TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Artjom Valdas

Juhuslike diagnooside trajektooride generaator

Bakalaureusetöö (9 EAP)

Juhendaja: Jaak Vilo

2021

Juhuslike diagnooside trajektooride generaator

Lühikokkuvõte:

Andmeteaduse üks esimesi samme on andmete kogumine. Mõnedel juhtudel on andmed privaatsed ja ei ole niisama kättesaadavad, eriti need, mis puudutavad isikuandmeid. Käesoleva lõputöö raames luuakse programm, mis genereerib reaalsetele andmetele lähedased patsiendiandmed koos aja ja diagnoosidega, mis võivad tulevasi haigusi süvendada või vastupidi vähendada. Selliseid andmeid saab kasutada tehisnärvivõrkude treenimiseks ning tulevaste haiguste ennustamiseks. Kuna andmed on täiesti juhuslikud ja genereeritud lihtsa mudeli baasil, ei sisalda need privaatsusriski mitte kellelegi ja teiseks on täpselt teada, millise mudeli baasil andmed on tuletatud. Seega on nende andmete analüüsis võimalik valideerida analüüsimeetodi kasulikkust konkreetse mudeli seisukohast.

Võtmesõnad:

Bayesi võrgud, andmete genereerimine, bioinformaatika

CERCS:

P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

B110 Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika

Random diagnoses trajectory generator

Abstract:

One of the first steps in data science is data collection. Sometimes the data is private and cannot be accessed in any way, especially those concerning personal data. This bachelor's thesis is about creating a program that generates data about patients close to real ones, as well as times and diagnoses that can exacerbate or, conversely, reduce future diseases. Such data can be used to train artificial neural networks and predict future diseases. Because the data is completely random and generated based on a simple model, it does not pose a privacy risk to anyone, and secondly, it is known exactly which model the data is derived from. Thus, in the analysis of these data, it is possible to validate the usefulness of the analytical method for a particular model.

Keywords:

Bayesian network, data generation, bioinformatics

CERCS:

P170 Computer science, numerical analysis, systems, control

B110 Bioinformatics, medical informatics, biomathematics, biometrics

Sisukord

Sissejuhatus

Viimaste aastate jooksul on näha olnud suurt huvi masinõppe ja tehisintellekti meetodite rakendamise, näiteks tehisnärvivõrkude vastu. Selliseid lahendusi rakendatakse erinevates valdkondades, alustades tervishoiust [1] ja lõpetades põllumajandusega [2]. Suur huvi närvivõrkude vastu on seotud sellega, et need on võimelised iseseisvalt lahendama eri tüüpi keerukusega ülesandeid, alustades tuleviku prognoosimisest ja lõpetades andmete klassifitseerimisega. Andmete kogumist ning korrektsete andmete valikut tehisnärvivõrgu õpetamiseks võib pidada üheks raskemaks etapiks andmeteaduses.

Üheks valdkonnaks, kus on andmetest puudus johtuvalt nende sensitiivsusest, on meditsiin. Andmeid on nii palju kui patsiente, kuid probleemiks on nende andmete tundlikkus [3], mis tähendab, et terviseandmed peavad olema kaitstud volitamata juurdepääsu eest. Terviseandmed viitavad isikuandmetele, mis on seotud nii tervisliku seisundiga (laboratoorsed uuringud, arstide saatekirjad jms) kui ka tervishoiualaste finantsteavetega (tervishoiuteenuste arved). Antud informatsioon hõlmab delikaatseid andmeid ja seetõttu kehtivad nende suhtes eriti ranged reeglid. Üldjuhul neid saavad töödelda ainult tervishoiutöötajad patsiendile ravi osutamise eesmärgil, seejuures seob neid täiendavalt meditsiini saladuse hoidmise kohustus [4]. Käesoleva bakalaureusetöö eesmärgiks on luua võimalikult lihtne andmete generaator, mille baasil genereerida päris andmetele sarnaseid, kuid samas lihtsamaid ja juba tuntud mudelile vastavaid meditsiiniandmeid.

Tänapäeval leidub antud probleemile sarnane tehniline lahendus, milleks on Synthea [5]. See avatud lähtekoodiga programm kasutab patsientide genereerimiseks Massachusettsi osariigi elanike andmebaasi. Patsiente luuakse ükshaaval, lastes läbi kõik haigusmoodulid, alustades vaktsineerimisest ja lõpetades sotsiaalsete teguritega. Kõik üleminekud ehk elusündmused omavad kindlaid tekkimistõenäosusi. Synthea nõrkuseks on saadud andmestikus tõenäosuste ja sõltuvuste muutmise võimalus. Kasutajal puudub võimalus lisada kõrvalekaldeid ning trajektoore.

Antud töö käigus võeti uue, võimalikult lihtsa meetodi loomise lähtekohaks Markovi ahelapõhine lähenemine. Programmi sisendiks on mudel, kuhu on sisestatud erinevate diagnooside kategooriate, haiguste sündmuste ja nende järgnevuste (trajektooride) tõenäosused sõltuvalt vanusest ja soost ning mis annab väljundiks genereeritud isikuid koos sünnipäeva, surmapäeva ja diagnooside massiiviga. Genereeritud andmete baasil luuakse fail (andmestik), mida saab omakorda importida OHDSI OMOP ühise andmemudeli kujul olevasse [6]

andmebaasi. OMOP andmemudeli järele on vajadus, sest andmeanalüüsi programmid luuakse ennekõike selle mudeli järgi päris patsientide andmete analüüsiks. Seega saab genereeritud andmete alusel testida ja arendada edasi uusi andmeanalüüsi lahendusi. Raskus seisneb selles, et diagnoosid peaksid olema ajalises järgnevuses ja sõltuvuses üksteisest, et tekiksid "haiguste trajektoorid", seega on sünteesitud andmete juures võimalik lisada ka sündmuse toimumise kuupäev.

Käesolev töö on jaotatud kolmeks peatükiks. Esimeses peatükis räägitakse genereeritud andmete taustast ning nende jaoks loodud meetoditest. Teises osas kirjeldab autor oma praktilist tööd, tuues välja loodud programmi arhitektuuri ja algoritme. Viimases peatükis vaadeldakse saadud tulemusi, võrreldakse reaalsete andmetega, hinnatakse tulemuse kvaliteeti ning antakse nõu võimaliku tulevase arendamise osas.

1. Mõisted ja terminid

RHK-10 (ICD-10) – rahvusvahelise haiguste klassifikatsiooni 10. versioon on normatiivdokument, mida kasutatakse tervishoiu juhtiva statistilise ning klassifitseerimise raamistikuna [7].

SNOMED CT (SNOMED Clinical Terms) – üks täpseim, terviklikem ja samas keerukaim kliinilise meditsiini terminoloogia nomenklatuur maailmas.

Markovi ahel – matemaatiline mudel, milles üleminek ühest olekust teise toimub juhuslikult [8]. Kui ülemineku funktsiooniks on tõenäosus $P(S_k, a_i \rightarrow a_j)$, kus $S_k$on sisendsignaal ja $a_i \rightarrow a_j$on üleminek olekust a_i olekusse a_j, siis peaks täituma tingimus

$$\sum_{j=1}^{n} P(S_k, a_i \rightarrow a_j) = 1$$

OHDSI (Observational Health Data Science and Informatics) – avatud lähtekoodiga modulaarne lahendus, mille eesmärk on pakkuda laia valikut tööriistu meditsiiniliste andmete (OMOP CDM kujul) analüüsimiseks [6].

Observational Medical Outcomes Partnership Common Data Model (OMOP CDM) – mudel, mis teisendab erinevates andmebaasides olevaid meditsiinilisi andmeid ühtsesse formaati, mis võimaldab kasutada OHDSI standardiseeritud analüüsivahendeid ja -meetodeid [6].

Diagnooside trajektoor – edaspidi ka lihtsalt trajektoor või signaal on diagnooside järjend, kus üks diagnoos järgneb teisele mingis kindlas perioodis.

2. Andmete genereerimise meetodid

Andmete sünteesimine on hea praktika kohtades, kus on vajalik detailsem teadmine andmetest, et arendada nende analüüsiks uusi meetodeid nii, et oleks teada andmetes tegelikult paikneva info täpne olemus. Vastavaid lähenemisi kasutatakse paljudes valdkondades, kus on tegu väga sensitiivsete andmetega nagu kliinilised uuringud, tervishoid, finantsid jne, mida ei saa teadlastele ja tudengitele piisavalt vabalt jagada. Sealjuures on meditsiiniandmete genereerimine üks keerulisematest ülesannetest [4]. Käesolevas peatükis tehakse ülevaade kunstlikest andmetest ning tuuakse välja põhilised andmete genereerimistehnikad.

