

Eveliina Uusitalo

Ohjelmistorobotiikan korvaaminen tekoälyllä asiakastiedon keruussa ja käsittelyssä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinööriytyö

8.5.2018

Tekijä Otsikko	Eveliina Uusitalo Ohjelmistorobotiikan korvaaminen tekoälyllä asiakastiedon keruussa ja käsittelyssä
Sivumäärä Aika	28 sivua + 2 liitettä 25.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	tieto- ja viestintäteknikka
Ammatillinen pääaine	Smart Systems and Software Engineering
Ohjaajat	Kimmo Sauren Kari Tarvainen
<p>Insinöörityön tarkoituksena oli suunnitella tekoäly, jolla pystyttäisiin korvaamaan ohjelmistorobotiikan ratkaisuja tai joka vaihtoehtoisesti toimisi niiden tukena. Tekoälyn tarkoitus on kerätä ja hyödyntää asiakastietoja vakuutuskauppojen tukena.</p> <p>Hyödyntämällä koneoppimista tämänkaltaisissa tilanteissa kyettäisiin takaamaan suurempi myynti virhemarginaalin pienentyessä. Koska ohjelmistorobotiikan hyödyntäminen asiakastietojen keräämisessä on haastavaa ja pidemmällä aikavälillä epäkäytännöllistä, uuden teknologian löytäminen ja käyttäminen tiedon keräämisessä ja mallintamisessa on edullinen ratkaisu voittoa tavoittelevalle yritykselle.</p> <p>Raportin lisäksi työhön kuuluu markkinointiin suunnattu myyntidokumentti ja diaesitys, jotka on kohdennettu työn tilaajayrityksen asiakkaalle. Käsiteltävän aiheen myötä lopputulokseksi syntyneet dokumentit voidaan hyödyntää uuden teknologian markkinoinnissa ja näin tilaajayrityksen myynnin parantamisessa.</p>	
Avainsanat	ohjelmistorobotiikka, koneoppiminen, tiedonlouhinta.

Author Title	Eveliina Uusitalo Replacing robotic process automation with artificial intelligence on collecting and processing customer data
Number of Pages Date	28 pages + 2 appendices 8 May 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Smart Systems and Software Engineering
Instructors	Kimmo Sauren Kari Tarvainen
<p>The purpose of this bachelor's thesis was to design an artificial intelligence that could replace robotic process automation or alternatively act as a support of them. The purpose of artificial intelligence is to collect and utilize customer information in support of insurance sales. By utilizing machine engineering in such situations, it would be possible to guarantee higher sales as the margin of error diminished. Since the utilization of robotic process automation in collecting customer data is challenging and in the longer term it is impractical, finding and using new technology in collecting and modeling information is a cost-effective solution to profit for the target company. In addition to the report, the work includes a sales document and a slideshow presentation targeted at the client of the job subscription.</p> <p>As a result of this thesis, the resulting documents can be used to market new technology and thus subscribe to the company's sales improvement.</p>	
Keywords	robotic process automation, machine learning, data mining.

Sisällys

Lyhenteet

1	Johdanto	1
2	Ohjelmistorobotiikka	2
2.1	Käyttötarkoitus	2
2.2	Historia	6
2.3	Haasteet	8
3	Koneoppiminen	9
3.1	Yleisesti	9
3.2	Neuroverkot	12
3.3	Koneoppimisesimerkki	15
4	Tiedonlouhinta	19
5	Markkinointiaineisto yritykselle	23
6	Yhteenveto	28
	Lähteet	29

Liitteet

Liite 1. Markkinointimateriaali

Liite 2. Esitysmateriaali

1 Johdanto

Insinööriyön tarkoitus on esitellä tehokas ja toimiva tapa korvata tekoälyllä valmiiksi robotisoituja epävarmoja tai muuten epäkäytännöllisiä ratkaisuja ja näin nopeuttaa ja helpottaa finanssialan työntekijöiden työtä. Tekoäly kohdennettaisiin vakuutuspuolelle käsittelemään asiakkaiden tietoja ja profiloimaan heitä, tarkoituksena rakentaa asiakkaalle kohdennettu vakuutuspaketti, joka täyttäisi asiakkaan senhetkiset tarpeet.

Insinööriyössä alussa paneudutaan ohjelmistorobotiikkaan ja tekoälyyn työkaluina. Tarkoituksena on esitellä kyseisten mallien heikkouksia ja vahvuuksia etenkin finanssialan näkökulmasta, esitellä mahdollinen tarve muokata ja parantaa nykyisiä toimintatapoja sekä ratkaisuja. Lopussa etsitään ratkaisu löydettyihin ongelmiin ja esitellään toimiva malli parantaa työntekoa ja tätä kautta asiakastyytyvää.

