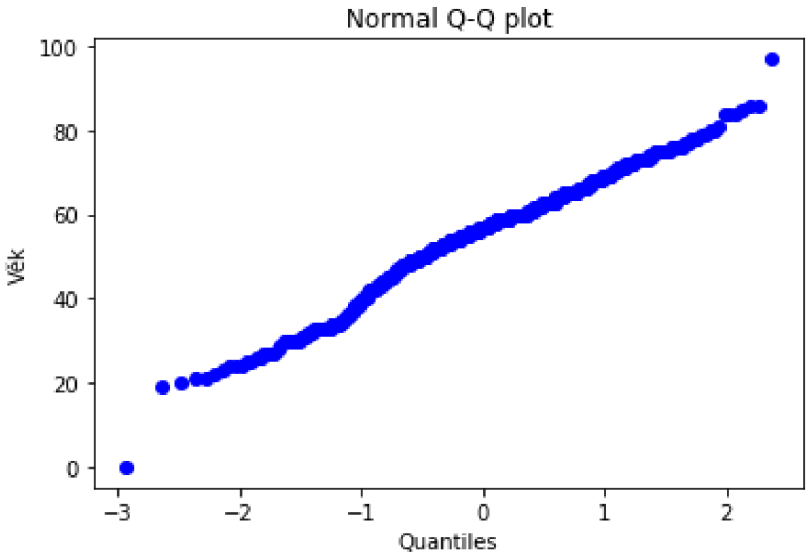
Volba statistických metod analýzy dat závisí na tom, jestli má vybraný vzorek normální rozložení nebo ne. Proto táto diplomová práce klade důraz na ověření normality dat. Pokud rozdělení dat se považuje za normální, lze je kontrolovat několika způsoby (grafické ověřování normality a/nebo pomocí testů pro ověřování normality).

Grafické ověření normality je jednoduchý způsob, jak přibližně odhadnout, zda data mají normální rozdělení. Nejběžnějším způsobem grafických metod je vytvoření histogramu distribuce dat a porovnání s křivkou popisující hustotu pravděpodobnosti normálního rozdělení (Obr. 3.4).



Obr. 3.4: Křivka popisující hustotu pravděpodobnosti normálního rozdělení.

Na ose X leží proměnné ve stanoveném intervalu a na ose Y je zobrazena distribuce těchto proměnných. Při normálním rozdělení histogram odebraného datového vzorku by měl připomínat znázorněnou křivku. Pro přesnější stanovení normality existují grafy Q-Q (*kvantil-kvantil*) a P-P (*pravděpodobnost-pravděpodobnost*). Q-Q graf (Obr. 3.5) je vhodnější pro testování normality na krajích rozdělení, zatímco P-P graf zdůrazňuje odchylky od normálního rozdělení poblíž střední hodnoty [15]. Kvantilem v matematické statistice se rozumí numerická charakteristika distribučního rozložení (rozdělení pravděpodobnosti) náhodných proměnných. Je takovým číslem, že daná náhodná proměnná může překročit ho pouze s pevnou pravděpodobností.



Obr. 3.5: Q-Q diagram.

Princip této metody spočívá v tom, že na jednu osu nanášíme kvantily hypotetického normálního rozdělení a na druhou osu – kvantily zkoumaného souboru. V případě normálního rozdělení Q-Q diagram má tvar přímky. Outliery jsou zobrazeny jako body, které leží daleko od celkového shromáždění bodů.

Postup při sestavení P-P grafu je podobný jako pro Q-Q graf. Jedna osa – kumulativní distribuce hypotetického normálního rozdělení a druhá představuje sebou hodnotu kumulativní distribuce zkoumaného souboru. Při normálním rozdělení body zase by měly ležet na přímce.

Všechny výše uvedené grafické metody mohou být ještě doplněny krabicovými diagramy popsanými na začátku.

Grafické metody ne vždy jsou vhodné i efektivní při analýze velkého množství datových vzorků, protože jsou moc náročné na zobrazování grafu pro každý vzorek. Proto se k vyřešení tohoto problému používají výpočtové testy pro ověření normality.

3.2 Jednorozměrné statistické metody detekce outlierů

Existuje řada testů, které se liší kvalitou a náročností provedení. V teto práce

se uvádějí pouze Shapirův-Wilkův, Andersonův-Darlingův a Kolmogorovův-Smirnovův testy.

Shapirův-Wilkův test se používá k testování hypotézy o normálním rozdělení.

V některých experimentech, zejména v lékařském výzkumu, je velikost vzorku malá. Zvláště pro testování normality rozložení malých (tří až padesát prvků) vzorků se používá tento test.

$$W = \frac{[\sum_{i=1}^{n} a_i \cdot x_i]^2}{\sum_{i=1}^{n} (x_i - \bar{x})}, \tag{3.3}$$

kde a_i jsou tabelované koeficienty, x_i jsou původní hodnoty, $\bar{x}$ je průměr původních hodnot. Je-li W menší než kvantil $W(\alpha)$, tak se hypotéza o normalitě výběru zamítá.

Kolmogorovův – Smirnovův test se používá k testování hypotézy o normálním rozdělení u velkých datových vzorků n > 50, a testováci statistika u tohoto testu má tvar:

$$D_n = \sup_{-\infty<x<\infty} \left| F_n(x) - \Phi(x) \right|, \tag{3.4}$$

kde $F_n(x)$ je výběrová distribuční funkce, $\Phi(x)$ distribuční funkcí normálního rozložení, *sup* je supremum množiny vzdáleností. Zde testujeme hypotézu o normálním rozdělení. Na hladině významnosti a zamítáme nulovou hypotézu, jestliže $D_n \geq D_n(a)$, kde $D_n(a)$ je tabelovaná kritická hodnota. Pro velký počet výběrů (n ≥ 50) lze aproximovat, že

$$D_n(a) \approx \sqrt{\frac{1}{2n} \cdot \ln \frac{2}{\alpha}}. \tag{3.5}$$

Podrobnější popis tohoto testu je popsán v publikace [16].

Andersonův-Darlingův test je založen na empirické distribuční funkci, kterou označíme jako $F_E(x)$. Tento test dává nám možnost ověřit nulovou hypotézu $H0 : F_E(x) = F_T(x)$ oproti alternativě $H1 : F_E(x) = F_T(x)$, kde $F_T(x)$ je distribuční funkce teoretického rozdělení, které je specifikováno, včetně jeho parametrů. Pro testový kritérium platí vztah

$$AD = -\frac{\sum_{i=1}^{m} (2i-1) \cdot (\ln z_i + \ln(1 - z_{n-i+1}))}{n} - n, \tag{3.6}$$

kde z_i jsou hodnoty distribuční funkce standardizovaného normálního rozdělení.

Pro parametr z_i pak platí:

$$z_i = F_T(x_{(j)}), \tag{3.7}$$

V případě, že poměr překročí (0,631, 0,752, 0,873, 1,035 nebo 1,159) při hladině významnosti (10%, 5%, 2,5%, 1% nebo 0,5%), zamítáme nulovou hypotézu $H0$.

Podrobnější popis těchto testů je uveden v [16].

Po zjištění, zda má vzorek normální rozložení nebo ne, je možné zvolit vhodnou statistickou metodu pro detekci outlierů v daném vzorku. Jednou z nejběžnějších metod

je test na násobek směrodatné odchylky. Nejjednodušším popisem tohoto testu je, že odstraňujeme vše, co překračuje 2 standardní odchylky. Na základě tohoto testu můžeme zcela jasně a přesně určit, zda daná hodnota je či není outlierem. Problémem však je, že tyto statistické testy a kritéria již v sobě obsahují předpoklad, že data mají normální rozdělení.

$$S_x = \sqrt{\frac{1}{n-1} \cdot \sum_{i-1}^{n} (x_i - \bar{x})^2}\,, \tag{3.8}$$

kde $\bar{x}$ je střední hodnota a S_x je směrodatná odchylka.

Implementace tohoto testu v jazyce Python je uvedena v příloze.

Vzhledem k tomu, že většina vzorků dat nemá normální rozdělení, bylo rozhodnuto použít MAD test pro stanovení outlierů.

Ve statistice MAD je spolehlivou měrou variability, která je jenom málo ovlivněna extrémními hodnotami. Kromě toho MAD je robustní statistika, která je odolnější vůči datové sadě, než je standardní odchylka, a nemá normální distribuci. Iglewicz a Hoaglin [17] doporučují použití MAD testu, když data nemají normální rozdělení, a popisují tento test takto:

$$M_i = \frac{0{,}6745 \cdot (x_i - \bar{x})}{MAD}\,, \tag{3.9}$$

kde MAD je medián absolutních odchylek a *0,6745* je tabulková hodnota kvantilu standardního normálního rozdělení *N (0,75)*.

$$MAD = median\left(\left|x_i - median(x)\right|\right), \tag{3.10}$$

Podle [17] je doporučeno, aby M_i s absolutní hodnotou větší než 3,5 byly označeny jako potenciální outliery.

Hlavní nevýhodou této metody je slabá různost dat. Je-li více než 50% dat mají stejnou hodnotu, parametr MAD bude nulový, a tím výsledek zkresluje. Proto před použitím této metody budeme se muset uchýlit k transformaci dat.

V současné době neexistuje přesně vhodná metoda pro detekce outlierů. V každém případě expert zkoumání dat vybírá unikátní soubor statistických metod k vyřešení problémů.

3.3 Vícerozměrné statistické metody

Většina dat pořízených při vědeckém výzkumu v klinických studiích je vícerozměrná, proto nám zpravidla nestačí u daných subjektů či objektů zjistit pouze jedinou vlastnost, ale celou řadu parametrů či proměnných, jako například hmotnost, barva, věk apod. Zřídkakdy nám stačí analyzovat každou proměnnou zvlášť, protože pro úplné pochopení vztahů mezi jednotlivými subjekty či objekty musíme analyzovat

většinu nebo dokonce všechny proměnné současně. V tom nám mohou pomoct vícerozměrné metody.

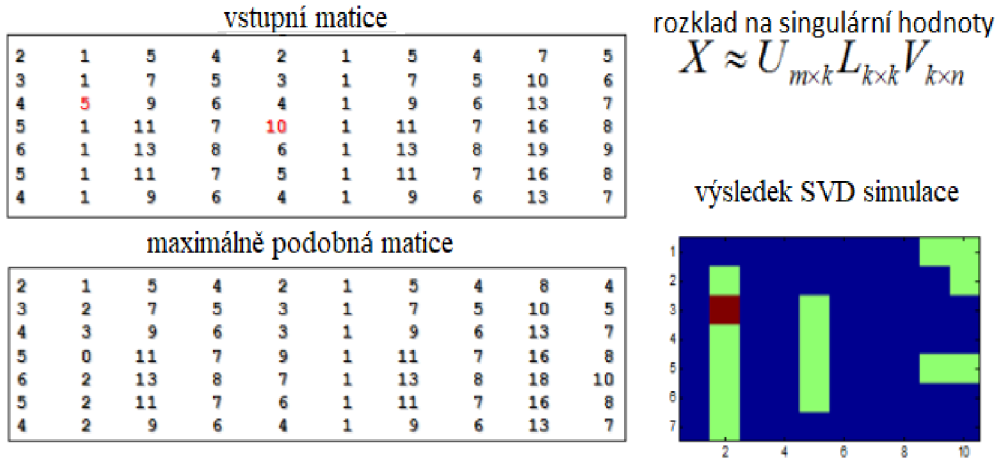
Každý objekt reálného světa můžeme popsat jeho pozicí v mnohorozměrném prostoru, v extrémním případě jde až o desetitisíce dimenzí.

Existuje cela řada metod detekce anomálie ve vícerozměrných datech:

- Metody založeny na modelování
- Iterační metody
- Metrické metody
- Metody strojového učení

1) Modelový přístup

Smysl těchto metod je jednoduchý – navrhujeme model, který popisuje vstupní data. Body, které se výrazně liší od modelu, jsou anomáliemi (Obr. 3.6). Taková metoda je dobrá pro detekce neobvyklých stavů (Novelty detection), ale méně vhodná pro hledání odlehlých hodnot. Ve skutečnosti při nastavení modelu používáme data obsahující anomálie a pak se nastavují a upravují se podle ně.

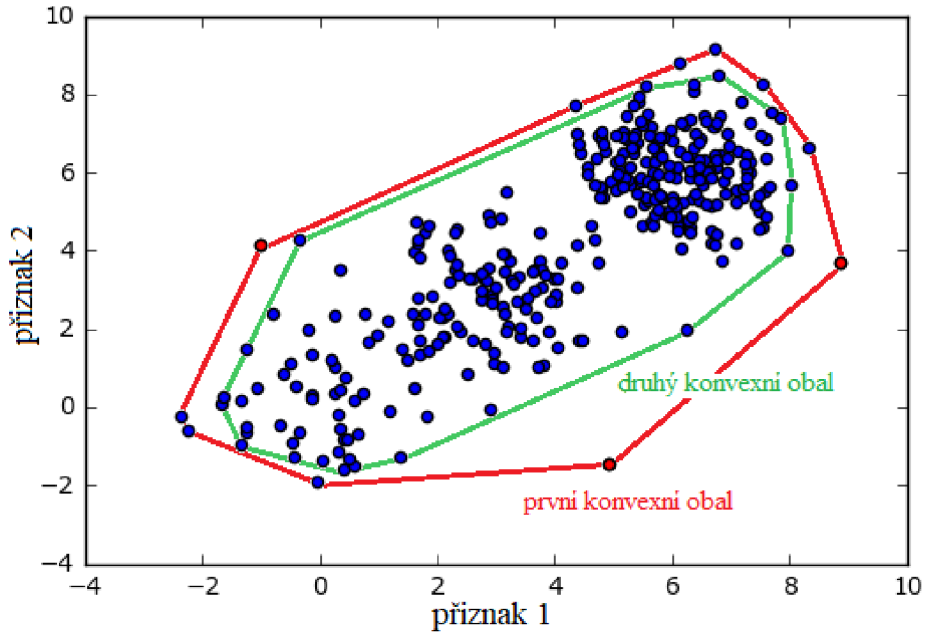


Obr. 3.6: Modelový přístup detekce anomálie v datech.