2.1. Kunstlikud andmed

Tänapäeval on andmed uus aare, mida paljud IT-ettevõtted uute toodete arendamisel ja ehitamisel kasutada soovivad [9]. Andmete abil on võimalik korjata statistikat erinevate protsesside kohta ning teha ka ennustusi, võttes kasutusele erinevaid masinõppe lahendusi nagu näiteks tehisnärvivõrgud. Kahjuks ei ole informatsiooni kättesaamine alati lihtne või üldse võimalik. Võib olla olukordi, mille korral andmeid veel ei eksisteeri, andmete eksportimine on kallis või piiravad hoopis privaatsusnõuded nende kättesaamist [10]. Teiseks tehakse andmete analüüsis alati teatud eeldusi andmete kohta, kuid ei pruugi teada, kas need eeldused paika peavad. Seega ei saa vastavaid analüüsimeetodeid eriti hästi arendada, kui näiteks ei ole teada, kui palju müra andmetes võib olla selleks, et mõni meetod veel töötaks. Sellist tüüpi probleemide puhul on võimalik võtta kasutusele kunstlikud andmed [9].

Kunstlikud genereeritud andmed on loodud sünteetiliselt, kasutades kindlat generatiivset mudelit, mitte saadud reaalsete sündmuste pealt. Nende mudelite põhieesmärgiks on olla rikkalik ja samaaegselt ka paindlik selleks, et neid saaks kasutada andmeanalüüsi meetodite arendamiseks ning ka teiste protsesside jälgimiseks ja testimiseks [11].

Sünteesitud andmete eeliseks on see, et kasutaja modelleeritud süsteemis on täpselt teada, missugusel viisil need genereeritud on. See tähendab, et andmete genereerimise juures ei saa tekkida olukorda, kui andmestikus esineb teadmata päritoluga anomaaliaid või puudulikkusi. Küll aga saab andmete genereerimise ajal lisada teadlikult müratasset või varieerida sisestatud signaali tugevust, et testida andmeanalüüsi programmide võimekust neid signaale sellest hoolimata tuvastada. Sünteesimine saab olla nii autonoomne kui ka reguleeritav inimese poolt

[12]. Genereeritud andmeid saab luua piiramatus koguses, hõlmates erineva perioodiga aega ning samaaegselt juhtida juhuslikku müra. Samuti ei rakendu sellistele andmetele privaatsuse poliitika ning andmete omanikuks saab pidada nende genereerijat [13].

2.2. Tervishoiu andmed

Tervishoiu valdkonnas suureneb sõltuvus andmetest iga päev, sest haiglad hoiavad aina rohkem arvutites patsientide informatsiooni. Andmekogu, mis sisaldab endas patsiendiga seotud ravimite, protseduuride ja diagnooside andmeid, nimetatakse digilooks [14]. Selliseid andmeid saab kasutada järgmistes kategooriates [4]:

- Haiguste diagnoosimine ja ennustamine – ei kulu enam lisaraha ega -aega valede diagnooside panemiseks, mis võib lisaks ka patsiendi tervist kahjustada. Siin saab kasutada ennustusmudeleid, mis ülesannet suurema täpsusega lahendavad.
- Ravitehnikate valik – nii arst kui ka patsient saavad valida parima ravivõimaluse erinevate ravitehnikate hulgast vastavalt tulevikuprognoosidele.
- Efektiivsete ravimite valik – on võimalik võrrelda ravitulemusi patsientidel, kes põdesid sama haigust, kuid kasutasid erinevaid ravimeid.
- Kindlustuspettuste ja kuritarvituste vähendamine – saadud andmete pealt on võimalik tuvastada patsientide ebaharilikke kaebuste mustreid. Seoses sellega on võimalik kokku hoida raha ning ravimpreparaate.

Meditsiiniliste andmete kättesaamine masinõppe analüüsiks on üsna problemaatiline isikuandmete privaatsuse tõttu. Eriti keeruline on see etapis, kus andmeanalüüsi meetodit ja tarkvara alles välja töötatakse, sest pole ette teada, kas ja millised signaalid on andmetes üldse tuvastatavad. Sellepärast ei ole enamikul juhtudel reaalsete andmekogumite kasutamine teostatav, kuna need on kättesaadavad ainult kliinilistes organisatsioonides. Näiteks tudengitele ülesannete andmisel või võistluste korraldamisel avalike andmetega ei pruugi olla võimalust jagada päris inimeste haiguslugusid. Küll aga oleks juhuslikud andmed priid vastavast privaatsuse murest.

2.3. Andmete sünteesimise meetodid

Andmete genereerimiseks võib kasutada erinevaid tehnikaid. Juhul, kui on saadaval tegelikud andmed, peab sünteesitud andmete valiidsust kontrollima. Tabelis 1 on välja toodud populaarsemad meetodid, mida kasutatakse tervishoiu andmete kontrollimiseks.

Tabel 1. Andmete analüüsimismeetodid [4]

Meetod	**Lühikirjeldus**	**Eelised**	**Puudused**
Otsustuspuu	Hierarhiline puustruktuur, mis koosneb reeglist kujul „Kui …, siis …“	• Ei vaja andmete normaliseerimist (töötab arvuliste ja kategooriliste andmetega) • Intuitiivne ja lihtsasti rakendatav	• Kui õppimiseks antud andmed ei ole tasakaalus, võib põhjustada kallutatust • Kipub ülesobituma
ANN (tehisnärvivõrk)	Matemaatiline mudel, mis on ehitatud bioloogiliste närvivõrkude põhimõttel. Tänu sellisele kujule nad on võimelised analüüsima ja jätma meelde erinevat tüüpi informatsiooni.	• On võimeline töötama mürarikaste andmetega • Tuvastab lihtsasti seoseid sõltuvate ja sõltumatute andmete vahel	• Kui õppimiseks antud andmed ei ole tasakaalus, võib põhjustada kallutatust • Kipub ülesobituma
Bayesi võrk	Suunatud tõenäosuslik graafiline mudel, mis koosneb olekutest ja üleminekutest	• Kiire ja täpne suurte andmekogumite jaoks • Lihtne struktuur	• Kui esineb muutujate vaheline sõltuvus, ei pruugi anda täpseid tulemusi
KNN (k-lähima naabri algoritm)	Algoritm, mis määrab objektid klassile, millel on enamus k lähimatest naabritest mitmemõõtmelises tunnusruumis	• Lihtne implementeerida • Treenimine on kiire	• Testimine on aeglane • Tundlik müra suhtes

Samuti saab genereeritud andmeid visualiseerida ning võrrelda tõeste andmetega, selleks võib visualiseerida andmete trajektoore või nende jaotust. Järgmised alapeatükid kirjeldavad andmete sünteesimismeetodeid.

2.3.1. SMOTE ja ADASYN

SMOTE ja ADASYN on klassifitseerimisalgoritmid, mida kasutatakse tasakaalustamata andmekoguste puhul [15]. See tähendab, et kui andmestikus on ühe klassi näidete arv palju väiksem võrreldes teistega, siis andmeid genereeritakse antud tehnikate abil. SMOTE põhimõte seisneb selles, et leitakse n lähim vähemusklassi naaber ning nende vahele tõmmatakse joon. Saadud sirgetele paigutatakse juhuslikult punkte ehk vähemusklassi andmestikke. ADASYN on SMOTE täiustatud versioon, mille puhul paigutatakse andmed mitte sirgele, vaid kallutatakse juhuslikult sirgest eemale [16].

2.3.2. Bayesi võrgud

Bayesi võrgud kujutavad andmeid tõenäosusgraafina, mis võimaldab üsna lihtsalt simuleerida uusi, sünteetilisi andmeid. Sellised võrgud koosnevad kahest komponendist: graafiline struktuur ja tingimuslike tõenäosusjaotuste kogum. Antud meetod on efektiivne, kui aluseks on vaja võtta suur hulk võimalikke juhtumeid [15].

2.3.3. Variatsioonilised autokooderid

VAE (*variational encoders*) on närvivõrkude eritüüp, mis üritab luua juhusliku uue väljundi, mis sarnaneb treeningandmetega. Antud algoritm üritab luua mitte niisama juhuslikult andmete variatsioone, vaid proovib seda teha konkreetses suunas [15, 17]. Näitena võib tuua meessoost pildi, millest sooviksime saada uut isikut. VAE võib lisada näiteks prillid või habeme, mille tulemusena saadakse samuti meessoost isik, kuid tegemist on juba sünteesitud ehk mitte eksisteeriva inimesega.

2.3.4. Generatiivne võistlev võrk

GAN (*generative adversial network*) on eelmisele alapeatükile 3.3.3 sarnane masinõppe algoritm, mis koosneb kahest närvivõrgust, millest üks genereerib andmeid ja teine püüab eristada õigeid andmestikke valedest [15, 18]. See tähendab, et omades sisendandmeid võrdleb ta neid loodud andmestikuga ning otsustab, kas saadud vektor võib vastata tõele või mitte.

3. Haiguste diagnooside ja nende trajektooride generaator

Antud peatükis kirjutab autor enda loodud mudelist ning andmete genereerimisest. Tuuakse välja programmi arhitektuur, selle komponendid ning andmete sünteesimise algoritm. Samuti toob autor välja erinevad moodulid – diagnooside sünteesimine, trajektooride lisamine, visualiseerimine. Kõik need osad on koondatud ühte projekti, kuid on sõltumatud ja üksteisest eraldatud. Kogu töö on kirjutatud Pythoni keeles versioonil 3.8.