Koska työn aihe on laaja ja sisältää monia erillisiä parametreja, on rajaaminen tärkeää. Tästä syystä työssä esitellään vain lopputuloksen kannalta relevantteja elementtejä eikä paneuduta pintaa syvemmälle suuriin aiheisiin, kuten koneoppimiseen tai tiedonlouhintaan.

Lopputuloksena syntyi raportin lisäksi asiakkaalle suunnattu markkinointimateriaali ja esitysmateriaali, joita Finanssi-Kontio, työntilaaaja yritys, pystyy hyödyntämään tekoälyn markkinoinnissa.

2 Ohjelmistorobotiikka

Ohjelmistorobotiikkaa käytetään automatisoimaan rutiininomaisia työtehtäviä, joiden tunnusmerkkinä on toistuvuus. Tämänkaltaisia tehtäviä ovat yleisesti automaattiset viestivastaukset, tiedon keruu eri kohdejärjestelmistä, hakemusten lukeminen ja täyttäminen ym. Ohjelmistorobotiikan käyttäminen muun muassa finanssi-, terveys- ja tuotantotalouden aloilla on jatkuvasti nousussa sen helppokäyttöisyyden vuoksi (1).

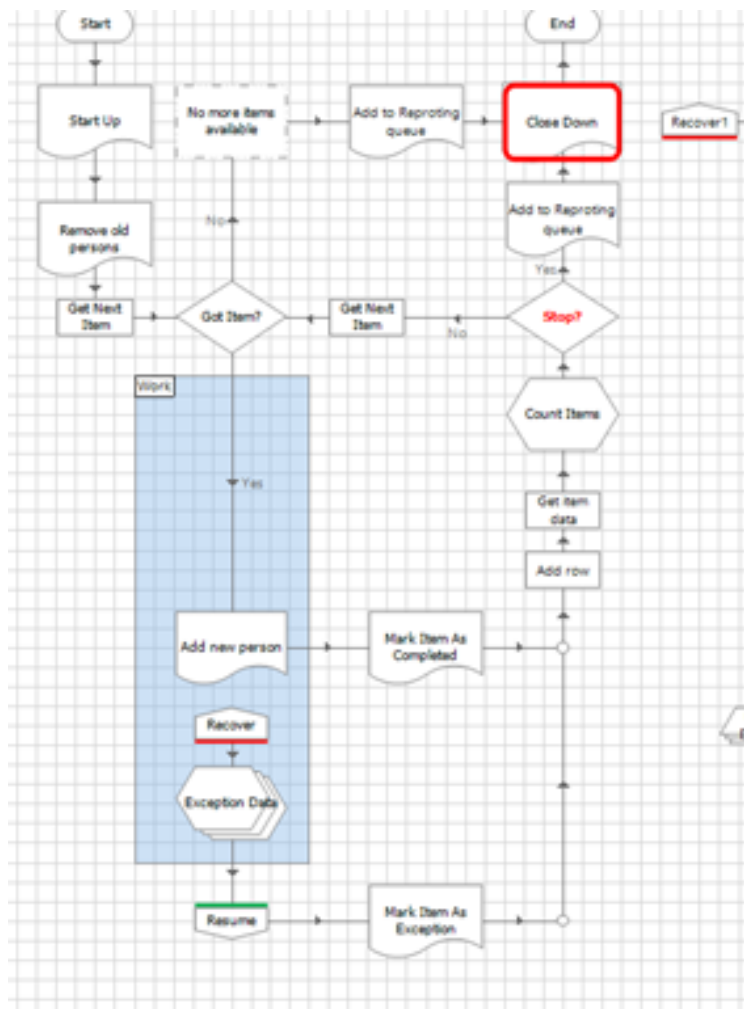
Institute for Robotic Process Automation on määritellyt ohjelmistorobotiikan seuraavasti:

“the application of technology allowing employees in a company to configure computer software or a ‘robot’ to capture and interpret existing applications for processing a transaction, manipulating data, triggering responses and communicating with other digital systems.” (2.)

Robotteja käyttämällä saadaan ohjattua työntekijöiden resurssit haastavampiin työtehtäviin ja näin kasvatettua yrityksen tuottavuutta. Ohjelmistorobotiikan sanotaan korvaavan keskimäärin kolme työntekijää ja maksavansa itsensä takaisin vuodessa (1). Tämä on kuitenkin vain raakaa dataa, ja jokainen ohjelmistorobotti työtehtävästään riippuen vapauttaa vaihtelevan määrän työntekijöiden resursseja.