Na obrázku 3.6 je vidět použití modelového přístupu. Na vstupu je matice a je nutné v ní zjistit outliery. Použijeme rozklad na singulární hodnoty (SVD, singular value decomposition) k nalezení maximálně podobné matice. Prvky, které se velmi liší od odpovídajících prvků nalezené matice, jsou považovány za anomálie. Metoda je podrobně popsaná v [18].

2) Iterační metody

Pojem iterace znamená opakování, nebo též opětovné použití konkrétních operací na výsledcích operace předchozí. Iterační metody jsou metody, které se skládají z iterací, na kterých se odstraní skupina zvláště podezřelých objektů [19]. Například v n-dimenzionálním charakteristickém prostoru je možné odstranit konvexní obal bodů za předpokladu, že to jsou outliery (Obr. 3.7).



Obr. 3.7: Vizualizace iterační metody.

Pro vysoce dimenzionální data (např. P >= 10000) využití iteračních metod a modelování je velmi náročné.

3) Metrické metody

Metrické metody jsou nejpopulárnějšími metodami v praxi. Tyto metody jsou založeny na využití některé metriky vzdálenosti, která umožnuje nalezení anomálie. Je intuitivní jasně, že outlier nemá hodně sousedů, proto vhodnou mírou anomálie může sloužit vzdálenost mezi body či vzdálenosti od nejbližších sousedů (viz metoda Local Outlier Factor) [20].

Vzdálenost a podobnost jsou konkrétní hodnoty vyjadřující vztah dvou bodů v prostoru, které definujeme pomocí konkrétního algoritmu, tzv. metriky. Vzdálenost může být definována jako míra nepodobnosti. Čím je vzdálenost mezi dvěma body větší, tím méně jsou si podobny. Výběr metriky vzdálenosti závisí vždy na řešené úloze. Volba konkrétní metriky záleží na několika kritérií, jako např. kvalita výsledků klasifikace, výpočetní nároky, charakter rozložení dat, apod. Obecně nelze doporučit vhodnou metriku pro určité standardní situace, zvolená metrika může ovlivnit výsledky analýzy. Proto je její výběr velice důležitý.

Euklidovská vzdálenost je nejvýznamnější metrika vzdálenosti. Je to geometrická vzdálenost objektu stejně, jako Pythagorova věta počítá přeponu pravoúhlého trojúhelníku.

$$d_E = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \,, \tag{3.11}$$

kde d_E euklidovská vzdálenost dvou bodů x_i a y_i.

Tato metrika má své vlastní nevýhody. Je citlivá na rozsah hodnot vstupních proměnných a také nebere v úvahu to, že některé proměnné jsou na sobě závislé.

Hammingova metrika Manhattan je součet rozdílů jednotlivých proměnných popisujících objekty.

$$d_{EW} = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}\,, \tag{3.12}$$

Výhodou této metriky je nižší výpočetní nároky než u Euklidovy metriky – použití v úlohách s vysokou výpočetní pracností.

Minkowského metrika je zobecnění Euklidovy a Hammingovy metriky. Volba metriky záleží na míře důrazu – čím větší metrika, tím větší váha na velké rozdíly mezi příznaky.

$$d_k = \sqrt[k]{\sum_{i=1}^{n} |x_i - y_i|^k} \tag{3.13}$$

Další metrikou je **Mahalanobisová vzdálenost**. Jde o obecné měřítko vzdálenosti beroucí v úvahu korelaci mezi parametry a není závislá na rozsahu hodnot parametrů (Mahalanobis 1936). Mahalanobisová vzdálenost mezi vícerozměrnými vektory $x = (x_1,...,x_n)^T$ a $y = (y_1,...,y_n)^T$ n dimensionálního prostoru je definována následovně:

$$d_M = ((x-y)^t C^{-1}(x-y))^{\frac{1}{2}}, \tag{3.14}$$

kde C je matice kovariance. Oproti euklidovské metrice tato metoda není závislá na variabilitě proměnných. Takže Mahalanobisová vzdálenost se používá pro detekce outlierů. Bod, který má největší vzdálenost od ostatních bodů v množině, je považován za odlehlou hodnotu.

Pro přesnější interpretaci výsledků v teto práce byla použita další metrika **Kosinová podobnost**. Kosinová podobnost (cosine similarity) je míra podobnosti dvou vektorů, která je založena na výpočtu kosinu úhlu těchto vektorů:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}\,, \tag{3.15}$$

kde A a B jsou dva vektory délky n. Rozsah hodnot pro tuto funkci je [−1, +1],

(-1) znamená přesně opačný směr, 0 je nezávislost a (+1) – naprostou shodu.

Samozřejmě existují i další metriky vzdálenosti či míry podobnosti, ale v této práce budeme se zabývat třemi: Euklidovská, Mahalanobisová vzdálenost a Kosinová podobnost.

4) Metody strojového učení

V současné době existuje několik algoritmů detekce anomálie založených na strojovém učení např. OneClassSVM (support vector machines), Isolation Forest a Elliptic Envelope.

OneClassSVM je obvyklý SVM algoritmus [21], který odděluje datový vzorek od začátku souřadnice. Objekty nacházející se blízko začátku souřadnice budou

považovány za anomálie. Tento algoritm má svou realizace v Python knihovně scikit-learn [22].

Isolation Forest je jednou z variace náhodného lesu (Random Forest). Isolation Forest se skládá ze stromů, kde každý strom je konstruován do konce datového vzorku. Pro tvoření větve ve stromu se náhodně vybírá příznak a štěpení. Pro každý objekt mírou jeho normality je aritmeticky průměr hloubky listu, ve kterém leží objekt. Logika algoritmu je jednoduchá, při náhodné generace stromů, anomálie padnou do listů v raných fázích (malá hloubka stromu) [23].

Nicméně většina těchto metod potřebuje vědět přesný počet anomálie či procento výskytu anomálie v datech. Problémem je v tom, že velmi často nejsou informace o tom, kolik anomálie data obsahují. Což znamená, že tyto algoritmy nejsou vhodné pro řešení problému detekce outlierů v datech s neznámým počtem anomálie.

4 HODNOCENÍ ÚSPĚŠNOSTI KLASIFIKACE

V předchozí kapitole už byly popsány metody klasifikace dat dělící se na normální a anomální data. Pro určení kvality detekce anomálii si představíme jednotlivé míry hodnocení úspěšnosti těchto klasifikace. Dále se budeme zabývat rozdělením datové sady na trénovací a testovací data pro správné hodnocení úspěšnosti klasifikace.

V praxe často ne vždy je k dispozici nezávislá datová sada k otestování natrénovaného klasifikátoru. V tomto případě je potřeba daný datový soubor správně využít pro natrénování a testování klasifikátoru. Celkem existuje 4 základní přístupy pro provedení tohoto otestování:

1. resubstituce (resubstitution)
2. náhodný výběr s opakováním (bootstrap)
3. predikční testování externí validací (hold-out)
4. křížová validace (cross-validation)

Trénovací data jsou data, které se používají pro natrénování klasifikátoru, zatímco definice testovacích dat ukazuje otestování úspěšnosti klasifikátoru. Reálná situace prozatím vypadá tak, že obvykle neexistují dvě nezávislé datové sady (např. sady pacientů se schizofrenií z různých měst). Kvůli tomu je třeba vhodným způsobem rozdělit tento datový soubor na skupiny testovacích a trénovacích dat. A protože princip hodnocení úspěšnosti klasifikace je stejný pro všechny případy, tady budeme uvažovat termíny „trénovací“ a „testovací“ data nehledě na to, budeme-li mít jeden nebo dva datové soubory.

Při **resubstituci** (resubstitution) jsou použita stejná data jak pro trénování, tak i pro otestování. Tento přístup se odlisuje od jiných jednoduchostí a rychlostí provedení, ale zároveň vede k moc nadhodnoceným výsledkům klasifikace. Pokud by se dále chtělo použít tento klasifikátor na novém datovém souboru, tak mohlo by dojít k velmi nízké úspešnosti klasifikace, kvůli tomu, že při resubstituci jsme měli vysokou úspěšnost a mysleli si, že náš klasifikátor je dobrý. Ale kdyby klasifikátor byl zase použít pro tento stejný soubor, došlo by k přeučení, tj. klasifikátor dokonale provádí klasifikaci natrénovaných dat, ale úplně selhává při klasifikaci nových.

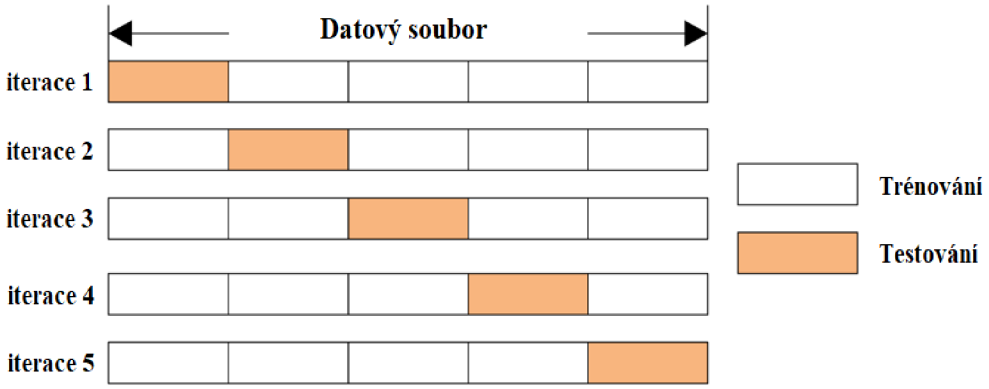
Druhý přístup je **náhodný výběr s opakováním** (bootstrap), který je založen na N-krát provedeném náhodném opakovaném výběru subjektů z původního datového souboru (má N subjektů) [24]. Tyto subjekty se použijí jako testovací sada. Ostatní subjekty, které nebyly vybrány ani jednou, se používají pro testování. Pokud jsou k dispozici rozumně velká data tak kolem 63,2% subjektů se používají pro učení klasifikátoru
a 36,8% subjektů – pro testování. Tento přístup má dvě hlavní výhody: první je to, že trénovací sada je stejně velká jako původní datový soubor, druhá je rychlost provedení oproti např. křížové validaci. Zatímco nevýhoda spočívá v opakujících se v trénovací sadě subjektech.

Třetí přístup je **testování externí validací** (hold-out), to znamená, že data se rozdělí na dvě části (zpravidla jedna třetina ku dvěma třetinám). První část sebou představuje data „odložena stranou“ pro testování, druhá – zbývající část dat pro učení klasifikátoru. Subjekty v nezávislé testovací a trénovací sadě se neopakují, což je hlavní výhodou tohoto přístupu. Zároveň je třeba uvést, že je málo dat pro trénování i testování a výsledek klasifikace je příliš závislý na výběru trénovacích dat.

Obr. 4.1: Rozdělení datového souboru na trénovací a testovací sadu.

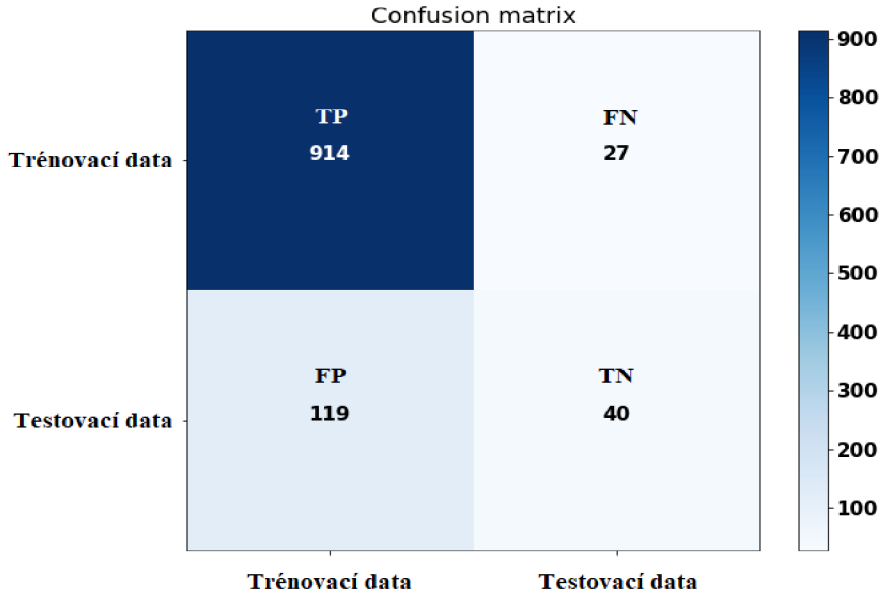
Proto bylo vyvinuto několik různých modifikaci testování externí validací. Například je možné použít část dat (obvykle polovinu) pro trénování a zbytek (polovinu) pro testování, následně přehodit testovací a trénovací sadu a výsledky těchto dvou klasifikací zprůměrovat. Nevýhodou v tomto případě je používání malých souborů a následkem je to, že polovina dat při rozdělení bude pro trénování příliš málo. V reálné praxi se častěji používá druhá modifikace, kdy se soubor R-krát náhodně rozdělí na trénovací a testovací sadu (většinou se použijí dvě třetiny pro trénování a třetina pro testování). Získané R výsledky klasifikací se následně zprůměrují. Výhodou spočívá v relativně přesném odhadu úspěšnosti klasifikace a v použití relativně velké části subjektů na trénování, zatímco nevýhodami jsou časová náročnost procesu a překrývání trénovacích a testovacích sad.