3.1. Meetodi valik

Nagu eespool mainitud, on andmete genereerimiseks olemas mitu erinevat lähenemist või algoritmi. Juhendajaga koos otsustati kasutada võimalikult lihtsat genereerivat mudelit, mille sisu oleks kerge kirjeldada, seetõttu välistati nn "musta kasti meetodid". Käesoleva töö juures valiti andmete sünteesimise meetodiks tõenäosuslik lõplik automaat, mille olekute vahelised ülekanded ja väljastatavad diagnoosid on sisuliselt peidetud Markovi mudelid. Markovi mudel on olekuautomaat, mille olekute vahel võib liikuda juhuslikult ja kus igas olekus saab väljastada etteantud tõenäosusjaotuse järgi sümboleid (näiteks diagnoose).

Antud meetod sai valitud mitmetel põhjustel. Esiteks, mudeli loomiseks ei ole vaja andmeid võrrelda teiste masinõppe mudelitega, mistõttu on mudel universaalne ning sobib erinevate andmete sisestamiseks. Samuti on suureks eeliseks see, et automaadi saab panna kokku erinevatest sõltumatutest osadest, antud juhul olekutest, mis annab kasutajale vabadust muuta kogu süsteemi sõltuvalt oma nõuetest. Viimaseks eeliseks masinõppe mudelite ees on arusaam kogu protsessist. See tähendab, et andes tehisnärvivõrkudele andmeid õppimiseks, ei tea kasutaja muidu ise lõpuni, mille järgi andmeid sünteesima hakatakse ning mida võetakse antud protsessi aluseks, kuid Bayesi võrgu sarnastel süsteemidel on kõik läbipaistev ja reguleeritav.

Eesmärgiks on kitsalt luua diagnooside loetelu, mida võib pidada isikule elu jooksul väljastatud diagnoosideks. Igal isikule väljastatud diagnoosil on kindel aeg ning neil on eeldatavad ajalised sagedasemad järgnevused, sealjuures krooniliste haiguste diagnoosid püsivad läbi kogu ülejäänud elu ning iga indiviidi elu lõpeb surmaga. Omavahel seotud diagnoosidest tekivad „trajektoorid", kus näiteks haigus aja jooksul progresseerub. Haiguste diagnooside aluseks on RHK-10 koodid, mis on organiseeritud peatükkide ja alamkategooriate kaupa kuni nelja tasemega hierarhiasse.

3.2. Süntees

Programmi eesmärgiks oli luua fail, mis hoiaks endas sünteesitud tehispatsiente koos järgmiste andmetega:

- ID – isikut identifitseeriv number
- Sugu
- Sünnikuupäev
- Surmakuupäev
- Vanus (surma- ja sünnikuupäeva vaheline aastate arv)
- RHK-10 peatükk – massiiv elementidest kujul *(peatüki kood, diagnoosi ilmumise päev)*
- RHK-10 alamkategooria – massiiv elementidest kujul *(peatüki alla kuuluv alampeatüki kood, diagnoosi ilmumise päev)*
- RHK-10 jaotis – massiiv elementidest kujul *(alampeatüki alla kuuluv jaotise kood, diagnoosi ilmumise päev)*
- RHK-10 alamjaotis – massiiv elementidest kujul *(jaotise alla kuuluv alamjaotise kood, diagnoosi ilmumise päev)*

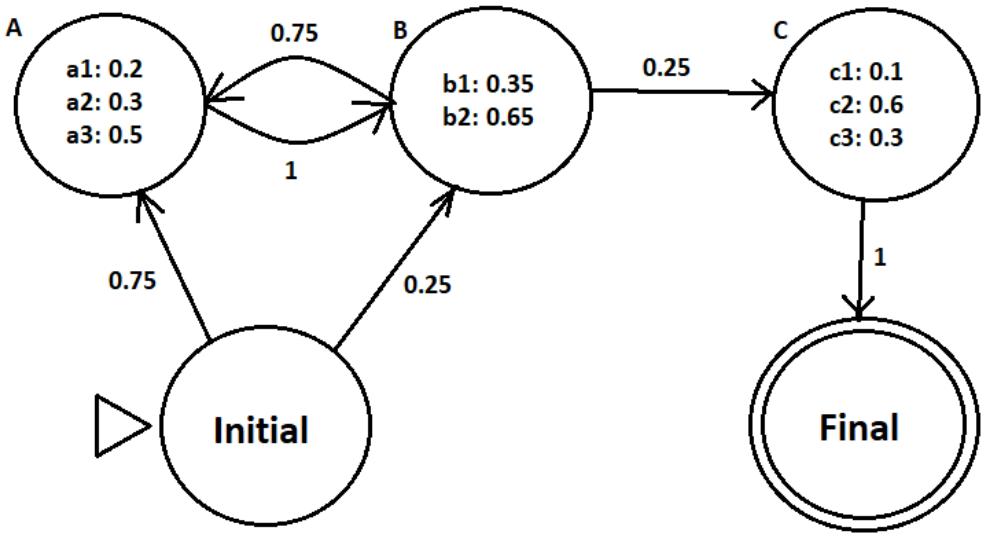
Kõik diagnoosidega seotud kodeeringud baseeruvad rahvusvahelise haiguste klassifikatsiooni RHK versioonil 10 (ICD-10) [7].

3.3. Mudeli arhitektuur

Praktilise töö alguses otsustas autor koos oma juhendajaga, et genereeriv automaat peaks koosnema väikestest ja võimalikult lihtsatest automaatidest, mida saaks omavahel mugavalt siduda. Täiendavaks tingimuseks on kasutaja võimalus muuta mudelit vastavalt oma nõuetele, sealhulgas olekute ning tõenäosuslike üleminekute lisamine, muutmine ja kustutamine.

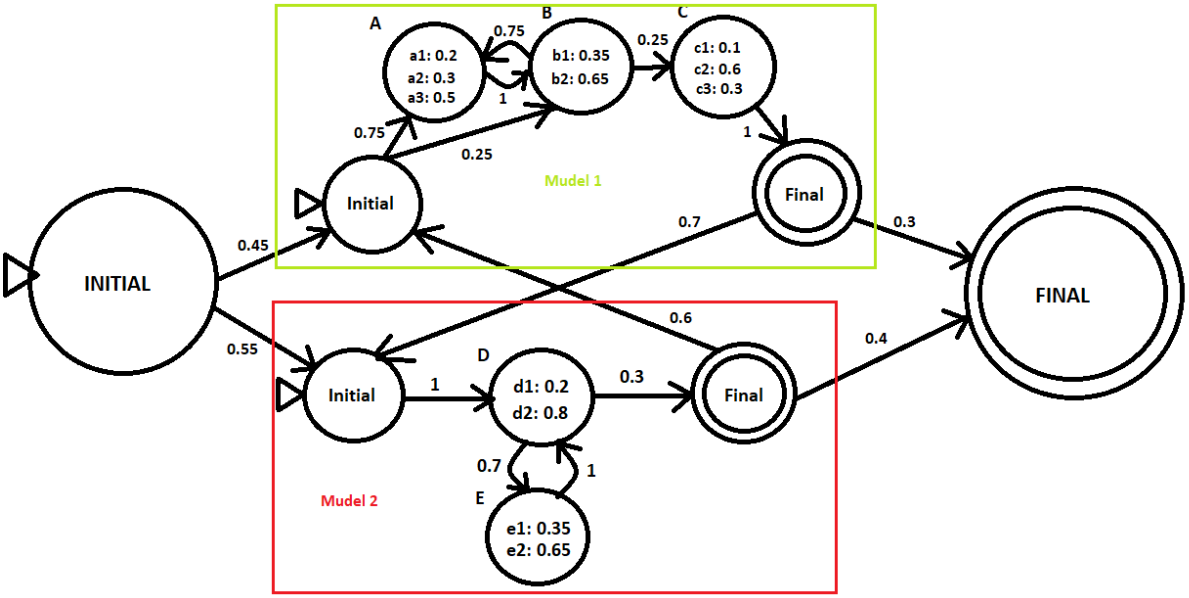
3.3.1. Esialgne mudel

Esimese etapina lõi autor väga lihtsa mudeli, mida on kujutatud joonisel 1. Antud mudel koosneb viiest olekust, millest kaks olid algolek (*Initial*) ja lõppolek (*Final*). Põhimõte seisneb selles, et igas olekus on võimalik liikuda kindla tõenäosusega teise olekusse. Siis kui programm jõuab olekusse, tagastab mudel oleku sees oleva informatsiooni, mis on samuti seotud tõenäosusega.



Joonis 1. Esialgne mudel

Kuna selline automaat oli juba töötav ning oskas genereerida sõnede järjendeid (näiteks a3b2a3b1a2b2c2 ja b2a3b2a1b1a3b1a3b2c3), siis mõtles autor välja süsteemi, mis nägi välja, nagu on kujutatud joonisel 2. Seejärel oli igas alamautomaadis võimalik üleminek teise alamautomaati. Sellise arhitektuuri juures käivitatakse esialgu suure automaadi algolek, mis käivitab väiksemaid automaate, mis omakorda hakkavad kas üksteisele viitama või jõutakse mingil hetkel suure automaadi lõppolekusse ning programm lõpetatakse.