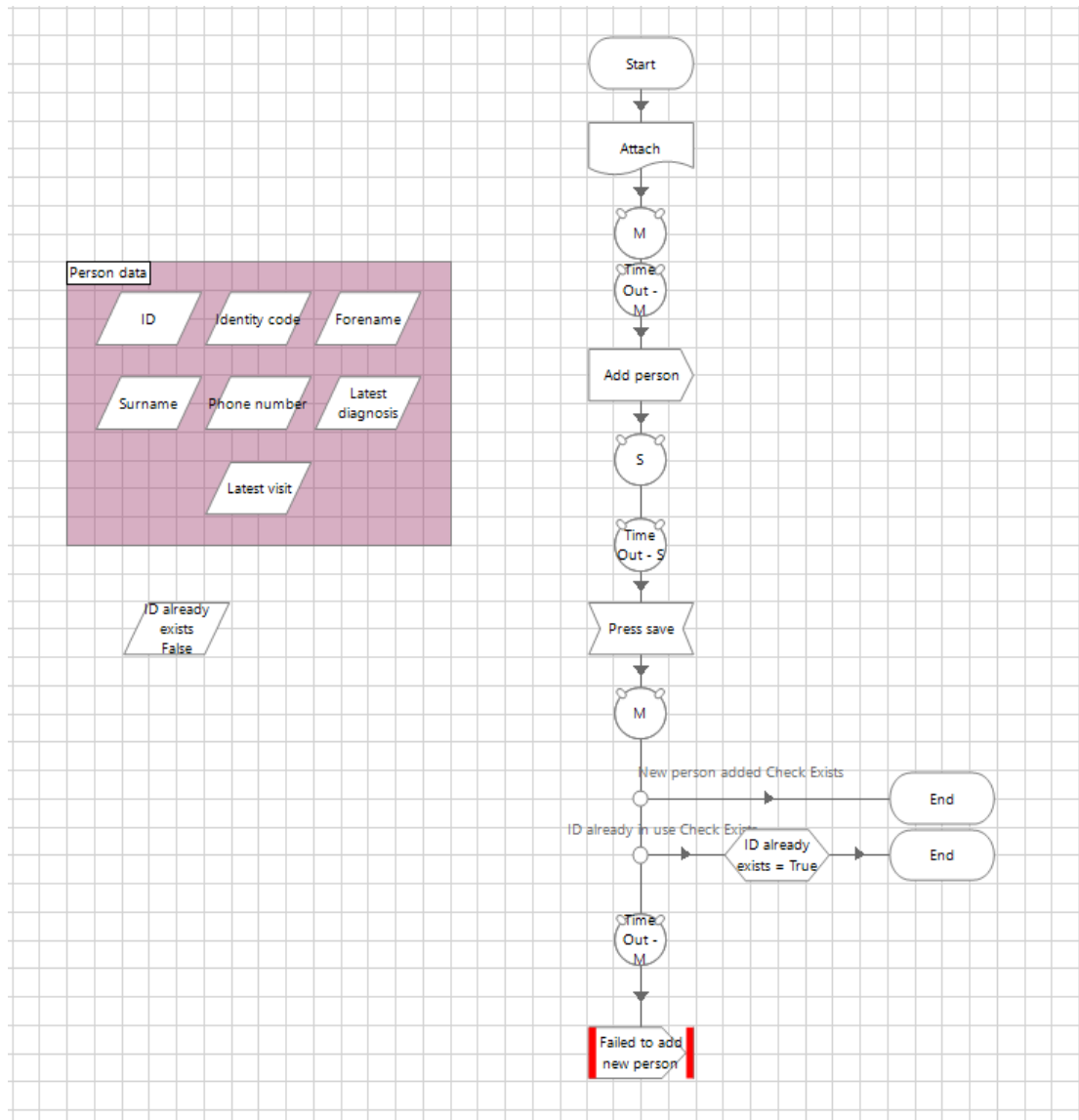
2.1 Käyttötarkoitus

Ohjelmistorobotiikka on kehitetty automatisoimaan mekaaninen tiedonkäsittely, joka tavallisesti veisi työntekijältä turhaan aikaa. Ohjelmistorobotiikan kehittyminen on alkanut 2000-luvun alkupuolella, mutta vasta viime vuosina se on alkanut saada nykyisen muotonsa ja yleistynyt suurten yritysten käytössä. Robottien vahvuudet näkyvät siinä, miten niiden voidaan kuvailla käyttäytyvän samalla tavalla kuin pääkäyttäjäkin, ilman inhimillisiä virheitä. Näin ollen ne pystytään mallintamaan tarkasti seuraamaan työntekijän liikkeitä ja vähentämään manuaalisen käsittelyn taakkaa. Itse mallintamiseen käytetään graafisen koodaamisen ohjelmia. Yksi suosituimmista ohjelmistorobotiikan käyttöympäristöistä on kuvassa 1 näkyvä Blue Prismin tarjoama ympäristö. (1.)