Problém překrývání testovacích sad při testování externí validací s R opakováními je možné řešit pomocí **K-násobné křížové validace** (taky může být označena jako K-násobná příčná validace, angl. K-fold cross-validation) [25]. Přičemž datový soubor se rozdělí na K částí, kdy jedna část je použita pro testování, a ostatní částí jsou použity pro trénování. Postup se opakuje pro každou část (Obr. 2). Výhodou tohoto přístupu je relativně přesný odhad úspěšnosti klasifikace, zatímco nevýhodou je časová náročnost.



Obr. 4.2: Rozdělení datového souboru na trénovací a testovací sady při k-násobné křížové validace (k = 5)

Než začneme analyzovat samotné míry klasifikace, je třeba vědět skutečné správné zařazení subjektů čili objektů do určitých tříd. V oblasti strojového učení a zejména podoblasti problematiky statistické klasifikace existuje specificky uspořádána tabulka tzv. matice záměn (angl. Confusion matrix), která umožňuje vizualizace výkonu algoritmu klasifikace nebo jiného algoritmu strojového učení (Obr. 4.1) [26].



Obr. 4.3: Matice záměn (angl. Confusion matrix).

Každý řádek matice představuje sebou předpovídané třídy (Predicted Class), zatímco každý sloupec – skutečné třídy (Actual Class) nebo naopak. Dále budeme uvažovat trénovací data za normálně zadané pacienty a testovací data za pacienty obsahující anomální záznamy.

Takže matice záměn zahrnuje:

1. Počet skutečně pozitivních výsledků (TP – True Positive), tj. počet pacientů správně klasifikovaných jako normální.
2. Počet falešně pozitivních výsledků (FP – False Positive), tj. počet pacientů chybně klasifikovaných jako normální.
3. Počet falešně negativních výsledků (FN – False Negative), tj. počet pacientů chybně klasifikovaných jako anomálie.
4. Počet skutečně negativních výsledků (TN – True Negative), počet anomálie správně klasifikovaných jako anomální záznamy.

Z matice záměn vyplývá, že chyby klasifikace spadají do dvou typů:

1. Falešně negativní (False Negative).

2. Falešně pozitivní (False Positive).

Pokud známe hodnoty matice záměn, tak následně lze získat míry hodnocení úspěšnosti klasifikace testovacích dát.

Celková přesnost (Accuracy) je to procento správných odpovědí algoritmu, tzn. poměr správných předpovědi k celkovému počtu testovacích subjektů.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4.1}$$

Celkovou přesnost lze snadno změnit na míru chybného zařazení nebo chybovost obrácením/změnou hodnoty. Takže chybu lze spočítat jako poměr chybně klasifikovaných subjektů ke všem subjektům.

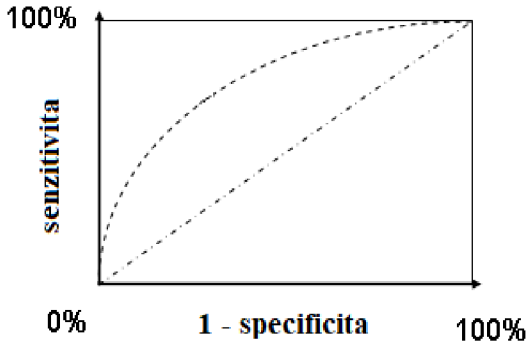
Další mírou je senzitivita / citlivost (Sensitivity) je poměr správně klasifikovaných subjektů (například, procento nemocných, kteří jsou správně označeny jako nemocní) ke všem testovacím subjektům.

$$TPR = \frac{TP}{TP + FN}, \tag{4.2}$$

Specificita / specifičnost (Specificity) je poměr správně klasifikovaných negativních výsledků, které jsou správně identifikovaní (například procento zdravých lidi, kteří jsou správně označení jako zdraví pacienty) ze všech zdravých pacientů.

$$TNR = \frac{TN}{TN + FP}, \tag{4.3}$$

Křivka ROC (angl. Receiver Operating Characteristic) je nástrojem pro hodnocení a optimalizaci binárního klasifikačního systému (testu), který ukazuje vztah mezi specificitou a senzitivitou daného testu nebo detektoru pro všechny přípustné hodnoty prahu.



Obr. 4.4: ROC křivka.

Pokud klasifikace bude správná (všechny objekty zařáděny správně), a tedy FP, FN chyby budou nulové, dostaneme ideální ROC křivku, která kopíruje okraj ROC prostoru, a sice nejdříve z bodu (0,0) do bodu (0,1) a následně do bodu (1,1).

5 APLIKAČNÍ ČÁST

Pro provedení jakoukoli analýzy je třeba se nejprve seznámit s registrem, tedy zjistit, o čem daný registr je a provést analýzu jednotlivých datových typů.

Pro zpracování algoritmu byla použita data ze tří klinických studií. Kvůli tomu, že této klinické studie jsou utajeny, tak budou pojmenovány jako studie 1, studie 2 a studie 3.

Studie 1 je soubor observačních kohortních studijních registrů (RWD) a výsledky dospělých pacientů se symptomatickou anémií související s chronickým selháním ledvin u pacientů na dialýze, kteří nedávno zahájili léčbu darbepoetinem-α (CRESP®) a pomocí dalších často používaných alternativních stimulujících erytropoézu agentů (ESAs). V této studii jsou 400 pacientů, z nichž 200 pacientů jsou na dialýze a 200 pacientů nejsou na dialýze. Doba trvání účasti pacienta: od výchozího stavu až do úmrtí, odběr pacienta, ztráta následků, transplantace ledvin nebo ukončení studia, podle toho, co nastane dříve.

Cílem této studie je porovnat retenční čas u dospělých pacientů se symptomatickou anémií spojenou s chronickým onemocněním na dialýze a těch, kteří nejsou na dialýze, kteří zahájili léčbu v posledních třech měsících buď CRESP®, Epogen ™ (epoetin-α originator), Neorecormon ™ (epoetin beta originator) anebo Mircera ™ (pegylated epoetin beta originator) a taky porovnat dopad léčby na individualizovaný podíl vzorků, které jsou mimo cílový rozsah.

Studie 2 je registr zaměřený na diagnózu pacientů s chronickou myeloidní leukemií, což znamená, že se sledují pacienty v docela dlouhém časovém horizontu, protože cílená léčba na konkrétní chromozomální změnu je velice účinná, nicméně ani ona se zatím neobejde bez nežádoucích účinků. U pacientů se hodnotí léčebné odpovědi – hematologická, cytogenetická a molekulární a to především ze začátku nasazení každé léčby – nejdříve ve 3., 6., 12. a 18. měsíci, pokud pacient v těchto měsících neodpovídá, léčba se změní, pokud odpovídá a nemá komplikace, stačí ho později sledovat už jen každý další rok (kontroly po době 12 měsíců). V současné době jsou zapojena 4 česká centra a zahrnují celkem více než 900 pacientů.

V hledáčku analýz jsou nyní především inhibitory tyrosin kináz – tj. **imatinib** (první existující svého druhu, v letošním roce se skončil jeho desetiletý patent, takže má konečně svoje generika, to znamená léky se stejnou účinnou látkou, ale jiným obchodním názvem, která výrazně sníží jeho cenu na trhu), **dasatinib** (většinou až pro druhou linii léčby), **nilotinib** (na podobném principu jako **imatinib**), **ponatinib** (velice drahý, používá se u pacientů, kteří skoro na nic neodpovídají) a **bosutinib** (nový, ale zatím ne až tak úspěšný). Pacienti mohou být stále ve výjimečných případech transplantováni.

Stěžejní je formulář **Patient Status and Treatment Overview**, kde jsou jednotlivé léčebné linie a jejich délka – podle nich se napojuji sledovací formuláře **Follow-up**, kde je možno dohledat jednotlivé léčebné odpovědi (hematologická, cytogenetická, molekulární), ale i laboratorní nehematologické nežádoucí účinky (z biochemie) a hematologické laboratorní účinky (neutrofily, hemoglobin, trombocyty). Nežádoucí

účinky se pak hodnotí podle gradu ve srovnání s tím, jaký grade měl pacient na začátku dané léčby (tedy srovnává se se startem léčby). Kromě toho jsou ještě klinické nehematologické nežádoucí účinky – ty jsou uvedeny ve formuláři zvlášť.

Při analýzách se sledují časy do úmrtí nebo do progrese, ale taky i do dosažení jednotlivých léčebných odpovědí. Zvláštní událostí bývá ztráta odpovědi, kterou je potřeba taky zaznamenávat – všechny události se uvádějí na jednotlivých **treatmentových formulářích – Imatinib treatment**, **Dasatinib treatment**, kde se uvádí souhrn dané linie léčby.

Na formuláři **Diagnosis** jsou klíčové údaje jako věk, krevní obraz, přesah sleziny přes žeberní oblouk, z nich se vyhodnocuje rizikové skóre, se kterým pacient vstupuje do sledování.

Z hlediska analýz je velice důležité, aby pacienti měli správně vyplněné linie léčby (pořadí) a typ léčby (pacienti mohou mít i tzv. předléčbu, která se příliš nehodnotí). Pomocí linie a typu léčby se na sebe navzájem napojují jednotlivé formuláře, je tedy nezbytné, aby tyto údaje byly správně vyplněné.

Studie 3 je neintervenční klinické hodnocení popisující jak pacienti vnímají antikoagulační léčbu a léčebný komfort spojený s léčbou pro prevenci cévní mozkové příhody u nechlopňové fibrilace síní (NVAF). Je to největší studie ze všech třech a má kolem 9000 pacientů.

Pacienti budou zařazeni do jedné z následujících kohort:

- Kohorta A: Pacienti, kteří v současné době užívají antikoagulační léčbu.
- Kohorta B: Pacienti, jimž byla nově diagnostikována fibrilace síní a kteří začnou užívat antikoagulační léčbu.

Pacienti budou monitorováni během sledovaného období 6 měsíců. Data budou shromažďována ve třech časových bodech:

1. Ve výchozím stavu, tj. zahájení léčby přípravkem Pradaxa® nebo VKA (Baseline)
2. 30 – 45 dnů po zahájení antikoagulační léčby.
3. 150 – 210 dnů po zahájení léčby (pokračovací období).

Primárním cílem studie je popsat, jak pacienti s nechlopňovou fibrilací síní (NVAF) vnímají léčbu pomocí dotazníku vnímání antikoagulační léčby PACT-Q® (Perception on Anticoagulant Treatment Questionnaire).

Hlavním účelem vytvořeného algoritmu bylo zjištění anomálie v každé studie.

Postup pro analýzu byl následující:

- Import datového souboru z databáze.
- Předzpracování dat.
- Jednorozměrná statistická analýza.
- Vícerozměrná metoda klasifikace dat.
- Hodnocení úspěšnosti klasifikace.

5.1 Příprava dat pro analýzu

Import datového souboru je zpravidla prvním krokem zpracování dat. Uvedený v kapitole 1 informační systém Clade IS obsahuje vestavěnou funkce kompletního importu dat z databáze. Výstupem kompletního importu je Excel soubor obsahující veškerou informace uváděnou v databázi včetně popisu všech subjektů a parametrů.

Pro můj vlastní případ využití kompletního importu lze dokonce označit za nevhodné, protože import Excel souboru do prostředí Python a následujícího předzpracování, filtrace i poskládaní všech subjektů a parametrů ze všech Excel listů do jedné matice je velmi náročná práce. Proto byl použit jiný přístup, který byl založen na přímém spojení PostgreSQL databáze a jazyku programovaní Python. Spojení bylo dosáhnuto pomocí 'psycopg2' balíčku, který umožnuje využití PostgreSQL příkazu v Python, čímž šetři čas předzpracování.

Na vstupu algoritmu je databáze KS viz tabulka č. 5.1, která obsahuje data různých typů: celá čísla, čísla s plovoucí čárkou, logické hodnoty (bud' true anebo false), znaky a text, datum a čas. Proto před začátkem statistické analýzy je třeba transformovat původní data na data numerického typu. Každý datový typ transformujeme svým vlastním způsobem.

- Celá čísla převádíme na čísla s plovoucí čárkou (1 – 1.0).
- Logické hodnoty (boolean) transformujeme na číslice (true – 1, false – 0).
- Znaky a text transformujeme selektivně, výběrové texty (číselníky) transformujeme na číslice (např. využity inhibitor tyrosin kináz: imatinib – 1, dasatinib – 2, nilotinib – 3, ponatinib – 4, bosutinib - 5). Volně zadané texty odstraníme kvůli tomu, že jejích interpretace je velice složitá (každý lékař píše své vlastní poznámky a jmenuje nemoci či jiné naměřené parametry podle svého vlastního představení).
- Datum a čas taky transformujeme na data numerického typu. Vypočítáváme počet dnů či hodin od začátku našeho letopočtu.

Tab. 5.1: Vstupní datová sada po SQL reportu.

Patient_id	Q9	Q89	Q3	Q5	Q6	Q7
DAR-0000003	2015-06	Diploma In Electronics	29.06.2015	59.5	false	52
DAR-0000004	2014-02	Diploma in Mechanical Engineering	01.07.2015	57	false	31
DAR-0000005	2015-01		15.07.2015	60	false	56
DAR-0000006	2014-01		25.06.2016	65	true	49
DAR-0000007	2016-01		01.07.2016	55	true	21
DAR-0000008	2016-05	B.A	01.07.2016	46.5	true	38
DAR-0000009	2016-05		04.07.2016	52	true	73
DAR-0000010	2016-06	Diploma in electrical engineering	13.07.2016	70.6	true	48
DAR-0000011	2014-05		26.07.2016	60	true	78

Dalším krokem je odstranění prázdných hodnot (nul) nebo Not a Number (NaN) hodnot. Již v tomto kroku lze předpokládat, že datové vzorky, které mají více než 80% NaN hodnot, mohou být anomáliemi. Proto tento algoritmus zapisuje identifikační číslo (ID) potenciálního anomálního pacienta do výstupní tabulky. Takže lze odstranit parametr z datové sady, jestli jeho hodnota je stejná u všech pacientů, protože její vliv při klasifikace bude stejný pro každého pacienta.