Joonis 2. Kaks ühendatud automaati ja nende baasil genereeritud võimalikud järjekorrad

Mudeli olekute ja üleminekute andmed võeti konfiguratsiooni failidest, mida kasutaja sai iseseisvalt muuta (Joonis 3).

```
name: Diagnosis 2
states:
 - D
 - E
start_states:
 D: 1
final_states:
 D: 0.3
transition_probability:
 D:
  E: 0.7
 E:
  D: 1
diagnoses:
 D:
  d1: 0.2
  d2: 0.8
 E:
  e1: 0.35
  e2: 0.65
```

Joonis 3. Lihtsa automaadi konfiguratsiooni fail

Vaatamata sellele, et sellise arhitektuuriga automaat töötas hästi ning oli võimeline sünteesima andmeid, leidus sellel ka mitu puudust. Vaadates 2. joonisel mudel 2 olekut, puudub D kasutajal võimalus lisada otse üleminekut mudelile 1 B olekusse, mille tõttu kaob üksteisest sõltumatute automaatide loomise tähendus. Samuti konfiguratsiooni failis, mis paneb paika automaadi kuju, jäi vahele tähtis aspekt – tõenäosuse sõltuvus isiku vanusest ja soost ehk pikemas perspektiivis ei tohiks automaat genereerida rasedusega seotud diagnoose meessoost isikutel.

3.4. Markovi ahel

Kuna autor võttis diagnoosi koodide aluseks RHK-10 süsteemi, siis on uus mudeli arhitektuur puukujuline ning üles ehitatud neljal põhitasandil – peatükk, alampeatükk, jaotis, alamjaotis. Igaüks neist koosneb iseseisvatest olekutest, mis omakorda viitavad teistele olekutele.

3.4.1. Oleku kuju

Iga olek omab endale vastava RHK-10 koodi. Peale seda nende objektide lahutamatuks osaks on esinemise tõenäosus sõltuvalt isiku soost ja vanusest. Samuti on võimalik lisada

lõppolekutele (olekud, mis ei viita enam teistele olekutele) ülemineku tõenäosust kõigisse olekutesse. Ülemineku puudumisel viib olek mudeli vaikimisi algolekusse. Viimased oleku näitajad on diagnoosi ilmumiste arv ehk võib esineda mitu korda elu jooksul või ainult üks kord ning haiguse kroonilisus: kas haigus on krooniline või mitte.

3.4.2. Tõenäosused

Nagu eelnevalt mainitud, peab iga olek kindlasti omama tekkimise tõenäosust. Selleks, et kasutajal oleks neid lihtsam paika panna, on programmis võimalus neid lisada protsentuaalselt vahemikus 0 kuni 1.
Kogu mudel võib koosneda rohkem kui 20 000 olekust, mille tõttu nende käsitsi lisamine ei ole väga ratsionaalne. Antud töö raames võttis autor aluseks andmed veebilehelt[1], mis kajastab kõigi Eesti elanike tegelike diagnooside kogusagedusi eri vanuses konkreetse aasta lõikes. Kuigi antud allikas omab informatsiooni ainult rahvusvahelise haiguste klassifikatsiooni peatükkide ja alampeatükkide kohta, on programmis vaikimisi olemas ka jaotise olekud, millel on sama tõenäosus sõltumatult vanusest ja soost.
Peale erinevate diagnooside tekkimise tõenäosuste korjas autor informatsiooni surma kohta, saades informatsiooni Eesti Statistikaameti veebilehelt [19].
Jõudes mudeli lõppolekutesse on kasutajal samuti olemas võimalus lisada tõenäosuslikku üleminekut teistesse olekutesse, kaasa arvatud algolekusse. Näiteks, jõudes olekusse, mis kirjeldab mingit X diagnoosi, võib see omakorda tekitada diagnoosi Y.

3.4.3. Andmete sisestamine

Kasutajal on võimalik lisada ja muuta tõenäosusi läbi konfiguratsiooni faili. Konfiguratsiooni fail kujutab endast oleku kirjeldust, milles on paika pandud olekut kirjeldavad andmed: kood, tõenäosuslik sõltuvus ning üleminek teistesse olekutesse (juhul kui tegemist on lõppolekuga). Kõik failid hoitakse nendele kuuluvatele kaustades, mille nimedeks on *chapter* ehk peatükk, *subchapter* ehk alampeatükk, *section* ehk jaotis, *subsection* ehk alamjaotis.
Peatüki kaustas on failid peatükkide nimedega, .yml laiendusega. Teistes kaustades on kaustad vanemate koodidega, milles asuvad sellesse kategooriasse kuuluvad koodid. Näiteks alampeatüki kaustas asub mingi peatüki nimega kaust, olgu see A00-B99, millesse kuuluvad

[1] https://www.stacc.ee/ehif-stacked-area/?lng=Et

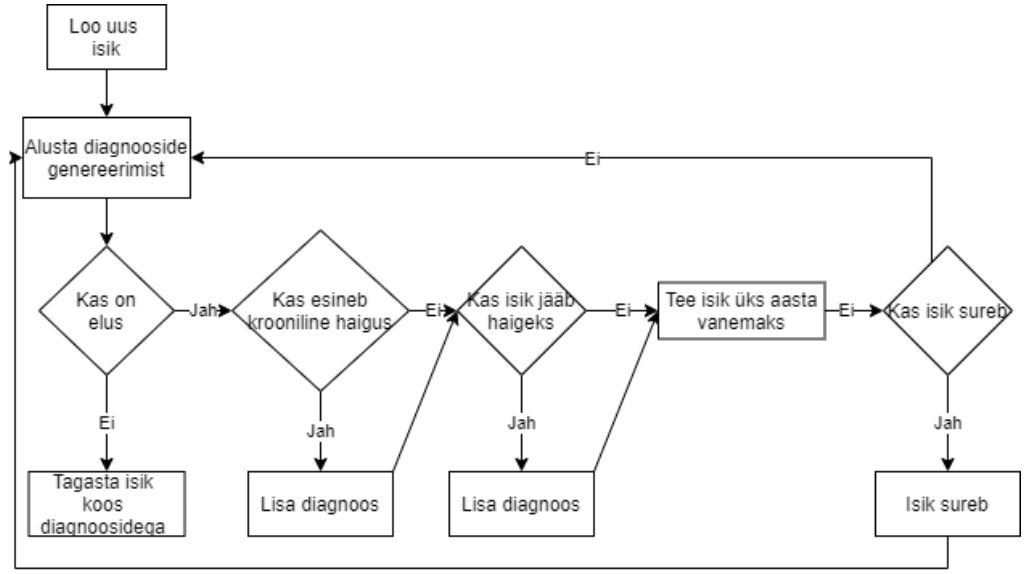
failid A00-A09.yml, A15-A19.yml ja teised. Teisisõnu, antud kauststruktuuris on järgitud hierarhilist järjestust.

3.4.4. Algoritm

Enne andmete genereerimist loeb programm sisse kõik kasutaja poolt sisestatud andmed, loob nendest olekuobjekte, seob neid ning loob ühe suure puusarnase mudeli. Pärast seda alustatakse tsüklit, mida on kujutatud joonisel 4, mis kordub kasutaja poolt soovitud isikute arvu võrra. Tsükli alguses luuakse isik, kellele omistatakse juhuslikult sugu ning sünnikuupäev vahemikus 1920. aastast 2020. aastani. Loodud isik antakse üle mudelile, mis hakkab kasutajalt tsükliliselt küsima 4 põhiküsimust:

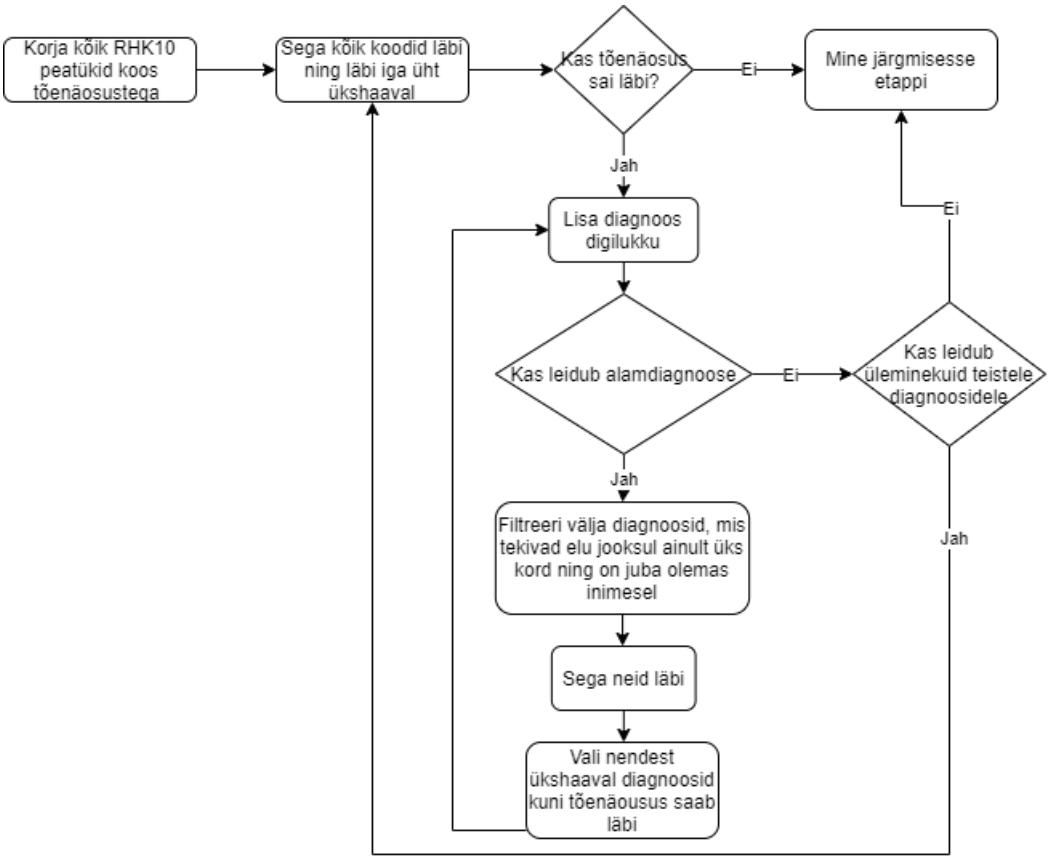
1) Kas isik on elus?
2) Kas esineb kroonilisi haigusi?
3) Kas isik jääb haigeks (kas omistatakse diagnoos)?
4) Kas isik sureb?