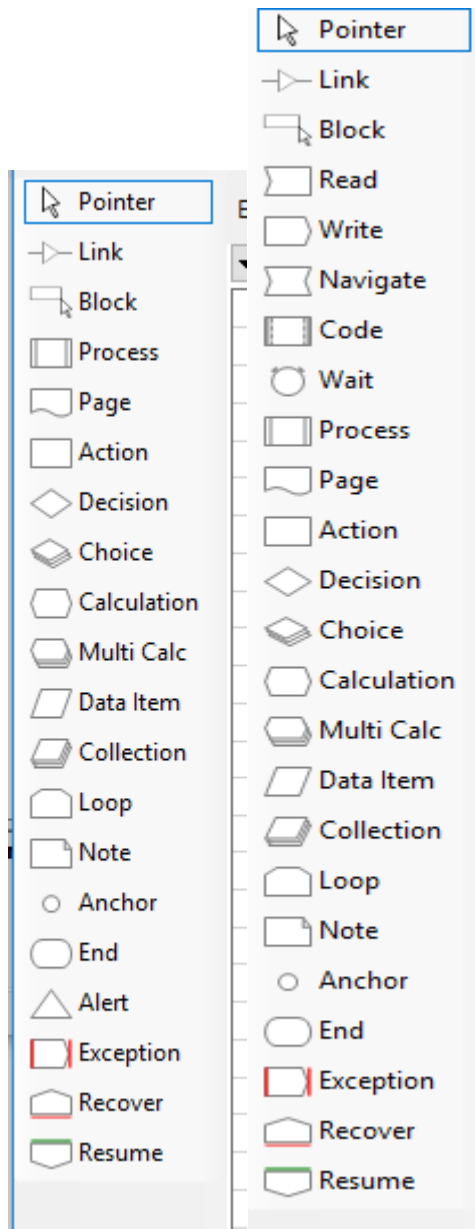
Kuva 1. Osa Blue Prismillä tehdyn prosessin pääsivua.

Ohjelmistorobotti koostuu itse pääprosessista ja siihen lisätyistä erillisistä businessobjekteista. Businessobjektit taas sisältävät useita funktioita (kuvassa 1 funktiot ovat merkitty suorakulmiolla). Jos ohjelmistorobotti tehtäisiin esimerkiksi Javalla, voitaisiin sanoa, että pääprosessi on ohjelman main-luokka ja businessobjektit muita luokkia, joissa luodaan funktioita, joita main-luokka käyttää. Esimerkkinä kuvan 2 businessobjektin funktio, jonka tehtävänä on täyttää internetsivuston kenttiä ja tallentaa tiedot.