Většina statistických testů nemůže zpracovávat velmi malé vzorky ($n < 3$), takže ony se odstraňují a nepodílejí se na další analýze. A zároveň se zapisují do výstupní tabulky jako příliš malé vzorky, což taky může byt anomálii.

Posledním krokem předzpracování dat je škálování, procedura stanovení šířky intervalů mezi jednotlivými proměnnými (od 0 do 1).

Výstupem předpracování je matice naměřených parametrů viz tabulka č. 5.2. Každý řádek této matice odpovídá vektoru otázek (parametrů) u jednoho pacienta, a každý sloupec je vektorem naměřených hodnot či parametrů u všech pacientů.

Tab. 5.2: Příklad výstupních dat po předzpracování

Patient_id	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
DAR-0000003	0.53	0.21	0.47	0.07	1.0	0.0	0.005
DAR-0000004	0.31	0.19	0.51	0.01	1.0	0.0	0.005
DAR-0000005	0.57	0.21	0.49	0.06	0.0	0.0	0.005
DAR-0000006	0.5	0.25	0.35	0.02	0.67	1.0	0.004
DAR-0000007	0.21	0.17	0.43	0.02	0.34	1.0	0.003
DAR-0000008	0.39	0.1	0.4	0.03	1.0	1.0	0.001
DAR-0000009	0.75	0.15	0.43	0.0	0.0	1.0	0.005
DAR-0000010	0.49	0.3	0.55	0.16	1.0	1.0	0.01
DAR-0000011	0.8	0.21	0.43	0.01	0.33	1.0	0.003
DAR-0000012	0.25	0.04	0.38	0.02	0.33	1.0	0.005
DAR-0000013	0.65	0.23	0.43	0.12	1.0	0.0	0.005
DAR-0000014	0.88	0.25	0.51	0.02	0.33	1.0	0.003
DAR-0000015	0.3	0.13	0.41	0.0	1.0	0.0	0.005
DAR-0000016	0.38	0.18	0.34	0.04	0.66	1.0	0.001

Funkce SQL importu, filtrace a transformace dat je uvedena v příloze.

Následující kapitola popisuje princip samotného algoritmu pro detekce odlehlých hodnot.

5.2 Jednorozměrná statistická analýza

Tato analýza slouží pro další (praktické) seznámení s registry. Zde budou využity dvě metody pro detekci odlehlých hodnot v datech:

- test na násobek směrodatné odchylky
- MAD test, detailní popis těchto testu je uveden v kapitole 3.2.

Test na násobek směrodatné odchylky potřebuje, aby data měla normální rozdělení. Proto před výběrem vhodné metody je nutné zkontrolovat, zda má určitý vzorek dat normální rozdělení. Pro tento účel algoritmus využívá dva statistické testy pro určení normality.

První test je Shapirův-Wilkův, který se používá pro příliš malý objem dat ($3 < n < 30$). Druhý je Kolmogorovův - Smirnovův test – pro testování hypotézy o normálním rozdělení u velkých datových vzorků ($n > 50$). Podrobný popis těchto testů byl uveden v teoretické části. Oba testy mají implementaci v programovacím jazyku Python v knihovně "SciPy„ (knihovna, která je určena k provádění vědeckých a technických výpočtů). Úroveň statistické významnosti pro oba testy byla stanovena za alfa = 0,05. Pokud je p-hodnota testu menší než alfa, to znamená, že data nemají normální rozdělení.

Typ distribuce dat je důležitým kritériem při výběru metody detekce outlierů. Jestli přijatá data mají normální rozdělení, tak ke stanovení outlierů se použije test na násobek směrodatné odchylky, v opačném případě – MAD test.

Výše popsaný algoritmus je navržen pro filtrace vstupní datové sady a odstranění outlierů pro trénování algoritmu klasifikace (viz další kapitola). Tato práce je velice náročná, protože je třeba procházet každý parametr zvlášť a kontrolovat normalitu.

Realizace tohoto algoritmu je uvedena v příloze.

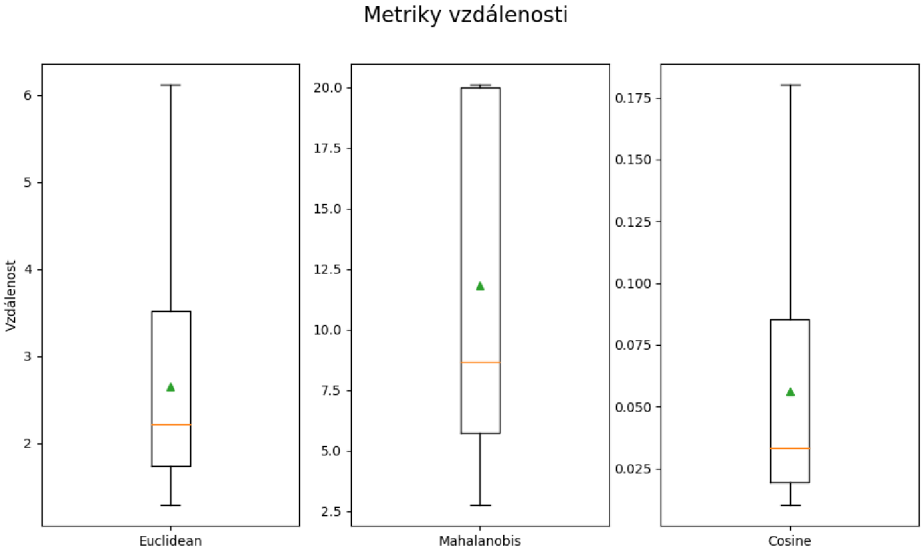
5.3 Algoritmus klasifikace

Původním cílem tohoto algoritmu bylo detekovat anomálie ve vícerozměrné datové sadě, bez ohledu na dimenze vstupního souboru dat. Realizovaný metod detekce anomálie využívá základní koncepce zdrojového učení (machine learning): trénování algoritmu na původním souboru dat a testování či využiti získaného modelu na nové datové sadě.

Postup analýzy je jednoduchý. Na vstup algoritmus přijme předzpracovaný datový soubor, kde každý řádek popisuje naměřené parametry jednoho pacienta. Dalším krokem je vypočet centroidu a vzdáleností mezi každým řádkem a centroidem pomocí třech metrik. Centroid se počítá jako vektor středních hodnot z každého sloupce (naměřené otázky). Tato práce zahrnuje výpočet vzdálenosti od centroidu pomoci třech metrik vzdálenosti:

- mahalanobisová,
- euklidovská,
- kosinová podobnost.

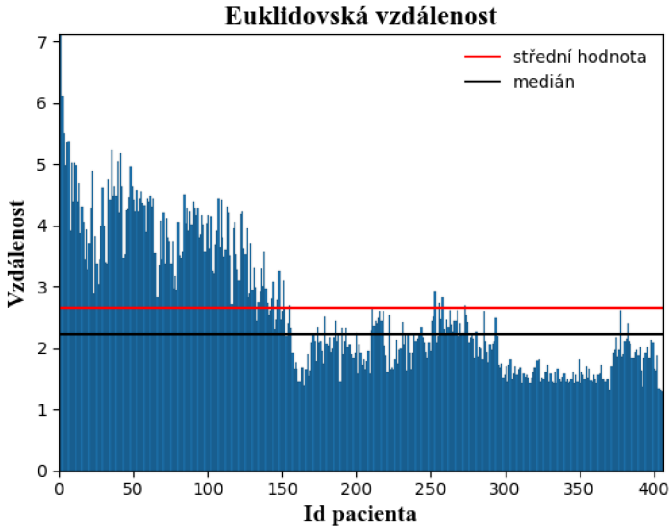
Po výpočtu vzdáleností budeme mít tři vektory. U každého vektoru zjistíme hodnoty přesahující zvoleny percentil. Tyto hodnoty budeme uvažovat za outliery.



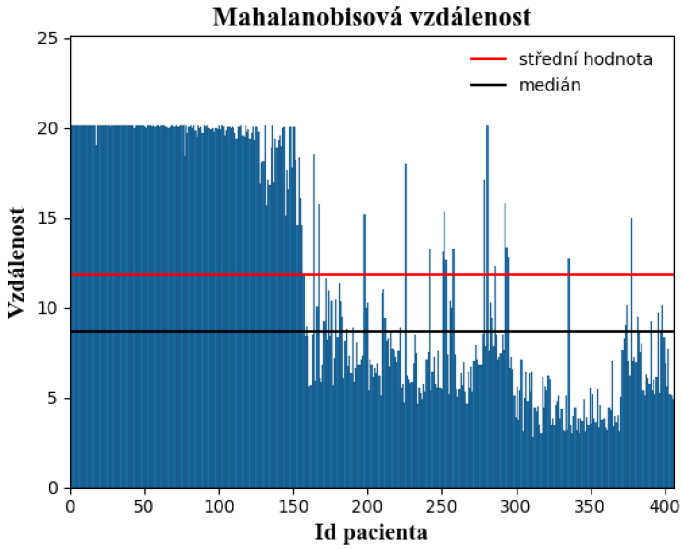
Obr. 5.1: Vzdálenosti mezi jednotlivými subjekty a centroidem.

Na obrázku 5.1 je vidět, že rozdělení euklidovské vzdálenosti většinou je podobné rozdělení kosinové podobnosti. Mahalanobisová vzdálenost má naopak jiny charakter rozloženi. Oranžová čára ukazuje střední hodnotu u každého rozdělení a zelená čára je medián.

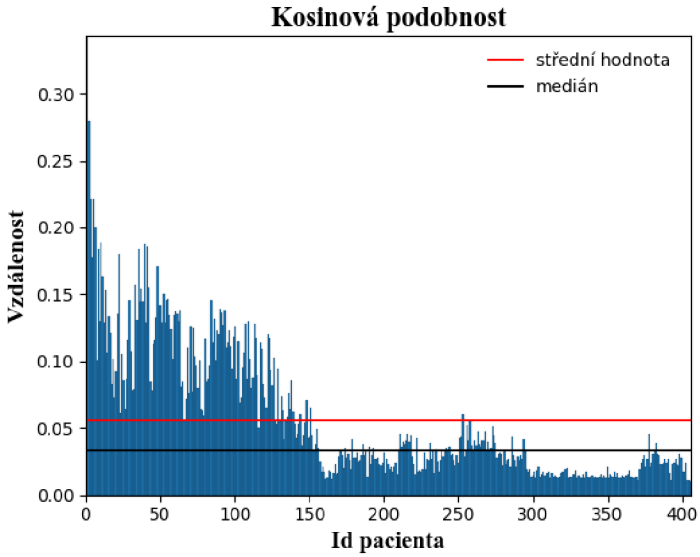
Na dalších grafech (viz. na obrázcích 5.2 – 5.4) je vidět histogramy ukazující hodnoty vzdálenosti pro každý subjekt v datové sade.



Obr. 5.2: Euklidovská vzdálenost mezi jednotlivými subjekty a centroidem.



Obr. 5.3: Mahalanobisová vzdálenost mezi jednotlivými subjekty a centroidem.



Obr. 5.4: Kosinová podobnost mezi jednotlivými subjekty a centroidem.

Podle typu rozdělení vzdálenosti, která je ukázána na obrázku 5.1, kosinová podobnost a euklidovská vzdálenost generují velice podobné výsledky a taky mají příliš shodné typy histogramů (viz. na obrázcích 5.2 a 5.3).

Navíc dá se říct, že data mají svou vlastní statistickou významnost, což nám ukazují výše uvedené grafy. Data se dá rozdělit do dvou hluků:

- prví skupina představuje sebou subjekty s vysokou hodnotou vzdálenosti nad mediánem,
- druhá část – subjekty s hodnotou vzdálenosti pod mediánem.

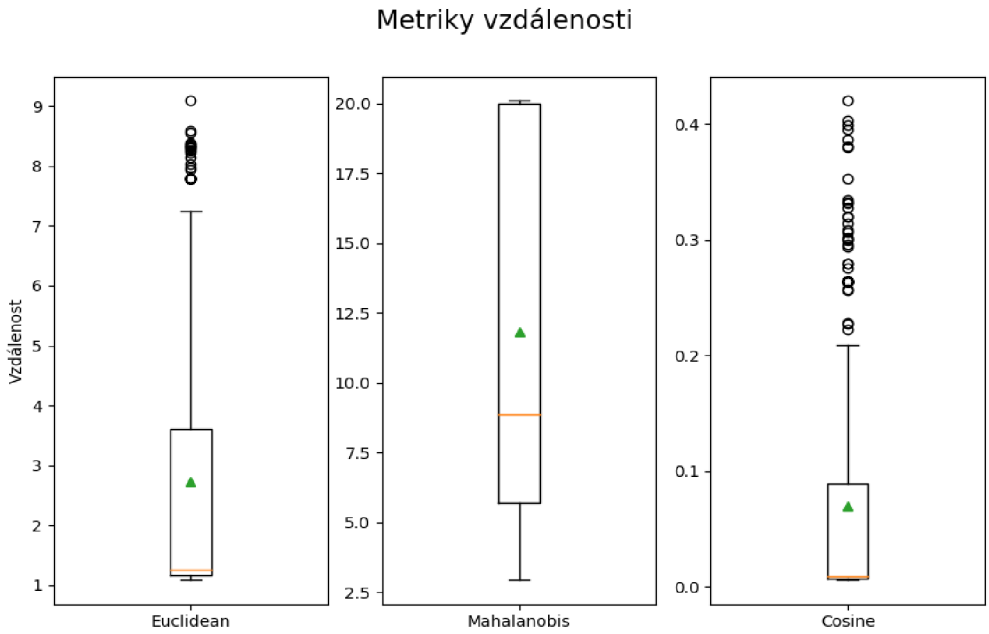
Jak bylo zmíněno výše, tato práce zahrnuje základní koncepci zdrojového učení. Proto algoritmus byl rozdělen na dvě částí: trénování a testování.