Kui esimese küsimusega selgub, et isik ei ole enam elus, siis tagastab mudel antud isiku koos kõikide diagnoosidega, mis talle elu jooksul pandud olid. Järgmise küsimusega kontrollitakse, kas esineb kroonilisi haigusi ning juhul, kui esineb, siis lisatakse sama haigus uuesti digiloo haiguste väljundisse. Kolmandas etapis otsustab algoritm, kas lisada isikule uut diagnoosi. Kui vastus osutub positiivseks, siis küsib mudel sünteesitud isikult tema sugu ja vanust ning käivitab endas oleva automaadi selleks, et tõenäosuse abil võimalik diagnoos leida. Viimases etapis otsustab mudel, kas inimene sureb või mitte. See tähendab, et kui programm otsustab, et isik sureb, siis sellele genereeritakse surma kuupäev ning algoritmi tsükkel peatakse. Sealjuures võetakse aluseks ainult inimese vanus ning vaadatakse Eesti 2019. aasta statistikat, mille järgi tehakse otsus.



Joonis 4. Mudeli algoritm

Raskeimaks osaks on algoritmi etapp, kus programm üritab lisada uut diagnoosi, see on kujutatud joonisel 5.



Joonis 5. Diagnoosi lisamise algoritm

Teades kõiki RHK-10 peatükke, käib programm neist igaühest läbi ning küsib nende tõenäosust, mis teoreetiliselt võib tähendada ka seda, et ühtegi peatükki ei valita. Joonisel 5 on olemas kirjeldus nagu „Kas tõenäosus sai läbi“, mis tähendab, et teades diagnoosi tekkimistõenäosust peab programm otsustama, kas haigus tekib või mitte. Selleks kutsub programm välja *random()*[2] funktsiooni, mis tekitab juhusliku arvu 0 ja 1 vahel. Kui saadud arv on väiksem, siis diagnoos lisatakse digilukku, muul juhul mitte.
Kui mingi peatükk osutub valituks, siis küsitakse sellele alluvaid diagnooside kategooriaid. Nüüd käiakse kõik kategooriad nii kaua läbi, kuni leitakse sobiv tõenäosus. Sama protseduuri korratakse, kuni mudelis leidub olek, millel ei ole alluvaid olekuid. Sellise lõppoleku juures kontrollitakse lisaks, kas tal leidub sõltuvusi ehk üleminekuid teistele diagnoosidele. Kui ei leidu, läheb mudel tagasi RHK-10 peatükkide juurde, kuni jõuab lõpuni välja.
Paralleelselt diagnooside lisamisele kontrollitakse, kas isikul on kroonilisi haigusi. Juhul, kui indiviidil oli varem krooniline haigus, siis hakkab see korduma vähemalt kord aastas kuni surmani.

3.5. Trajektoorid

Kuna ühe suure ühendatud mudeli eesmärgiks oli sünteesida diagnoose, siis trajektooride funktsionaalsuse lisamine tegi sünteesimise keeruliseks. Seetõttu otsustati luua eraldiseisvad tõenäosuslikud automaadid, mis olid struktuuri poolest sarnased, kuid täitsid teist ülesannet – trajektooride lisamist.

3.5.1. Mudeli kuju

Eespool mainitud mudelile sarnaselt koosneb antud automaat samuti olekutest ja üleminekutest. On olemas algolek, millest saab alguse trajektoor. Antud olek omab vastavat nimetust ehk RHK-10 koodi ning üleminekuid teistesse olekutesse. Üleminek kujutab endast massiivi olekutest, mis omavad koodi, ülemineku tõenäosust ning perioodi, mille jooksul antud diagnoos võib tekkida.
Antud mudelid asuvad projektis kaustas *data/trajectories*, mille sees on olemas eraldiseisvad kaustad koos iga trajektoori algoleku nimedega. Igas sellises kaustas asub vähemalt üks fail algoleku nimega (näiteks *I10.yml*), mille sisu on näidatud joonisel 6, ning sellele trajektoorile

[2] https://docs.python.org/3/library/random.html

kuuluvate olekutega ja antud trajektoori omavate inimeste arvu protsendiga. Teised olekud omavad täpset samasugust kuju, kuid nendel ei ole välja toodud protsenti.

```yaml
code: Z04
percent: 0.4
transaction:
  B20:
    probability: 0.2
    period: 3
  B89:
    probability: 0.8
    period: 3
```

Joonis 6. Trajektoori oleku kirjeldus

Samuti on olemas võimalus lisada *None* ehk tühja olekut, mis tähendab, et sellesse olekusse sattumise puhul trajektoor peatub. Juhul kui trajektooris esinevad samasugused olekud, kuid need suunavad erinevate diagnoosideni, siis koodi ja selle faili (laiendiga *.yml*) peaks nimetama *code_X*, kus *X* vastab numbrile. Iga järgnev number peab olema suurem eelmisest (näiteks *I10_1*, *I10_2* jne).

3.5.2. Trajektooride lisamine

Pärast peatükis 3.4.4. kirjeldatud diagnooside lisamist antakse sünteesitud isikud üle trajektooride osale. Isikud jaotatakse võrdsetesse gruppidesse ning igast grupist valitakse juhuslik arv inimesi, kellele hakatakse rakendama trajektoore. Iga trajektoor rakendatakse kindlale patsientide arvule (trajektoori sees olev *percent* parameeter). See tähendab, et leidub indiviide, kes ei oma üldse trajektoore, kes omavad mõnda trajektoori ning ka sellised, kes saavad omada kõiki trajektoore. Sellel viisil välditakse andmete sarnasust ning andmed hakkavad omandama realistlikku välimust.

Võttes ette ühe isiku, leitakse üles tema sünni- ja surmapäev. Saadud kahe kuupäeva vahele juhuslikult käivitatakse valitud trajektoori lisamise automaat. Selleks võetakse signaali algolek, mis lisatakse patsiendi ajalukku koos vastava ajaga. Pärast valitakse järgmine olek, millele viib eelmine olek. Saadud olek lisatakse jälle patsiendi digilukku koos juhusliku ajaga, mis kuulub eelmise oleku tekkimiskuupäeva ning antud oleku perioodivahemikku. Selline protsess jookseb seni, kuni jõutakse olekuni, mis enam kuhugi ei vii.

3.6. Andmete lisamine andmebaasi

Viimaseks tööetapiks oli andmete lisamine SQLite andmebaasi OMOP CDM kujul. Andmed võetakse failist, kuhu programm genereerib diagnoosid koos trajektooridega. Selles etapis tekkis probleem, kuna antud baas nõudis SNOMED'i, mitte rahvusvahelise klassifikatsiooni 10. versiooni koode. Õnneks oli autori juhendajal fail, mis kirjeldas SNOMED'i ja RHK-10 koodide suhet ning millistesse tabelitesse millised andmed lähevad, mistõttu oli autoril võimalik baasi lisada õiged koodid õigetesse tabelitesse. Antud fail ei ole avaliku juurdepääsuga, mille tõttu andmete lisamiseks andmebaasi peab võtma ühendust autoriga või bioinformaatika uurimisrühmaga.

Andmebaas koosneb kokku 38 tabelist, kuid antud töö raames piisas kuuest tabelist:

- *CONCEPT* – identifitseerib iga kliinilise ühiku, antud juhul diagnoosi
- *PERSON* – hoiab endas iga patsiendi informatsiooni
- *OBSERVATION_PERIOD* – näitab ajavahemikku, mille jooksul registreeriti isikul kliinilised sündmused
- *CONDITION_OCCURANCE* – diagnoosid, mida arst on täheldanud või millest patsient ise teatas
- *PROCEDURE_OCCURANCE* – protsessid või tegevused, mida patsiendile on tehtud
- *OBSERVATION* – diagnoosid, mis on leitud uuringu või protseduuri ajal

Andmete lisamine oli vajalik selleks, et tulevikus oleks võimalik võrrelda sünteesitud andmeid tõeliste andmetega ning võtta neid arendamiseks teistes uurimisrühmades.