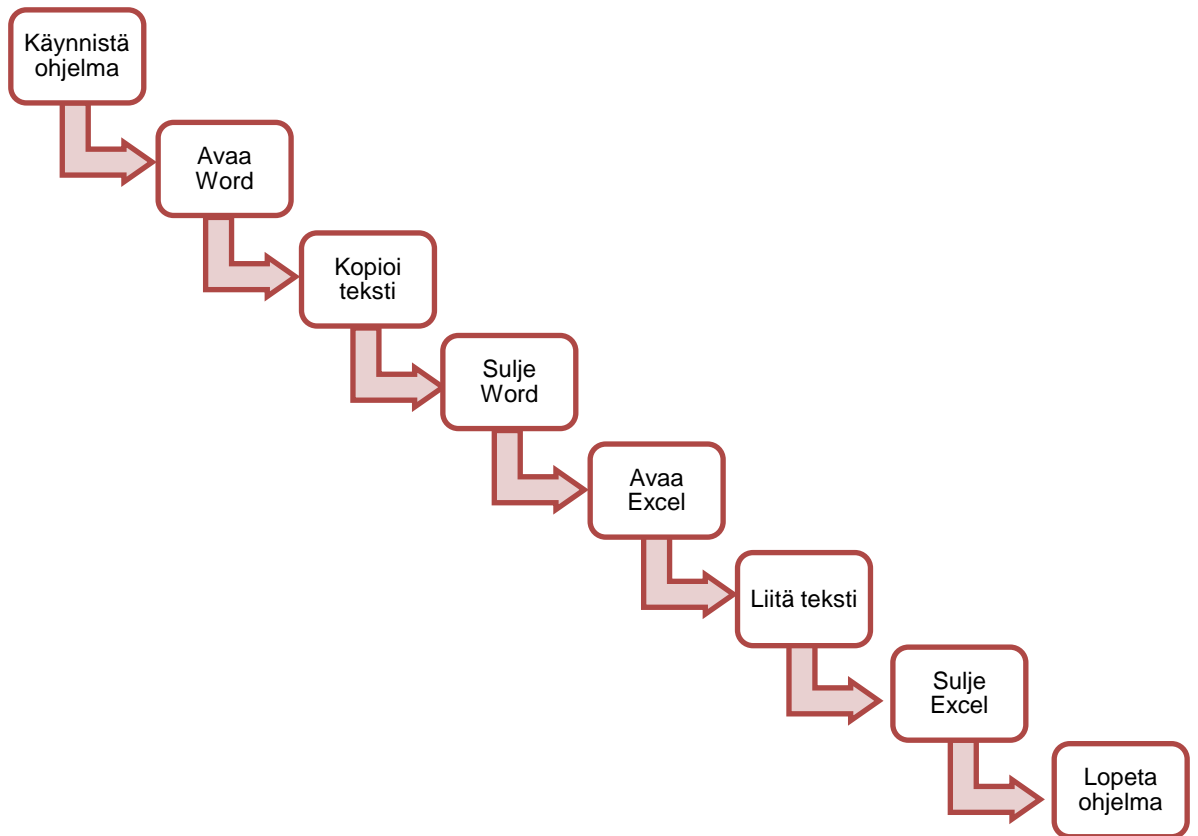
Kuva 2. Esimerkki businessobjektin funktiosta.

Prosessitason ja businessobjektin suurin eroavaisuus on eri toiminnot. Kun kuvia 1 ja 2 vertaa keskenään, on helppo huomata niiden näyttävän rakenteeltaan aivan erilaisilta. Kuvassa 3 voidaan havaita, miten BO sisältää muun muassa odotus tason, kun taas projektitasolla samanlaista toimintoa ei ole.



Kuva 3. Toimintavalikko, vasemmalla prosessitaso, oikealla BO-taso.

Esimerkkinä kokonaisesta prosessista voidaan käyttää yksinkertaista ohjelmistorobottia, joka kopioi tietoja Wordista ja luo niistä Excel-tiedoston. Käytännössä tämä tapahtuu kuvan 4 esittämällä tavalla. Esimerkin avulla on helppo havainnollistaa ohjelmistorobottiikan edut ja haitat. Erilaisten tiedostojen, kuten Excelin, Csv:n, Wordin ym. käsitteleminen ohjelmistorobottiikan avulla on helppoa, mutta se vaatii useita, erikseen rakennettuja businessobjekteja. (1.)



Kuva 4. Prosessikaavio.

Prosessikaaviossa ainoastaan vaiheet 1 ja 8 tapahtuvat itse prosessissa, loput ovat businessobjektien funktioita, joita pääprosessi kutsuu. Käytännössä ohjelmistorobotin mallintaminen vastaa monella tavalla ohjelmointia, vaikka tapahtuukin graafisesti. Tätä voidaan pitää yhtenä teknologian vahvuutena, sillä prosessin ymmärtäminen ei vaadi vahvaa ohjelmointitaustaa.

2.2 Historia

Ohjelmistorobotiikka on nykyisessä muodossaan tunnettu jo 2000-luvun alusta, vaikka se onkin vasta tällä vuosi kymmenellä noussut laajemmin yritysmailman käyttöön. Kehitys kuitenkin on alkanut monta vuosikymmentä sitten. Ohjelmistorobotiikan niin sanotut avainkehittäjät ovat näytön kaavintaohjelmistot, työnkulun automaatio ja hallintotyökalut ja tekoäly.

Näytön kaavintaohjelma syntyi jo ennen internetiä, jolloin se oli ensimmäinen teknologia, jolla saatiin luotua silta senhetkisten järjestelmien ja vanhojen, yhteensopimattomien järjestelmien välille. Nykyään tätä työkalua käytetään enemmän tietojen poimimiseen verkon esittelykerroksesta. Koska näytön kaavinta perustuu vahvasti HTML-koodiin, nähdään sen hyödyntäminen haastavaksi yritysmaailmassa, jossa tietotekniikan ymmärtäminen voi olla hyvinkin vähäistä.