5.3.1 Trénování modelu

Pro trénování navřeného algoritmu byly použity dvě metody:

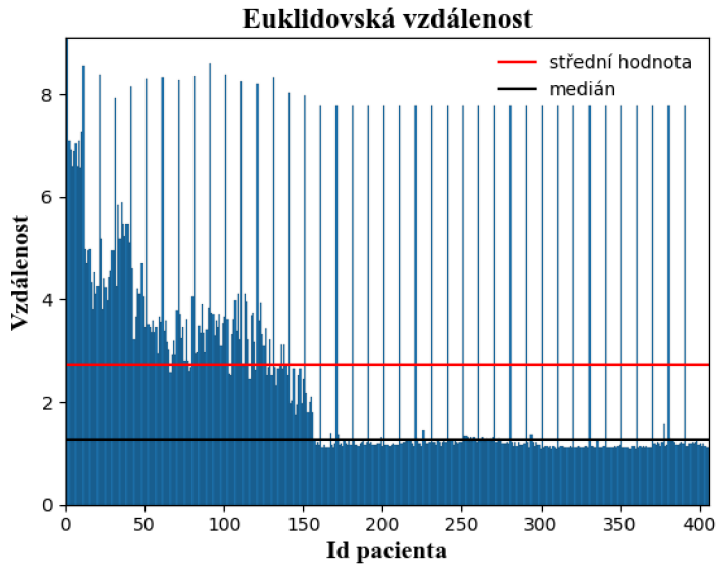
- resubstituce (resubstitution),
- predikční testování externí validací (hold-out).

Při resubstituci byla použita stejná data jak pro trénování, tak i pro otestování. V datové sadě před trénováním bylo třeba vytvořit samotné anomálie, aby bylo možné zjistit kvalitu detekce. Trénování bylo provedeno při čtyřech stupních znečištění vstupních dat (1%, 5%, 10%, 15%). Anomální záznamy byly uměle vytvořeny u náhodně vybraných pacientů. Potom byla spočítána vzdálenost od centroidu a odhadnuty různé kombinace percentilů (od 80% až do 99%) u každé z třech metrik, což jsou 60 kombinaci. Přičemž pro rozšíření této klasifikace byly spočítány další kombinace dvou metrik: shodné anomálie u mahalanobisové a euklidovské metriky, mahalanobisové a kosinové, euklidovské a kosinové, a odlehlé hodnoty se vyskytující zároveň u třech metrik. Celkem bylo spočítáno 140 kombinace, každá z těchto kombinace zahrnuje různý počet odlišných či stejných anomálie. Dalším krokem bylo testování.

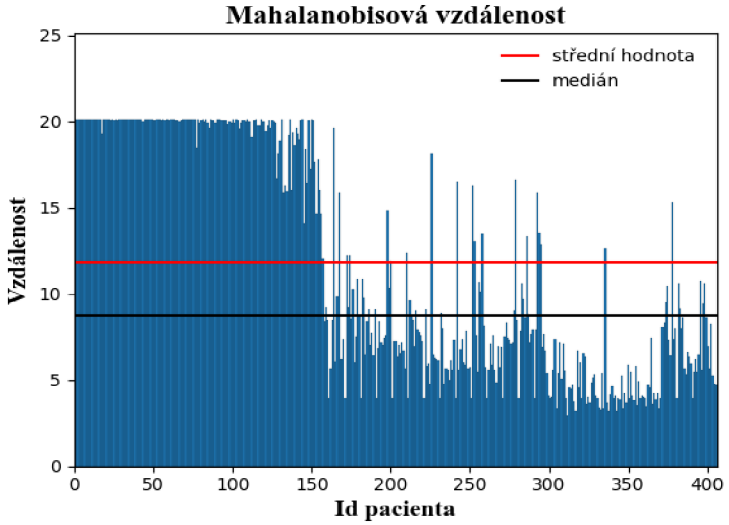


Obr. 5.5: Vzdálenost mezi jednotlivými subjekty a centroidem při 10% znečištění vstupních dat.

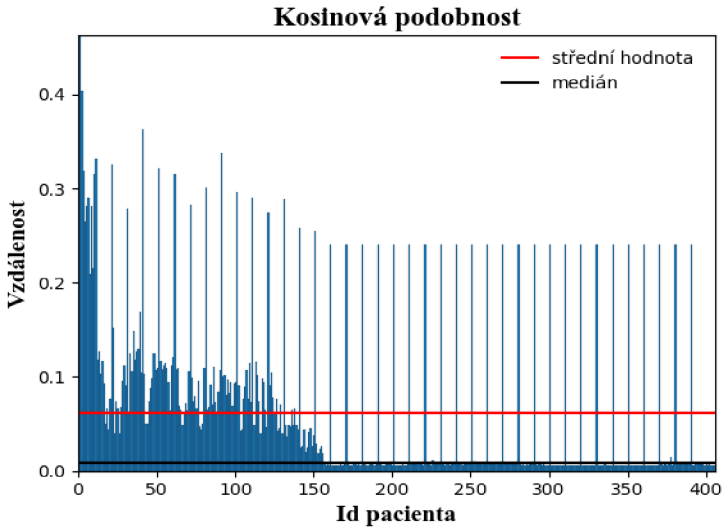
Obrázek č. 5.5 ukazuje rozdělení třech metrik vzdálenosti, které byly spočítány při 10%-m znečištění vstupních dat. Zde je dobře vidět výskyt odlehlých hodnot u dvou ze třech metrik.



Obr. 5.6: Euklidovská vzdálenost při 10% znečištění vstupních dat.



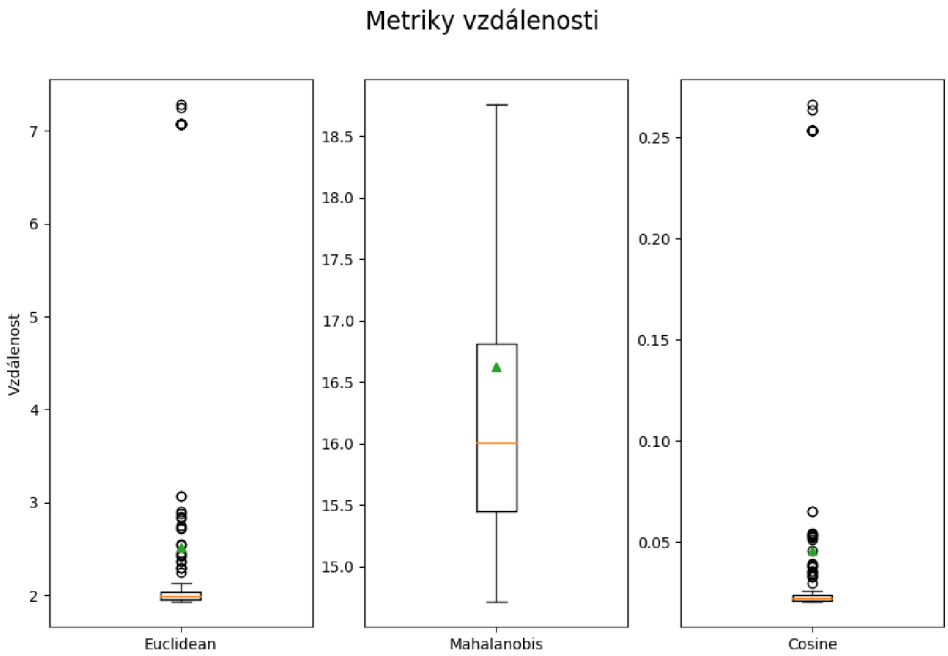
Obr. 5.7: Mahalanobisová vzdálenost při 10% znečištění vstupních dat.



Obr. 5.8: Kosinová podobnost při 10% znečištění vstupních dat.

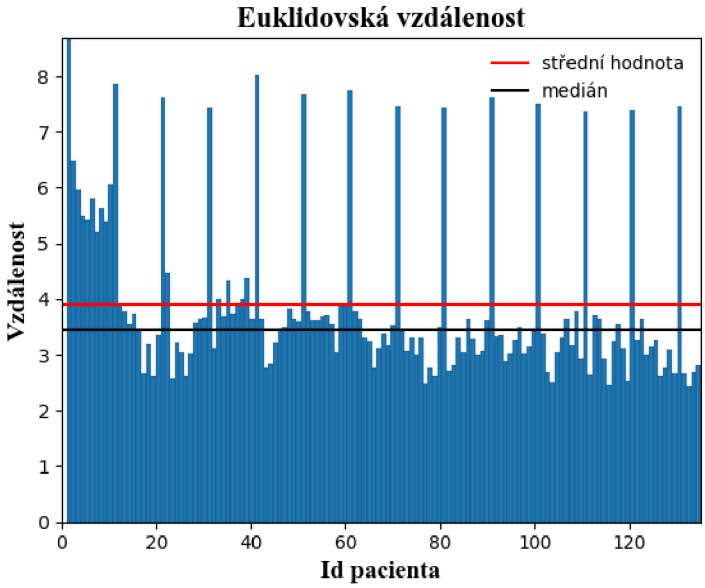
Obrázky č. 5.6 – 5.8 ukazují uměle 10% znečištění datové sady a je dobře vidět výskyt impulzů v histogramů, které jsou outliery. Podle grafů se da říct, že kosinová a euklidovská metriky detekují anomálie lépe, než mahalanobisová metrika. Další část analýzy se zabývá výpočtem odhadu outlierů při různých parametrech percentilu u všech možných kombinaci metrik. Přesnost každé kombinace je uvedena v další kapitole.

Predikční testování externí validací znamená to, že data se rozdělí na dvě části (30% pro trénování a 70% pro otestování). Stejným způsobem byly vytvořeny anomální subjekty ve vstupní datové sadě jak v předchozí metodě. Ale na rozdíl od resubstituce, kde centroid byl vypočten u cele datové sady, zde centroid se spočítá jenom u trénovací sady. Postup další analýzy je stejný jak při resubstituce. Avšak zde je třeba odhadnout kombinace, která dává co nejvíce přibližné výsledky klasifikace k uměle zadaným anomáliím, kde dalším krokem je otestování této kombinace na druhé části datové sady.

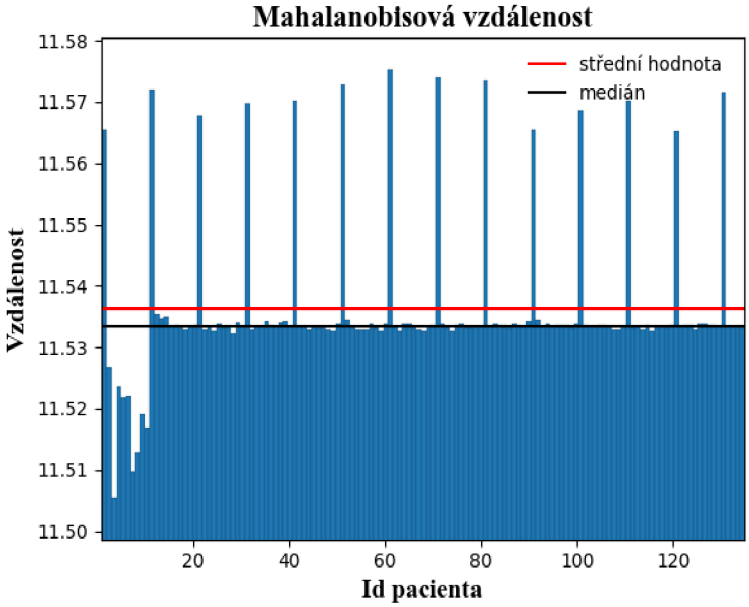


Obr. 5.9: Vzdálenost mezi jednotlivý subjekty a centroidem při 10% znečištění vstupních dat (trénovací datová sada).

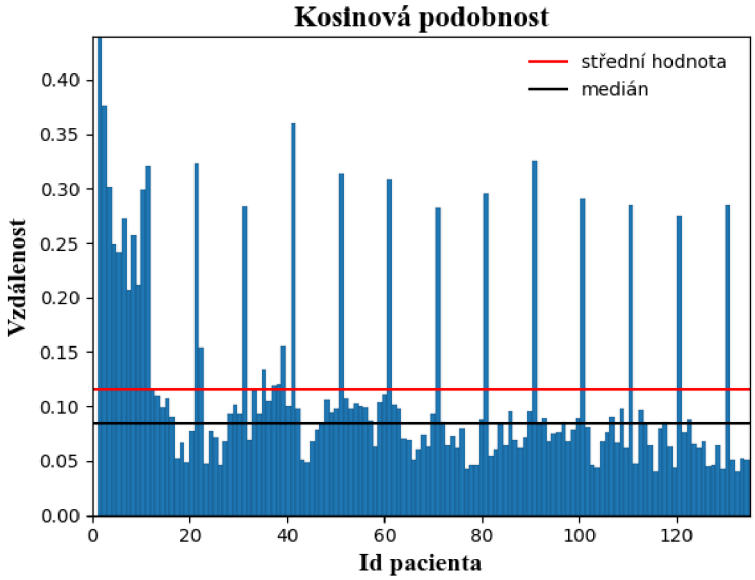
Obrázek č. 5.9 ukazuje rozdělení třech metrik vzdálenosti na trénovacím souboru dat při 10% znečištění. Krabicový grafy Euclidean a Cosine zobrazuje rozdíl mezi extrémními anomáliemi a odlehlými hodnotami, což pro nás není tak důležité, protože chceme odhadnout jak extrémní anomálie, tak i odlehlé hodnoty. Na dalších grafech (5.10 – 5.12) je možné vidět histogramy ukazující hodnoty vzdálenosti pro každý subjekt ve trénovací datové sade.



Obr. 5.10: Euklidovská vzdálenost při 10% znečištění vstupních dat (trénovací datová sada).



Obr. 5.11: Mahalanobisová vzdálenost při 10% znečištění vstupních dat (trénovací datová sada).