4. Tulemused

Antud töö tulemuste hindamine on üsna subjektiivne, kuna reaalses elus võib tekkida erinevaid olukordi seoses inimese tervisega. Siiski õnnestus autoril visualiseerida saadud mudelit ning jälgida diagnooside jaotust.

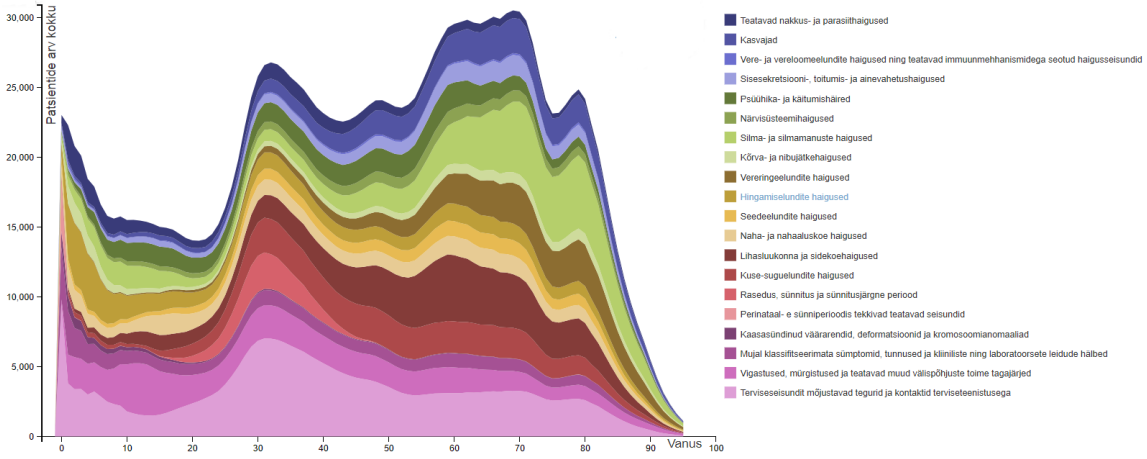
4.1. Visualiseerimine

Kasutusele võeti Python'i visualiseerimisteegid:

- matplotlib.pypplot[3] – funktsioonide kogu, mida kasutatakse graafikute joonistamiseks
- pyvis.network[4] – teek, mis võimaldab visualiseerida graafide võrgustiku

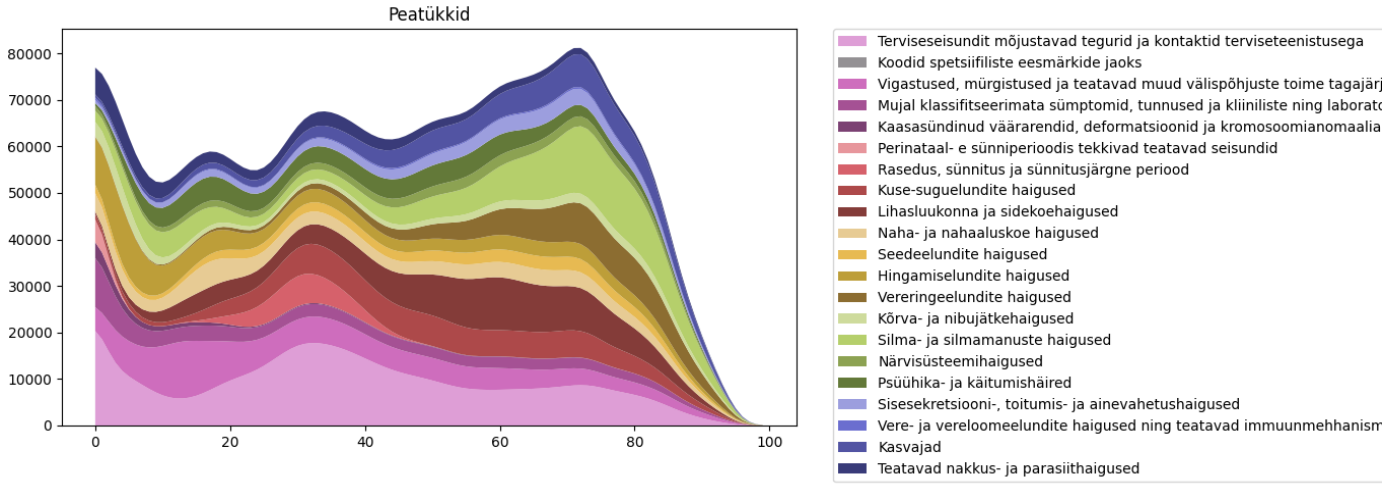
Graafi visualiseerimine annab võimaluse näha, kuidas näeb välja mudel ning kuidas erinevad olekud omavahel seotud on. Joonisel 11 on hästi näha, et võrk on puukujuline, kuna lõppolekud antud juhul ei vii teistele olekutele. Samuti on antud võrk tõlgendatud JavaScript keelde, mille tõttu saab seda kujutist brauseris avada, sisse ja välja suumida, klõpsata olekutele, mis näitavad oma naabreid, ning üleminekutele, mis näitavad selle tõenäosust. Kokku on antud graafis 1970 olekut.

Samuti sai genereeritud 50 000 isikust luua jaotusgraafiku (Joonis 8), mis näitas, mis vanustes millised diagnoosid esinesid. Võrdluseks sai võtta originaalgraafiku (Joonis 7), mis näitas, milline seis oli Eestis 2019. aastal. Peale selle saab võrrelda ka alampeatükke joonistel 9 ja 10, kus aluseks on võetud teatavad nakkus- ja parasiithaigused.

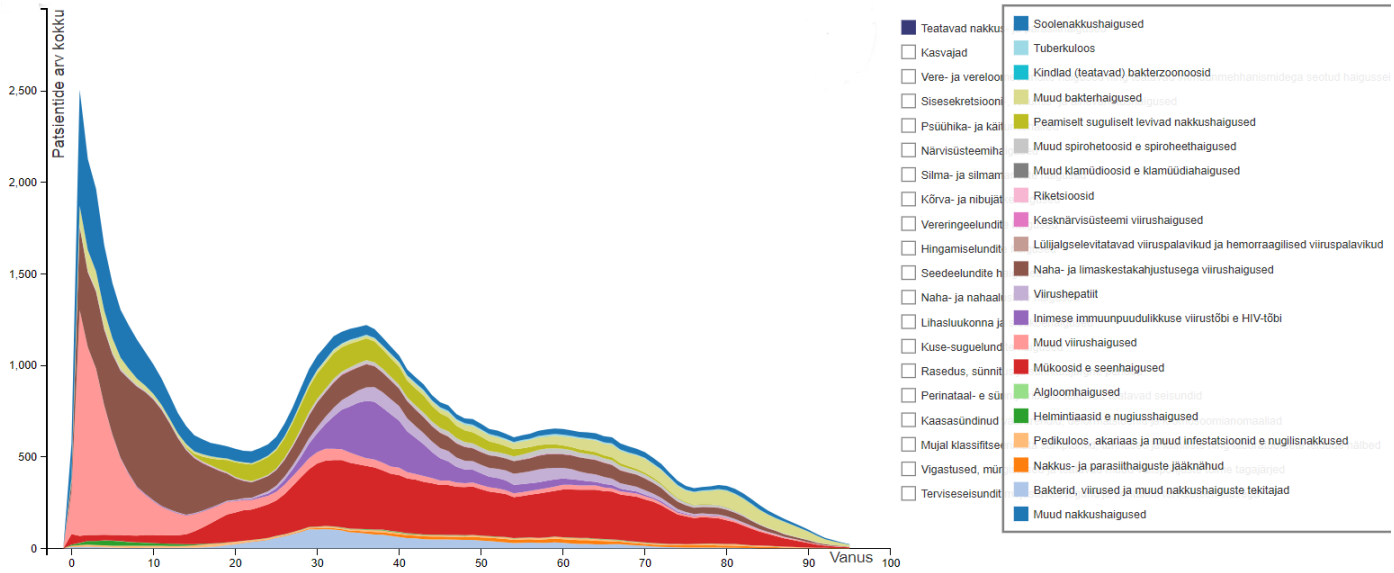


Joonis 7. Originaalgraafik diagnooside peatükkidest

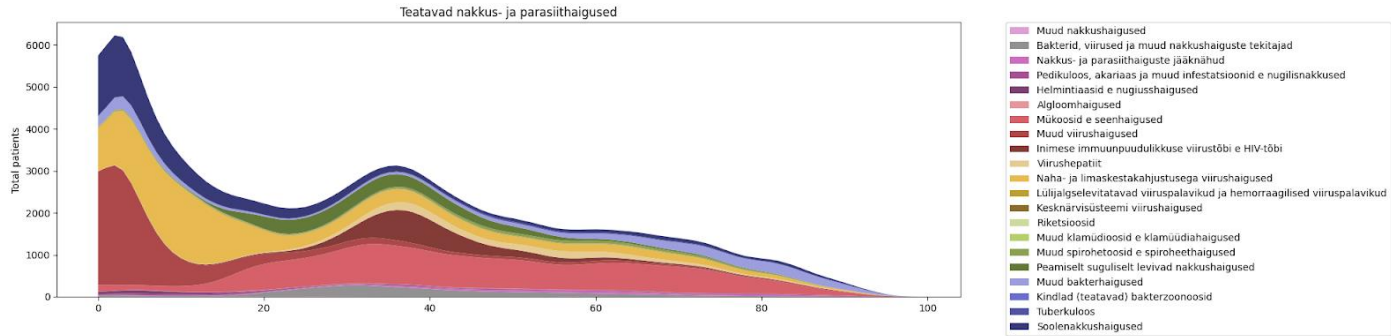
[3] https://matplotlib.org/2.0.2/users/pyplot_tutorial.html
[4] https://pyvis.readthedocs.io/en/latest/_modules/pyvis/network.html