Työnkulun automaatio alkoi kehittyä jo 1920-luvulla, kun tuotteita alettiin valmistaa liukuhihnalla. Tätä ajatusta alettiin kehittämään 1990-luvulla ja luotiin teknologia, jonka avulla pystyttiin keräämään yksinkertaista tietoa, kuten asiakkaiden yhteystietoja tai tilaustietoja, ja lisäämään näitä tietoja työntekijöiden tietokantoihin. Tämä nopeutti työntekijöiden työskentelyä vähentämällä turhaa tiedonkäsittelyä ja etsimistä. (13.)

Kolmantena ja ehkä tärkeimpänä ohjelmistorobotiikan kehityksen avainteknologiana on tekoäly. Vaikka ohjelmistorobotiikka itsessään ei ole älykästä, ja esimerkiksi sen virheen hallinnasta puuttuu päättelykyky, se on saanut vaikutteita tekoälystä ja kehittynyt sen mukana. Tämä kolmas vaihe on tärkein myös siksi, että sillä on edelleen vahva asema ohjelmistorobotiikan kehityksessä. Ohjelmistorobotiikkaan erikoistuneet yritykset etsivät jatkuvasti keinoja tehdä roboteistaan älykkäämpiä ja näin helpottaa entisestään yritysmaailman työntekijöitä. Mitä itsenäisempi robotti on, sitä suuremman hyödyn se omistajalleen tuo, ja sen rahallinen arvo kasvaa. (13.)

Vaikka ohjelmistorobotiikka nykyään onkin täysin itsenäinen, kehittyvä teknologian osa-alue, se nojaa edelleen vahvasti edellä mainittuihin kolmeen erilaiseen teknologiaan. Se on yhdistelmä, joka luotiin helppokäyttöiseksi työkaluksi yritysmaailmalle. Tärkeimpänä ominaisuutena ohjelmistorobotiikassa voidaan pitää sen loogisuutta. Ymmärtääkseen monimutkaistakin prosessia henkilöllä ei tarvitse olla ohjelmointitaustaa vaan pelkällä hyvällä loogisella päättelykyvyllä pystyy seuraamaan prosessin kulkua helposti. (13.)

On ennustettu, että vuoteen 2025 mennessä automaattisten prosessien, kuten ohjelmistorobotiikan, liikevaihto tulee olemaan lähes 6,7 triljoonaa dollaria. Tämä lukema antaa viitteitä siitä, kuinka suuri merkitys ohjelmistorobotiikalla ja muulla automatisoinnilla on lähitulevaisuudessa. Ohjelmistorobotiikan odotetaan ottavan jalansijaa yhtenä johtavana teknologisenä alustana ja muodostuvan osaksi standardisoitua liiketoiminnan mallia. Esimerkiksi Suomessa pankit ovat ottaneet ohjelmistorobotiikan hyödyntämisen

viime vuosina tehtäväkseen ja pyrkivät antamaan roboteille yhä enemmän tehtäviä joka-päiväisissä tehtävissä. Tällaisia tehtäviä ovat mm. asiakkaiden viestien esikäsittely ja erilaisten lomakkeiden täyttäminen ja tiedon siirtäminen. Ohjelmistorobotiikan tulevaisuudennäkymät ovat siis kirkkaat, etenkin jos tekoälyn yhdistäminen automaatoratkaisuihin saadaan sulavammaksi ja toimivammaksi. (13.)

2.3 Haasteet

Haasteelliseksi ohjelmistorobotiikan tekee sen staattisuus. Muiden ohjelmistojen tavoin robotti ei pysty itse muuttumaan tai oppimaan. Käytännössä tämä tarkoittaa pientenkin muutosten sekoittavan robotin toiminnan kokonaan. Tällaisia muutoksia voivat olla ohjelmistopäivitykset tai esimerkiksi käsiteltävässä tiedostossa tapahtuvat lähes huomattomat muutokset, kuten rivien paikan vaihdot tai nimen muuttaminen. Tämänkaltaisissa tapauksissa robotti on opetettava uudestaan, mikä itsessään lisää työtunteja, joita robotin avulla on yritetty alun perin vähentää.