Obr. 5.12: Kosinová podobnost při 10% znečištění vstupních dat (trénovací datová sada).

Obrázky č 5.10 – 5.12 mají skoro podobné znázornění jako při resubstituce, kromě toho, že zde je zobrazena jenom 1/3 datové sady. Při detailním pohledu na obrázku č. 5.11 je vidět, že osa y (vzdálenost) má jiny rozsah hodnot, což je způsobeno manuálním zvětšením detailů v obrazu pro lepší pozorování. Jak už bylo zmíněno při resubstituce, kosinová a euklidovská metriky detekují anomálie lépe než mahalanobisová metrika. Proto při testování byl kladen největší důraz na kombinace těchto dvou metrik.

5.3.2 Testování modelu

Testování modelu klasifikace zahrnuje vypočet matice záměn a celkové přesnosti klasifikace. Obecný postup testování se provádí takto:

- Porovnání výsledků s vektorem uměle zadaných anomálie.
- Vypočet matice záměn a přesnosti.
- Zjištění nejlepší přesnosti klasifikace.

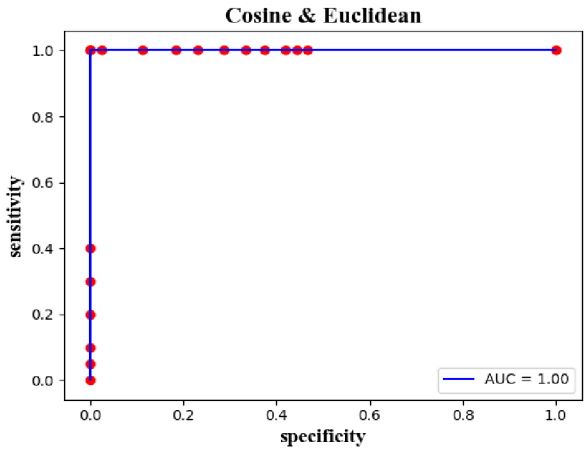
Při resubstituce porovnáváme každou kombinace výsledků s vektorem uměle vygenerovaných anomálií. Přitom pro každou kombinace spočítáme přesnost klasifikace a zjistíme nejvyšší hodnotu přesnosti, která bude ukazovat na nejlepší kombinace klasifikace. Vypočet přesnosti je uveden v tabulce č. 5.3.

Tab. 5.3: Hodnocení kvality algoritmu detekce anomálie při resubstituce.

Výskyt outlierů	Percentil	Kombinace metrik	Přesnost (Accuracy)
1%	99%	Kosinová a Euklidovská	99.8%
5%	95%	Kosinova	100%

10%	90%	Kosinová a Euklidovská	99.75%
15%	86%	Euklidovská	99.6%

Z tabulky 5.3 vyplývá, že kombinace Kosinové a Euklidovské metriky dávají lepší výsledky s vysokou přesnosti 99% - 100%. Takže odsud plyne, že existuje závislost parametru percentilu na procentu zaražení outlierů v datové sadě. Na obrázku č. 5.13 je vidět ideální ROC křivku, což je grafickým důkazem přesnosti klasifikace.



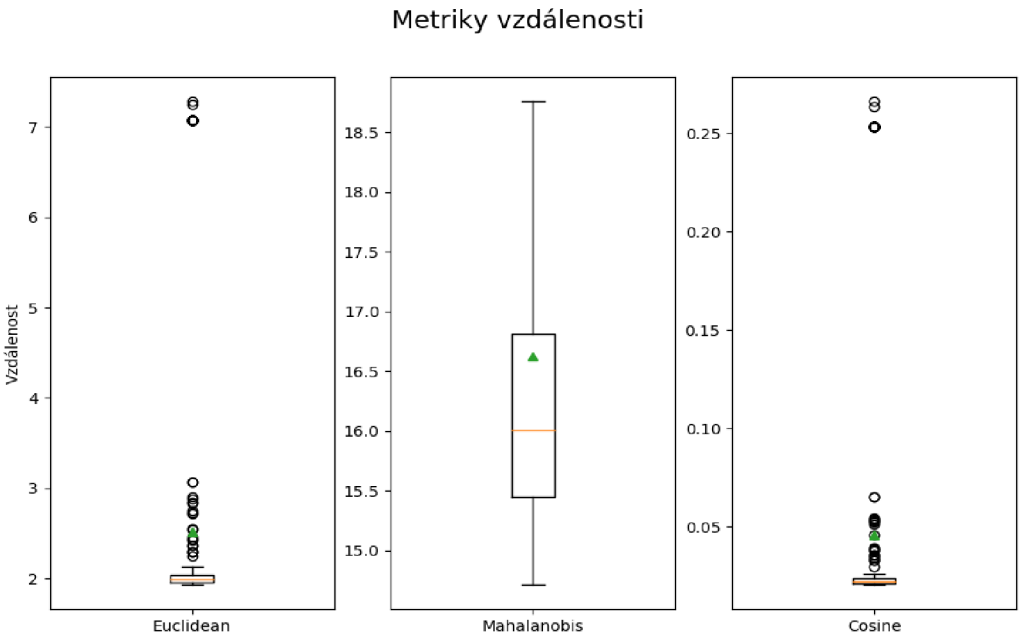
Obr. 5.13: ROC křivka kombinace Kosinové a Euklidovské metriky při 10% výskytu outlierů (resubstituce).

Parametr AUC (Area Under Curve – AUC), který je videt na obrazku č. 5.13, je plochou pod křivkou. Následující tabulka ukazuje ohodnocení přesnosti testu:

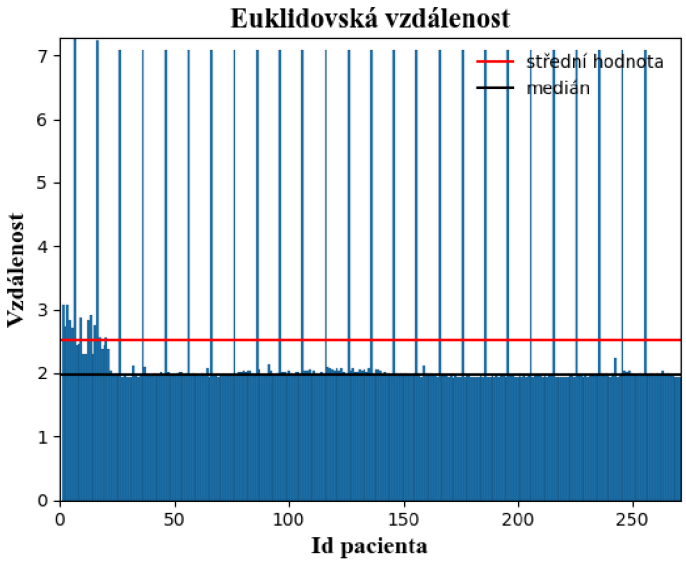
Tab. 5.4: Hodnocení kvality klasifikace podle plochy pod křivkou.

velikost AUC	**hodnocení**
0,9 - 1.0	výborně
0,8 - 0,9	velmi dobře
0,7 - 0,8	dobře
0,6 - 0,7	dostatečně
0,5 - 0,6	nedostatečně

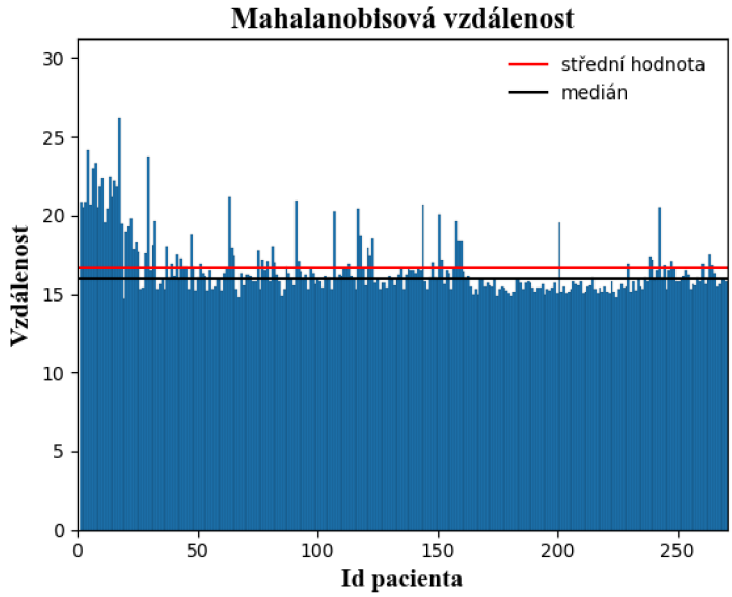
Testování u modelu externí validací provádíme na nové datové sadě. Testování "hold out" modelu zahrnuje vypočet nejlepší vybrané při trénování, kombinace a porovnání výsledku této kombinace s vektorem uměle vygenerovaných anomálií. Na konci je třeba provést hodnocení kvality vytvořeného algoritmu.



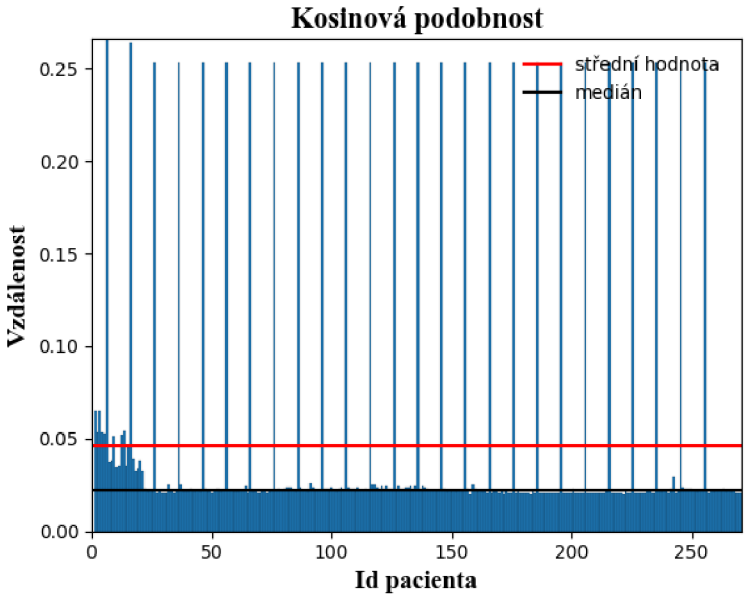
Obr. 5.14: Metriky vzdálenosti při 10% znečištění testovací datové sady.



Obr. 5.15: Euklidovská vzdálenost při 10% výskytu outlierů (testovací datová sada).



Obr. 5.16: Mahalanobisová vzdálenost při 10% výskytu outlierů (testovací datová sada).

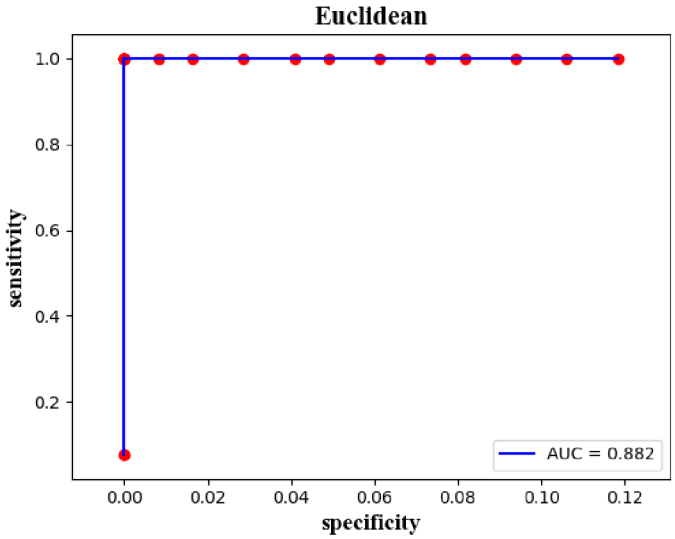


Obr. 5.17: Kosinová podobnost při 10% výskytu outlierů (testovací datová sada).

Výsledky posledních třech obrázku korespondují s dosaženými výsledky z předchozí analýzy při resubstituce. V tabulce č. 5.4 je uvedena nejlepší kombinace a přesnost klasifikace.

Tab. 5.5: Hodnocení kvality algoritmu detekce anomálie při externí validaci.

Výskyt outlierů	Percentil	Kombinace metrik	Přesnost (Accuracy)
1%	99%	Kosinová a Euklidovská	99%
5%	95%	Euklidovská či Mahalanobisová spárované s Kosinovou	99%
10%	90%	Kosinová a Euklidovská	100%
15%	85%	Kosinová a Euklidovská	99.8%



Obr. 5.18: ROC křivka kombinace Kosinové a Euklidovské metriky při 10% výskytu outlierů (externí validace)

Podle uvedených grafů a výsledků klasifikace dá se říct, že realizovaný algoritmus detekce anomálie má vysokou přesnost i kvalitu a zároveň s tím nezávisí na rozměru vstupní datové sady. Dany algoritmus byl naučen detekovat anomálie v datech pomoci dvou přístupu zdrojového učení: resubstituce a externí validace. Výsledkem učení bylo stanoveno, že nejlepší odhad dává kombinace Kosinové a Euklidovské metriky s ohledem na vhodný práh percentilu. Při jinem poctu anomálních záznamů v datové sadě přesnost se nemění, kvalita klasifikace zůstává stejná jako při 10% znečistění. Takže tento algoritmus byl použít na dalších dvou registrech, kde algoritmus zjistil, že tyto registry obsahují anomální záznamy. Výsledky této analýzy byly odeslány na kontrolu správce dat. V příloze je uveden python skript který obsahuje realizace popsaného výše algoritmu.

6 ZÁVĚR

Detekce anomálie je aktuálním trendem v oblasti kvality dat. Jak už bylo zmíněno v teoretické části této práce, poctivost a pravdivost jsou základními principy vědeckého výzkumu. Dodržování těchto zásad je nezbytné jak pro rozvoj vědy, tak pro veřejné vnímání vědeckých výsledků. Odchylky od těchto zásad mohou být považovány za vědecké pochybení nebo podvod. V oblasti klinického výzkumu může nedodržování těchto zásad vést k ohrožení lidského života.