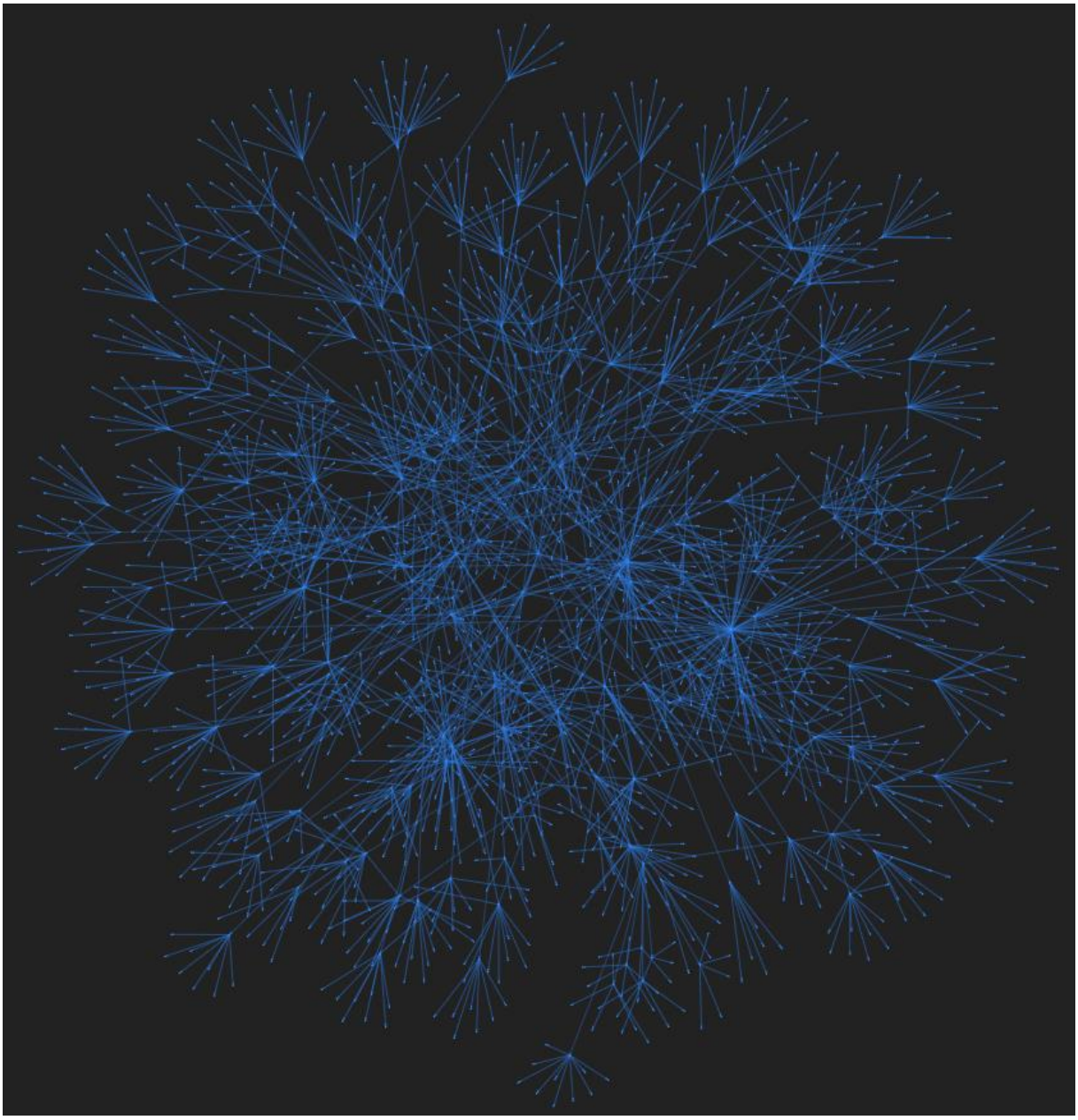
Joonis 8. Genereeritud andmetega graafik diagnooside peatükkidest



Joonis 9. Originaalgraafik nakkus- ja parasiithaigustest



Joonis 10. Genereeritud andmetega nakkus- ja parasiithaiguste graafik



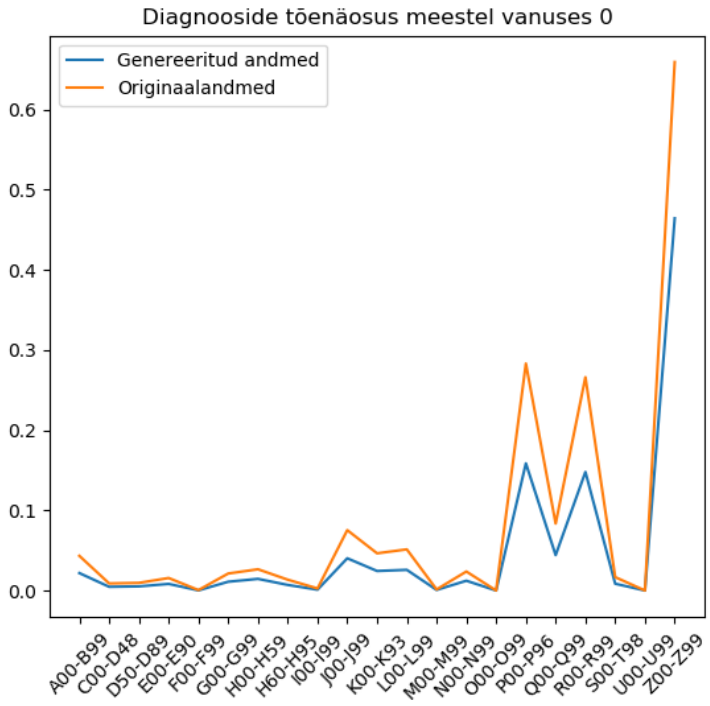
Joonis 11. Mudeli võrgustik

Programmis on võimalik joonistada kõiki alampeatükkide diagnoose, põhinedes diagnoosi koodile ning isiku soole.

4.2. Tulemuste analüüs

Tulemuste analüüsiks kasutas autor enda loodud graafikuid, mille peal oli võimalik jälgida diagnooside jaotust. Vaatamata sellele, et graafikud on visuaalselt sarnased ja neil esinevad samade vanuste puhul samad diagnoosid, ei olnud see piisav.

Autor võrdles originaalsete ja loodud andmete diagnooside tekkimistõenäosusi, mis on välja toodud joonisel 12.

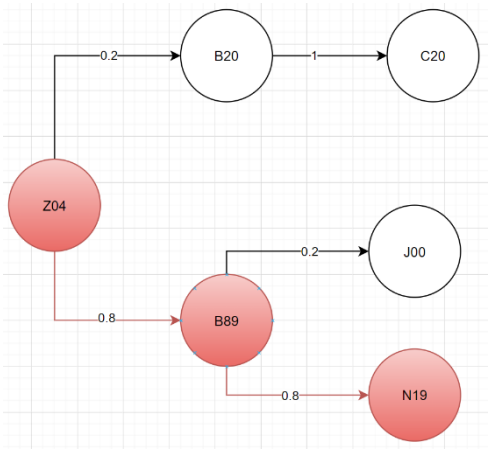


Joonis 12. Diagnooside tekkimistõenäosuste võrdlemine

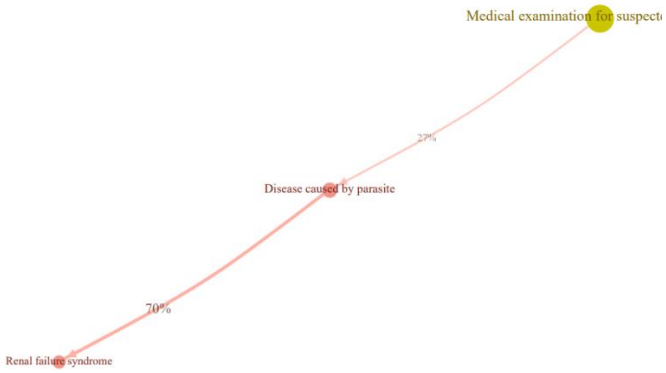
Diagnooside tõenäosuste vaheliseks keskmiseks kauguseks tuli 0.031 ning loodud isikute keskmiseks elueaks tuli 77.78 aastat. Saadud arvude järgi võib väita, et loodud andmed on lähedased reaalsusele, kuid siiski ei võimalda need hinnata saadud diagnooside kvaliteetsust.