On myös huomioitava, että viallinen robotti saattaa helposti kaksinkertaistaa ihmisen työmäärän. Esimerkkinä voidaan käyttää tapausta, jossa robotin tehtävänä oli tehdä työvuorotilauksia lisäämällä ilmoitus avoimesta työvuorosta yrityksen sivuille, josta työntekijät pystyivät ilmoittautumaan vuoroon. Päivityksen myötä oli alasvetovalikon järjestys muuttunut siten, että sen alkuun oli lisätty uusi valinta. Koska alasvetovalikkojen tunnistaminen on toisinaan haastavaa roboteille, oli kyseinen robotti opetettu liikkumaan valikossa nuolinäppäimillä alaspäin ja valitsemaan aina kolmas vaihtoehto. Päivityksen myötä kuitenkin oikea valinta olisi ollut vasta neljäs, mutta ilman tarkistusta ei robotti kyennyt tätä havaitsemaan. Tämän seurauksena avautui näytölle väärä taulukko, josta robotti keräsi tiedon ja teki työvuorolistat. Robotin virheen vuoksi yrityksen sivuille ilmestyi työvuorolista, joka sisälsi yli 120 ”turhaa” tilausta. Kun virhe huomattiin, olivat työntekijät jo ehtineet valita näitä ylimääräisiä vuoroja, joita ei todellisuudessa ollut olemassaakaan. Tämä pienen päivityksen myötä tullut virhe johti tilanteeseen, jossa robotin liiketoiminta joutui perumaan vuoroja ja ilmoittamaan virheellisistä tilauksista.

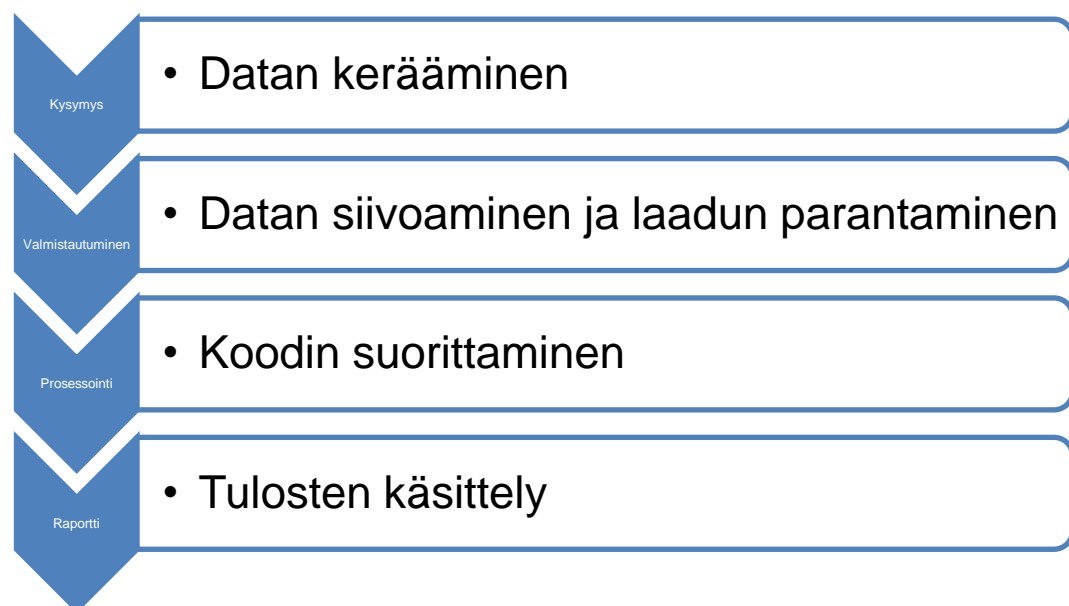
Edellä mainitun kaltaisia tilanteita syntyy robottien kanssa jatkuvasti. Koska moni prosessi suoritetaan öisin, on virheitä liki mahdotonta havaita ajoissa ja pysäyttää robotin

toiminta. Toisaalta, esimerkiksi pankkialalla robotit joutuvat tekemään töitään tavalliseen työaikaan, sillä näihin tehtäviin kuuluu usein asiakkaan kanssa kommunikointi. Koska monet robottien käyttämät kohdejärjestelmät ovat selainpohjaisia, aiheutuu roboteille usein vaikeuksia selvittää hitaista järjestelmistä. Satunnaiset hitaudet sivustoilla kaatavat robotteja, ja ne korjataan usein lisäämällä funktioihin eri tapahtumien välillä olevia odotusaikoja, mikä hidastaa robotin toimintaa myös silloin, kun kohdejärjestelmät toimivat ongelmitta. Ohjelmistopäivityksistä toki usein on mahdollista saada tieto etukäteen, niin että niihin pystytään varautumaan ja ottamaan robotti pois ajosta siksi aikaa, että päivitys saadaan tarkistettua. Aina näin ei kuitenkaan ole, sillä tieto liikkuu helposti hitaasti ja jää sähköpostien syövereihin. Laajemmin katsottaessa on selvää, että robotiikka vähentää työtunteja yrityksessä, kun se korvaa monta työntekijää, mutta se vaatii silti jatkuvaa ylläpitoa usein ulkoistetussa yrityksessä.