V diplomové práci jsem na základě literární rešerše popsal typy KS a základní zdroje nekvalitních dat. Cílem této práce bylo realizovat algoritmus detekce anomálie pro rozšíření palety funkce u vybraného informačního systému (CLADE-IS: Clinical Data Warehousing Information System), který umožňuje provedení analýzy, sběr, zálohování a archivace dat z reálné klinické praxe. Proto byla provedena teoretická analýza existujících metod detekce odlehlých hodnot. Často používané vizuální metody detekce anomálií v datech nejsou vždy efektivní. Kromě vizuálních metod existují další metody zjištění anomálií ve vícerozměrných datech. Bohužel většina těchto metod potřebuje vědět přesný počet anomálie či procento výskytu anomálie v datech, což v reálné praxe ne vždy je možné zjistit. Proto byl realizován vlastní algoritmus detekce anomálie na základě známých statistických metod.

Poslední kapitola práce je věnována ukázce navrženého algoritmu detekce anomálií na reálných, anonymizovaných datech získaných z národních a nadnárodních neintervenčních klinických studií. Realizovaný algoritmus využívá základní koncepce zdrojového učení: trénování algoritmu na původním souboru dat a testování či využití získaného modelu na nové datové sadě. Pro trénování navřeného algoritmu byly použity dvě metody: resubstituce (resubstitution) a externí validace (hold-out). Pro správnou klasifikace vstupní datový soubor byl umělé modifikován, přidáním anomálních záznamů. Výsledek detekce byl porovnán s vektorem umele vygenerovaných anomálie. Přesnost klasifikace při resubstituce je v rámci 99% a nejlepší výsledek detekce generuji kombinace Euklidovské vzdálenosti a Kosinové podobnosti. Kromě toho bylo zjištěno, že parametr percentilu je závislý na procentu výskytu outlierů v datové sadě, což je uvedeno v tabulce č. 5.3. Proto při neznámém počtu anomálie v datech je možné projít manuálně celý soubor detekovaných anomálií při různém prahu percentilu. Testování modelu při externí validací dává skoro stejné výsledky: přesnost 99%, nejlepší kombinace je Euklidovská vzdálenost a Kosinova podobnost (tabulka č. 5.4).

Algoritmus, který je součástí diplomové práce, splňuje cíl vyhledání anomálních záznamů, čímž umožní zlepšení kvality klinických registrů a tím zvýší jejich výpovědní hodnotu.

LITERATURA

[1] SVOBODNÍK, A., DEMLOVÁ, R.., PECEN, L., HANÁKOVÁ, M., KADLECOVÁ, P., KOČA, J., KOSTKOVÁ, H., MACHULKA, T., SOUČKOVÁ, I., SOUČKOVÁ, L., ŠIMÍČEK, M., ŠTĚPÁNOVÁ, R., 2014: *Klinické studie v praxi*. Brno: Facta Medica, 229 p. ISBN 97880-904731-8-8.

[2] SPILKER, B, *Guide to Clinical Trials*. [online]. [cit. 2017-15-10]. Dostupné z: http://www.virginia.edu/vpr/irb/HSR_docs/CLINICAL_TRIALS_Phases.pdf.

[3] FRIEDMAN, L. M., FURBERG, C., DEMETS, D. L. *Fundamentals of clinical trials. 4th ed.* New York: Springer, 2010. ISBN 978-1-4419-1586-3.

[4] *Different types of clinical trials*. [online]. [cit. 2017-21-10]. Dostupné z: http://www.scientific-european-federation-osteopaths.org/different-types-of-clinical-trials/.

[5] VALENCIA, E. *What is Real World Evidence and why does it matter?* [cit. 2017-21-10]. Dostupné z: https://www.meaningcloud.com/blog/real-world-evidence

[6] *Typy studií* [online]. [cit. 2017-28-12]. Dostupné z: https://mefanet.upol.cz/download.php?fid=115

[7] BABRE D. *Electronic data capture – Narrowing the gap between clinical and data management.* Perspectives in Clinical Research. 2011; doi:10.4103/2229-3485.76282.

[8] *CLADE-IS (Clinical data warehouse – information system)* [online]. [cit. 2017-04-12]. Dostupné z: http://www.biostatistika.cz/index.php?pg=produkty--clade-is

[9] *Real World Evidence* [online]. [cit. 2017-22-12]. Dostupné z: http://www.apteka.ua/article/403741

[10] STEPHEN, L. G., BUYSE, M. *Data fraud in clinical trials.* Clinical Investigation. 2015, Str. 161-173. ISSN 2041-6792.

[11] KNEPPER, D. *Statistical monitoring in clinical trials: best practices for detecting data anomalies suggestive of fabrication or misconduct.* Therapeutic Innovation & Regulatory Science, SAGE Publishing. 2016, doi: 10.1177/2168479016630576.

[12] SMITH, T. C., STOCKEN, D. D., DUNN, J., COX, T., GHANEH, P., CUNNINGHAM, D., NEOPTOLEMOS, J. P. *The Value of Source Data Verification in a Cancer Clinical Trial*. PLoS ONE. 2012. ISSN 1932-6203.

[13] EDWARDS, P., SHAKUR, H., BARNETSON, L., PRIETO, D., EVANS, S. *Central and statistical data monitoring in the Clinical Randomisation of an Antifibrinolytic in Significant Haemorrhage (CRASH-2) trial*. Clinical Trials. 2014, ISSN 1740-7745.

[14] GROSSMAN, R., SENI, G., ELDER, J., AGARWALl, N., LIU, H. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions.* Morgan & Claypool. 2010. doi:10.2200/S00240ED1V01Y200912DMK002.

[15] GHASEMI, A., SALECH, Z. *Normality Tests for Statistical Analysis: A Guide for*

Non-Statisticians. International Journal of Endocrinology and Metabolism. 2012.

[16] CORDER, G. W., FOREMAN, D. I. *Nonparametric Statistics: A Step-by-Step Approach*. Wiley. 2014. ISBN 978-1118840313.

[17] ROUSSEEUW, P. J., CROUX, C. *Alternatives to the median absolute deviation.* Journal of the American Statistical Association. 1993. doi:10.1080/01621459.1993.10476408.

[18] BANERJEE, S., ROY, A. *Linear Algebra and Matrix Analysis for Statistics, Texts in Statistical Science (1st ed.)*, Chapman and Hall/CRC, 2014. ISBN 978-1420095388.

[19] GE, MAOCHEN. *Source location error analysis and optimization methods*. Journal of Rock Mechanics and Geotechnical Engineering, 2012.

[20] ZIMEK, A., CAMPELLO, R., SANDER, J. R. *Ensembles for unsupervised outlier detection*. ACM SIGKDD Explorations Newsletter. 2015. doi:10.1145/2594473.2594476.

[21] CHROSTOPH, H. L. *Kernel Methods in Computer Vision*, Foundations and Trends in Computer Graphics and Vision. 2009.

[22] *OneClassSVM.* [online]. [cit. 2018-15-02]. Dostupné z: http://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html.

[23] FEI, T. L., KAI, M. T., ZHI-HUA, Z. *Isolation Forest*. Data Mining, 2008. ICDM '08. Eighth IEEE International Conference. ISSN: 2374-8486.

[24] KOHAVI, R., *A study of cross-validation and bootstrap for accuracy estimation and model selection*. [online]. [cit. 2018-11-04]. 1995. Dostupné z: https://sebastianraschka.com/blog/2016/model-evaluation-selection-part1.html.

[25] CHRISTENSEN, R. *Thoughts on prediction and cross-validation*. [online]. [cit. 2018-23-04]. Department of Mathematics and Statistics University of New Mexico. 2015. Dostupné z: http://www.math.unm.edu/~fletcher/Prediction.pdf.

[26] OHSAKI, M. *Confusion-matrix-based kernel logistic regression for imbalanced data classification*. IEEE Trans. Knowl. Data Eng. 2017. doi:10.1109/TKDE.2017.2682249

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

- KS - klinické studie
- EDC - systém elektronického sběru dat
- eCRF - elektronický záznam subjektů studie
- RWE - evidence skutečného světa
- RWD - Real World Data (data z reálné klinické praxe)
- RWE - Real World Evidence (evidence z reálné klinické praxe a)
- EAV - Entity -Attribute -Value model
- FDA - Food and Drug Administration (Úřad pro kontrolu potravin a léčiv)
- EMA - European Medicines Agency (Evropská agentura pro léčivé přípravky)
- MAD - median absolute deviation (mediánová absolutní odchylka)
- CLADE-IS informační systém pro skladování klinických dat
- NVAF - Nonvalvular atrial fibrillation (nechlopňové fibrilace síní)
- SVD - singular value decomposition (rozklad na singulární hodnoty)
- SVM - support vector machines
- TP – True Positive (skutečně pozitivní výsledek)
- FP – False Positive (falešně pozitivní výsledek)
- FN – False Negative (falešně negativní výsledek)
- TN – True Negative (skutečně negativní výsledek)
- ROC – Receiver Operating Characteristic

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

PŘÍLOHY

Realizovaný algoritmus je uložen v https://github.com/maxbond007/Outlier_Detection.git

1. **Využité balíčky:**
   - pandas
   - numpy
   - psycopg2
   - datetime
   - scipy
   - sklearn.metrics
   - matplotlib.pyplot

2. **Funkce SQL importu, filtrace a transformace dat:**

```
def date_import(cursor):
    """
       This function imports "subject_id" "form_id",
       "question_id" and "data_type" from the database

    """
    cursor.execute('''
            select Q1."Patient's ID" from
               (select s.id as "Patient's ID"
                   from cls_form_study_phase f join
                   cls_subject s on s.id = f.subject_id
                   where f.form_structure_id = 1
                   and f.soft_delete = false
                   and s.soft_delete = false
                   and s.test_subject = false
                   order by "Patient's ID") as Q1
                ''')
    Data = cursor.fetchall()
    Data = pd.DataFrame(Data, columns=["Patient_id"])
```

```
cursor.execute('''
        select f.form_structure_id as form_id,
            q.question_id as question_id,
            qq.question_datatype_id as data_type
            from cls_question_group_form_structure f
            inner join cls_question_group_question q on
            f.question_group_id = q.question_group_id
            inner join cls_question qq on q.question_id = qq.id
            order by f.form_structure_id ASC, q.question_id ASC
            ''')

Parameters_id = cursor.fetchall()
Parameters_id = pd.DataFrame(Parameters_id,
                    columns=["Form_id", "Question_id", "Data_type"])
Parameters_id.Form_id[Parameters_id.Form_id == 3] = 2
"""
   This part of the function imports questions from the database, filters and
   transforms data and creates dataframe of questions.

"""
for i,item in enumerate(Parameters_id.Form_id):
   try: # import questions
      cursor.execute('''
         select "Q1" from
         (select s.secondary_id as "Patient's ID",
         cls_get_question_value_for_validation(f.form_data-> %s) as "Q1"
         from cls_form_study_phase f join
         cls_subject s on s.id = f.subject_id
         where f.form_structure_id = %s
         and f.soft_delete = false
         and s.soft_delete = false
         and s.test_subject = false
         order by "Patient's ID") as Q1
```

```
        '',('Q' + str(Parameters_id.Question_id[i]),
        int(Parameters_id.Form_id[i])))
except psycopg2.InternalError:
    print("Caught error:   iter:",i,
          ' Form_id:',Parameters_id.Form_id[i],
          ' Question_id:', Parameters_id.Question_id[i])
    continue
# filtering and transformation of data
list_of_questions = cursor.fetchall()
if (list_of_questions == [] or len(set(list_of_questions)) == 1 or
    count_element(list_of_questions,None)>len(list_of_questions)*0.8 or
    count_element(list_of_questions,np.nan)>len(list_of_questions)*0.8 or
    Parameters_id.Data_type[i] == "string" or
    Parameters_id.Data_type[i] == "heading" or
    Parameters_id.Data_type[i] == "text"):
    continue

if Parameters_id.Data_type[i] == "boolean":
    list_of_questions = pd.DataFrame(list_of_questions)
    list_of_questions[list_of_questions == 'true'] = 1
    list_of_questions[list_of_questions == 'false'] = 0

if Parameters_id.Data_type[i] == "date":
    list_of_questions = pd.DataFrame(list_of_questions)
    list_of_questions = conver_to_date(list_of_questions)

if (Parameters_id.Data_type[i] == "real" or
    Parameters_id.Data_type[i] == "int" or
    Parameters_id.Data_type[i] == "discrete_value"):
    list_of_questions = [float(np.nan if i[0] is None else i[0])
        for i in list_of_questions]
    list_of_questions = pd.DataFrame(list_of_questions)

if list_of_questions.isnull().sum()[0] >= len(list_of_questions)*0.8:
```

```
                continue

            # add list of questions to the dataframe
            Data['Q' + str(Parameters_id.Question_id[i])] = list_of_questions
        cursor.close()
        return Data
```