Samuti oli autoril võimalus saata sünteesitud andmetega täidetud SQLite andmebaas andmeteaduse õppetooli bioinformaatika rühmale, kes tegelevad haiguste trajektooride leidmisega.

Trajektooride tulemuste kontrollimiseks autor on loonud lihtsa trajektoori, mis on kujutatud joonisel 13 ning genereeris 1000 isikut. Loodud indiviidide juures leidus 83 inimest, kellel leidus diagnoos „Läbivaatus ja jälgimine muul põhjusel“ (Z04), mis oli antud trajektoori algolek. Enamik inimesi on läbinud kõige tõenäolisema trajektoori teed, milleks oli Z04 -> B89 -> N19. Seda sama teed tuvastas ka bioinformaatika uurimisrühmalt, mida on hästi näha joonisel 14.



Joonis 13. Autori poolt loodud trajektoor



Joonis 14. Bioinformaatika uurimisrühmalt saadud trajektoor

Autor sai positiivse tagasiside, kuna ei leitud anomaaliaid, genereeritud andmed vastasid enam-vähem tõele ning saadud andmetest tulid välja trajektoorid.

4.3. Edasiarendamise võimalused

Kuigi programm töötab ning genereerib enam-vähem adekvaatseid andmeid, saab saadud mudelit siiski edasi arendada. Selleks, et andmed veelgi täpsemad oleksid, peaks lisama eraldi surmade tõenäosusi sõltuvalt soost. Samuti saaks mudelis kõiki arvulisi väärtusi vahetada uuemate vastu, kuna praegu pärineb kogu andmestik 2019. aastast.

Peale numbriliste asenduste peaksid võimalikud tulevased arendajad välja mõtlema sõltuvussüsteemi. Praeguses programmis on võimalik lisada sõltuvusi ühest olekust teise, kuid ei arvestata läbi põetud ega võimalikke pärilikke haigusi. See tähendab, et näiteks kui indiviidil oli lapsepõlves tuberkuloos, siis tõenäosus taas mõnda kopsuhaigust põdeda on suurem ning see võib ajaperioodis muutuda. Samuti oleks vajalik erinevate välistegurite lisamine, lähtudes elukeskkonnast, mis võivad mõjutada inimese tervist.

Lisaks saaks trajektooride esinemist muuta konkreetsemaks. See tähendab, et praeguses programmis ilmuvad need juhuslikult, sõltumatult diagnoosidest, kuid tegelikult peaksid trajektooridele eelnema kindlas järjekorras teised diagnoosid, mis võiksid üksteisega sõltuvustes olla.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua mudel, mis oskab genereerida võimalikult reaalsusele lähedased diagnoosid rahvusvahelise haiguste klassifikatsiooni 10. versiooni kodeeringus koos tekkimise kuupäevaga.

Töö käigus loodi süsteemid, mis põhinevad Markovi ahelatel. Üks mudel koosneb 1970 olekust ning teine mitmest väiksemast trajektoori mudelist. Mudelite aluseks võeti Eesti rahvastiku 2019. aasta andmed. Projekt on loodud kasutajasõbralikult, saab lisada või muuta diagnooside jaotust või haiguste trajektoore. Programmis on võimalik visualiseerida sünteesitud andmeid ning lisada ka OMOP CDM kujul andmebaasi. Kõik loodud moodulid asuvad ühe projekti sees, kuid on üksteisest sõltumatud.

Kuigi töö peamine eesmärk on täidetud, on meditsiiniliste andmete genereerimise juures palju faktoreid, mida peab diagnooside sünteesimise aluseks võtma, alustades pärilikkusest ja lõpetades elustiiliga. Selle tõttu on antud tööl hea alus, mida saab teha aina keerulisemaks, lisades igale isikule erinevaid elufaktoreid, mis mõjutavad tervist ning haiguste tekkimist.

Kasutatud materjalid

[1] N. SHahid, T. Rappon and W. Berta, "Applications of artificial neural networks in health care organizational decision-making: A scoping review," Toronto, 2019.

[2] Sciforce, „Machine Learning in Agriculture: Applications and Techniques," Sciforce, 22 Märts 2019. [Võrgumaterjal]. Available: https://medium.com/sciforce/machine-learning-in-agriculture-applications-and-techniques-6ab501f4d1b5. [Kasutatud 20 Jaanuar 2021].

[3] Andmekaitse inspektsioon, „Isikuandmed ja töötlemine," 8 November 2019. [Võrgumaterjal]. Available: https://www.aki.ee/et/eraelu-kaitse/isikuandmed-ja-tootlemine. [Kasutatud 20 Jaanuar 2021].

[4] S. C. Pandey, „Data Mining Techniques for Medical Data: A Review," November 2016. [Võrgumaterjal]. Available: https://www.researchgate.net/publication/318130038_Data_Mining_Techniques_for_Medical_Data_A_Review. [Kasutatud 27 Veebruar 2021].

[5] J. Walonoski, M. Kramer, J. Nichols, A. Quina, C. Moesel, D. Hall, C. Duffett, K. Dube, T. Gallagher ja S. McLachlan, „Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *Journal of the American Medical Informatics Association,* kd. 25, nr 3, pp. 230-238, 2017.

[6] O. H. D. S. a. Informatics, The Book Of OHDSI, Independently published, 2019.

[7] Eesti Sotsiaalministeerium, Rahvusvaheline haiguste ja nendega seotud terviseprobleemide statistiline klassifikatsioon Kümnes väljaanne, Tallinn: Tallinna Raamatutrükikoda, 1996.

[8] J. Rocca, „Introduction to Markov chains," 25 Veebruar 2019. [Võrgumaterjal]. Available: https://towardsdatascience.com/brief-introduction-to-markov-chains-2c8cab9c98ab. [Kasutatud 7 Detsember 2020].

[9] T. Sarkar, „Synthetic data generation — a must-have skill for new data scientists," Towards data science, 19 Detsember 2018. [Võrgumaterjal]. Available: https://towardsdatascience.com/synthetic-data-generation-a-must-have-skill-for-new-data-scientists-915896c0c1ae. [Kasutatud 3 Veebruar 2021].

[10] C. Dilmegani, „Synthetic Data Generation: Techniques, Best Practices & Tools,“ 13 Jaanuar 2021. [Võrgumaterjal]. Available: https://research.aimultiple.com/synthetic-data-generation/. [Kasutatud 25 Veebruar 2021].

[11] C. Dilmegani, „The Ultimate Guide to Synthetic Data in 2021,“ AI Multiple, 2 Veebruar 2021. [Võrgumaterjal]. Available: https://research.aimultiple.com/synthetic-data/. [Kasutatud 3 Veebruar 2021].

[12] J. Eno ja C. W. Thompson, „Generating Synthetic Data to Match Data Mining Patterns,“ Juuni 2008. [Võrgumaterjal]. Available: https://www.researchgate.net/publication/3420044_Generating_Synthetic_Data_to_Match_Data_Mining_Patterns. [Kasutatud 5 Veebruar 2021].

[13] E. Siht, „OSKUS-TEST MUDELIL PÕHINEVATE SÜNTEETILISTE ANDMETE GENEREERIMINE,“ Tallinna Tehnika Ülikool, Tallinn, 2020.

[14] S. Schiff, M. Gehrke ja R. Möller, „Efficient Enriching of Synthesized Relational Patient Data with Time Series Data,“ Jaanuar 2018. [Võrgumaterjal]. Available: https://www.researchgate.net/publication/328756974_Efficient_Enriching_of_Synthesized_Relational_Patient_Data_with_Time_Series_Data. [Kasutatud 18 Veebruar 2021].

[15] F. Clemente, „Synthetic Data,“ 2 Aprill 2020. [Võrgumaterjal]. Available: https://medium.com/ydata-ai/synthetic-data-1cd0ba907609. [Kasutatud 6 Veebruar 2021].

[16] I. Bhattacharyya, „SMOTE and ADASYN (Handling Imbalanced Data Set),“ 3 August 2018. [Võrgumaterjal]. Available: https://medium.com/coinmonks/smote-and-adasyn-handling-imbalanced-data-set-34f5223e167. [Kasutatud 6 Veebruar 2021].

[17] I. Shakat, „Intuitively Understanding Variational Autoencoders,“ 4 Veebruar 2018. [Võrgumaterjal]. Available: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf. [Kasutatud 6 Veebruar 2021].

[18] J. Rocca, „Understanding Generative Adversarial Networks (GANs),“ 8 Jaanuar 2019. [Võrgumaterjal]. Available: https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29. [Kasutatud 6 Veebruar 2021].

[19] Statistikaamet, „RV56: Surnud surmapõhjuse, soo ja vanuserühma järgi,“ 2021. [Võrgumaterjal]. Available: http://andmebaas.stat.ee/Index.aspx?DataSetCode=RV56#. [Kasutatud 1 Märts 2021].

Lisad

I. Programmi GitHub repositoorium

Valminud tarkvaralahenduse programmikood, koos diagnooside konfiguratsiooni failidega on kättesaadavad aadressilt:
https://github.com/valart/diagnoses-synthesis

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Artjom Valdas,

1. Annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Juhuslike diagnooside trajektooride generaator**, mille juhendaja on Jaak Vilo, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Artjom Valdas

07.05.2021