3. Algoritmus detekce anomálie:

```
def calculate_dist(data):
    """
        This function replaces NaN values for median and scales values to [0 1]
        calculating three vectors of Euclidean,Mahalanobis and Cosine distances.

    """
    import scipy.spatial.distance as dist
    import numpy as np
    import scipy as sp
    from ploting import boxplot, histogram

    # scaling
    del data['Patient_id']
    for k in  data.columns:
        data[k] = data[k].fillna(round(data[k].median()))
        data[k] = (data[k] - data[k].min())
        data[k] = (data[k] - data[k].min())/data[k].max()
        if data[k].isnull().sum() >  len(data[k])*0.8 or len(set(data[k])) == 1:
            del data[k]

    centroid =  np.mean(data)
    centroid = centroid.as_matrix()
    numpyMatrix = data.as_matrix()

    #Calculate covariance matrix
    covmx = data.cov()
```

```
    invcovmx = sp.linalg.pinv(covmx)

    #Calculate Euclidean,Mahalanobis and Cosine distance
    Mahaldist, Eucliddist, Cosinedist  = [],[],[]
    for h in range(len(numpyMatrix)):
        Mahaldist.append(dist.mahalanobis(numpyMatrix[h], centroid, invcovmx))
        Eucliddist.append(dist.euclidean(numpyMatrix[h], centroid))
        Cosinedist.append(dist.cosine(numpyMatrix[h], centroid))

    # ploting
    boxplot(Eucliddist,Mahaldist,Cosinedist)
    histogram(Eucliddist,Mahaldist,Cosinedist, len(data))

    return Mahaldist, Eucliddist, Cosinedist

#%%

def detector(Mahaldist, Eucliddist, Cosinedist):
    '''
        Outlier Detection by percentile
    '''
    import pandas as pd
    import numpy as np
    size = len(Eucliddist)

    outliers_euc, outliers_mah, outliers_cos = [],[],[]
    outliers_1, outliers_2, outliers_3, outliers_4 = [],[],[],[]

    euclid = pd.DataFrame([[0]*20 for i in range(size)])
    mahal = pd.DataFrame([[0]*20 for i in range(size)])
    cosin = pd.DataFrame([[0]*20 for i in range(size)])
    macos = pd.DataFrame([[0]*20 for i in range(size)])
    maeuc = pd.DataFrame([[0]*20 for i in range(size)])
    eucos = pd.DataFrame([[0]*20 for i in range(size)])
```

```
maeuccos = pd.DataFrame([[0]*20 for i in range(size)])


for k in range(20):
    P1=np.percentile(Euclliddist, 80+k)
    P2=np.percentile(Mahaldist, 80+k)
    P3=np.percentile(Cosinedist, 80+k)

    my_list1 = pd.DataFrame([[x,index] for index, x in enumerate(Euclliddist)
        if x >= P1], columns=['parametr','id'])
    my_list2 = pd.DataFrame([[x,index] for index, x in enumerate(Mahaldist)
        if x >= P2], columns=['parametr','id'])
    my_list3 = pd.DataFrame([[x,index] for index, x in enumerate(Cosinedist)
        if x >= P3], columns=['parametr','id'])

    # search for similar anomalies
    outliers1 = pd.DataFrame(list(set(my_list2.id) & set(my_list3.id)))
    outliers2 = pd.DataFrame(list(set(my_list1.id) & set(my_list2.id)))
    outliers3 = pd.DataFrame(list(set(my_list1.id) & set(my_list3.id)))
   outliers4=pd.DataFrame(list(set(my_list1.id)&set(my_list2.id)&set(my_list3.id)))

    outliers_4.append(list(set(my_list1.id) & set(my_list2.id) & set(my_list3.id)))
    outliers_3.append(list(set(my_list1.id) & set(my_list3.id)))
    outliers_2.append(list(set(my_list1.id) & set(my_list2.id)))
    outliers_1.append(list(set(my_list2.id) & set(my_list3.id)))

    outliers_euc.append(list(set(my_list1.id)))
    outliers_mah.append(list(set(my_list2.id)))
    outliers_cos.append(list(set(my_list3.id)))

    if not outliers1.empty:
        macos[k][outliers1[0]] = 1
    if not outliers2.empty:
        maeuc[k][outliers2[0]] = 1
```

```
        if not outliers3.empty:
            eucos[k][outliers3[0]] = 1
        if not outliers4.empty:
            maeuccos[k][outliers4[0]] = 1

        euclid[k][my_list1.id] = 1
        mahal[k][my_list2.id] = 1
        cosin[k][my_list3.id] = 1

    outliers = {}
    #dictionary of dataframe of anomalies
    outliers['euclid'] = pd.DataFrame(outliers_euc).T
    outliers['mahal'] = pd.DataFrame(outliers_mah).T
    outliers['cosine'] = pd.DataFrame(outliers_cos).T
    outliers['mahal+cosine'] = pd.DataFrame(outliers_1).T
    outliers['mahal+euclid'] = pd.DataFrame(outliers_2).T
    outliers['euclid+cosine'] = pd.DataFrame(outliers_3).T
    outliers['mah+euc+cos'] = pd.DataFrame(outliers_4).T
    return euclid, mahal, cosin, macos, maeuc, eucos, maeuccos, outliers



#%%
def quality_of_classification(euclid, mahal, cosin, macos, maeuc, eucos,
                              maeuccos, true_data)
    '''
        This function calculates accuracy, specificity and sensitivity
        in different combinations of distance.

    '''
    import pandas as pd
    from sklearn.metrics import confusion_matrix, accuracy_score
    #true_data = np.zeros(len(euclid))
    accuracy_euclid, accuracy_mahal, accuracy_cosin = [],[],[]
    specificity_euclid, specificity_mahal, specificity_cosin = [],[],[]
```

```
sensitivity_euclid, sensitivity_mahal, sensitivity_cosin = [],[],[]
accuracy_macos, specificity_macos, sensitivity_macos = [],[],[]
accuracy_maeuc, specificity_maeuc, sensitivity_maeuc = [],[],[]
accuracy_eucos, specificity_eucos, sensitivity_eucos = [],[],[]
accuracy_maeuccos, specificity_maeuccos, sensitivity_maeuccos = [],[],[]
accuracy = []
max_acc = []

frame_dict = {k: pd.DataFrame(columns=['sensitivity','specificity','accuracy'])
   for k in range(20)}

for i in range(20):

   tn,fp,fn,tp = confusion_matrix(true_data, euclid[i]).ravel()
   accuracy_euclid.append(accuracy_score(true_data, euclid[i]))
   specificity_euclid.append(fp/(fp+tp))
   sensitivity_euclid.append(tp/(tp+fn))

   tn,fp,fn,tp = confusion_matrix(true_data, mahal[i]).ravel()
   accuracy_mahal.append(accuracy_score(true_data, mahal[i]))
   specificity_mahal.append(fp/(fp+tp))
   sensitivity_mahal.append(tp/(tp+fn))

   tn,fp,fn,tp = confusion_matrix(true_data, cosin[i]).ravel()
   accuracy_cosin.append(accuracy_score(true_data, cosin[i]))
   specificity_cosin.append(fp/(fp+tp))
   sensitivity_cosin.append(tp/(tp+fn))

   tn,fp,fn,tp = confusion_matrix(true_data, macos[i]).ravel()
   accuracy_macos.append(accuracy_score(true_data, macos[i]))
   specificity_macos.append(fp/(fp+tp))
   sensitivity_macos.append(tp/(tp+fn))

   tn,fp,fn,tp = confusion_matrix(true_data, eucos[i]).ravel()
```

```
        accuracy_eucos.append(accuracy_score(true_data, eucos[i]))
        specificity_eucos.append(fp/(fp+tp))
        sensitivity_eucos.append(tp/(tp+fn))

        tn,fp,fn,tp = confusion_matrix(true_data, maeuc[i]).ravel()
        accuracy_maeuc.append(accuracy_score(true_data, maeuc[i]))
        specificity_maeuc.append(fp/(fp+tp))
        sensitivity_maeuc.append(tp/(tp+fn))

        tn,fp,fn,tp = confusion_matrix(true_data, maeuccos[i]).ravel()
        accuracy_maeuccos.append(accuracy_score(true_data, maeuccos[i]))
        specificity_maeuccos.append(fp/(fp+tp))
        sensitivity_maeuccos.append(tp/(tp+fn))



        frame_dict[i].loc['euclid'] = [sensitivity_euclid[i],
                specificity_euclid[i],accuracy_euclid[i]]
        frame_dict[i].loc['mahal'] = [sensitivity_mahal[i],
                specificity_mahal[i],accuracy_mahal[i]]
        frame_dict[i].loc['cosin'] = [sensitivity_cosin[i],
                specificity_cosin[i],accuracy_cosin[i]]
        frame_dict[i].loc['mahal+cosine'] = [sensitivity_macos[i],
                specificity_macos[i],accuracy_macos[i]]
        frame_dict[i].loc['euclid+cosine'] = [sensitivity_eucos[i],
                specificity_eucos[i],accuracy_eucos[i]]
        frame_dict[i].loc['mahal+euclid'] = [sensitivity_maeuc[i],
                specificity_maeuc[i],accuracy_maeuc[i]]
        frame_dict[i].loc['mah+euc+cos'] = [sensitivity_maeuccos[i],
                specificity_maeuccos[i],accuracy_maeuccos[i]]

        max_acc.append(frame_dict[i]['accuracy'].max())
        accuracy.append(frame_dict[i]['accuracy'][frame_dict[i]['accuracy']                    ==
max_acc[i]].to_dict())
```

```
    #max_acc.index(max(max_acc))
return max_acc, accuracy, frame_dict
```

4. **Algoritmus generace anomálních záznamu v datové sadě:**

```
def generator(test, percent):
    import numpy as np
    # генерация аномальных пациентов 1%
    if percent == 1:
        true_data = np.zeros(len(test))
        anom_patients = [100,303,404,256]
        for i in anom_patients:
            test.iloc[int(i),1:50] = 5
            test.iloc[int(i),300:370] = 5
            true_data[i] = 1

    # 5% anomálních záznamů
    elif percent == 5:
        true_data = np.zeros(len(test))
        anom_patients = [0,1,2,3,4,5,6,7,9,11,28,40,53,89,201,203]
        for i in anom_patients:
            test.iloc[i,0:50] = 5
            test.iloc[i,300:370] = 5
            true_data[i] = 1

    # 10% anomálních záznamů
    elif percent == 10:
        true_data = np.zeros(len(test))
        anom_patients = [i for i in np.arange(0,400, 10)]
        for i in anom_patients:
            test.iloc[int(i),1:100] = 5
            test.iloc[int(i),300:370] = 5
            true_data[int(i)] = 1

    # 15% anomálních záznamů
```

```
    elif percent == 15:
        true_data = np.zeros(len(test))
        anom_patients = [i for i in np.arange(0,400, 7)]
        for i in anom_patients:
            test.iloc[i,1:100] = 5
            true_data[i] = 1
    else:
        print('Wrong number of percent, please choose one from these: 1,5,10,15')
    return test, true_data, anom_patients
```

5. **Funkci kresleni obrázků (boxplot a histogram):**

```
    def boxplot(Euclideandist,Mahalanobisdist,Cosinedist):
        '''
            Ploting boxplot

        '''
        import matplotlib.pyplot as plt

        fig = plt.figure()
        fig.suptitle('Metriky vzdálenosti', fontsize=16)
        ax = fig.add_subplot(131)
        ax.boxplot(Euclideandist, labels=['Euclidean'], showmeans=True,
              showfliers=True)
        ax.set_ylabel('Vzdálenost', fontsize=10)
        ax = fig.add_subplot(132)
        ax.boxplot(Mahalanobisdist, labels = ['Mahalanobis'], showmeans=True,
              showfliers=True)
        ax = fig.add_subplot(133)
        ax.boxplot(Cosinedist, labels = ['Cosine'], showmeans=True,
              showfliers=True)
        fig.show()
        return 0
```

```
def histogram(Euclideandist,Mahalanobisdist, Cosinedist, length):

    import numpy as np
    import matplotlib.pyplot as plt

    # hist of Euclidean
    x = np.arange(1,length+1)
    plt.figure()
    plt.hist(x, weights = Euclideandist, bins=length,
             histtype='bar', ec='k', linewidth=0.1)
    plt.title ("Euklidovská vzdálenost",  {'fontname':'Times New Roman'},
               fontsize=16)
    plt.xlim(0, length)
    plt.ylim(0, max(Euclideandist))
    plt.ylabel('Vzdálenost',  {'fontname':'Times New Roman'}, fontsize=14)
    plt.xlabel('Id pacienta',  {'fontname':'Times New Roman'}, fontsize=14)
    plt.axhline(y= np.mean(Euclideandist), color='r', linestyle='-',
                label = 'střední hodnota')
    plt.axhline(y= np.median(Euclideandist), color='k', linestyle='-',
                label = 'medián')
    plt.legend(loc='left right', frameon = False)
    plt.show()

    # hist of Mahalanobis
    x = np.arange(1,length+1)
    plt.figure()
    plt.hist(x, weights = Mahalanobisdist, bins=length,
             histtype='bar', ec='k', linewidth=0.1)
    plt.title ("Mahalanobisová vzdálenost", {'fontname':'Times New Roman'},
               fontsize=16)
    plt.xlim(0, length)
    plt.ylim(0, max(Mahalanobisdist)+5)
    plt.ylabel('Vzdálenost', {'fontname':'Times New Roman'}, fontsize=14)
    plt.xlabel('Id pacienta', {'fontname':'Times New Roman'}, fontsize=14)
```

```
plt.axhline(y= np.mean(Mahalanobisdist), color='r', linestyle='-',
          label = 'střední hodnota ')
plt.axhline(y= np.median(Mahalanobisdist), color='k', linestyle='-',
          label = 'medián')
plt.legend(loc='left right', frameon = False)
plt.show()

# hist of Cosine
x = np.arange(1,length+1)
plt.figure()
plt.hist(x, weights = Cosinedist, bins=length,
        histtype='bar', ec='k', linewidth=0.1)
plt.title ("Kosinová podobnost", {'fontname':'Times New Roman'},
          fontsize=16)
plt.xlim(0, length)
plt.ylim(0, max(Cosinedist))
plt.ylabel('Vzdálenost', {'fontname':'Times New Roman'}, fontsize=14)
plt.xlabel('Id pacienta', {'fontname':'Times New Roman'}, fontsize=14)
plt.axhline(y= np.mean(Cosinedist), color='r', linestyle='-',
          label = 'střední hodnota ')
plt.axhline(y= np.median(Cosinedist), color='k', linestyle='-',
          label = 'medián')
plt.legend(loc='left right', frameon = False)
plt.show()

return 0
```