3 Koneoppiminen

3.1 Yleisesti

Koneoppiminen perustuu big dataan, tiedon louhimiseen ja tiedon hyödyntämiseen määrätyn tehtävän suorittamisessa. Kuten kuva 5 esittää, koneoppimisen vaiheet voidaan pääpiirteissään jakaa neljään eri vaiheeseen.



Kuva 5. Yksinkertainen malli koneoppimisen askelista (3).

Vaiheista ensimmäinen, datan keruu, on tärkein neljästä vaiheesta. Esimerkkinä voidaan käyttää vaikka kirjaimen A tunnistamista. Jotta algoritmi pystyisi tunnistamaan kirjaimen eri variaatiot ja sen onko sille syötetty kirjain juuri A, sen on kerättävä ensin dataa mahdollisimman monesta tavasta esittää A. Kuten kuvasta 6 voidaan huomata, on tapoja lukematon määrä. Tästä syystä puhutaan koneoppimisesta. Algoritmi oppii jatkuvasti uusia tapoja ja kehittyy näin älykkäämmäksi järjestelmäksi. (3.)



Kuva 6. Esimerkki useasta eri tavasta esittää A-kirjain. Otos on hyvin pieni. (4.)

Kun data on kerätty, siirrytään vaiheeseen 2 eli datan siistimiseen. Kuten Bell teoksessaan *Machine Learning: Hands-On for Developers and Technical Professionals* esittää, data on usein sotkuista, joten sen siivoaminen on usein välttämätöntä. Esimerkkinä teoksessa esitetään tapaus, jossa halutaan kerätä yhteystiedot. Variaatioita tälle kuitenkin on useita, sillä etunimi, sukunimi ja vaikkapa sähköpostiosoite voidaan esittää eri järjestyksissä. Näin ollen on tärkeää saada siivottua hyödytön data, ennen kuin sitä voidaan alkaa käsitellä. (3.)

Seuraavat kaksi vaihetta ovat itse algoritmin suorittaminen ja tulosten esittäminen ja talentaminen. Tässä kohtaa jo kerätty ja siistitty data hyödynnetään viimein haluttuun käyttötarkoitukseen. Usein käytettyjä kieliä koneoppimisessa ovat:

- R
- Python
- Matlab
- Scala
- Clojure
- Ruby
- Java.

Kaikki nämä kielet sisältävät koneoppimiseen tarkoitettuja kirjastoja, jotka auttavat algoritmien rakentamisessa. (3.)

Koneoppimista käytetään laajasti mm. ohjelmistoissa, markkinoinnissa (sekä tavallisessa mainonnassa että e-mainonnassa), robotiikassa, pelien analysoinnissa ja lääketieteessä. Ohjelmistoista hyvänä esimerkkinä voidaan pitää roskapostintunnistusohjelmia. Epäilyttävän sähköpostin saapuessa ohjelmisto kysyy käyttäjältä lupaa siirtää posti roskapostiin. Jos käyttäjä hyväksyy pyynnön, ohjelmisto oppii pitämään tämänkaltaisia sähköposteja haitallisina ja siirtää ne suoraan roskapostiin. (3.)

Markkinoinnissa koneoppiminen on hiukan uudempi ilmiö, mutta nykyään lukuisat suuret yritykset seuraavat asiakkaidensa ostoksia ja koneoppimisen avulla rakentavat räätälöityjä tarjouksia ja mainoksia. Tämä on kuitenkin osoittautunut välillä haitalliseksi, sillä moni asiakas tuntee olonsa uhatuksi, kun kone tietää heistä liikaa. Siksi jotkut yritykset ovat ottaneet tavaksi mainostaa asiakkaalle suunnattuja tuotteita, mutta sekoittaa joukkoon myös vääränlaisia mainoksia. Niin ne luovat asiakkaalle vaikutelman, ettei häntä seurata tai tarkkailla. (3.)

Robotiikassa koneoppimista käytetään lähinnä liikkeiden tallentamiseen ja toistamiseen. Esimerkiksi tehdasroboteissa käytetään koneoppimista ja luodaan näin hiukan monimutkaisempia laitteita. (3.)

Peleissä koneoppiminen on yleisesti käytössä, kun luodaan botteja. Monet pelit sisältävät itse pelaajien lisäksi tietokoneella luotuja hahmoja. Esimerkiksi monelle tutussa Counter strike -pelissä voi toisten pelaajien sijasta pelata botteja vastaan. Silloin pelaaja pelaa tietokonetta vastaan joukkueessa, jossa on hänen lisäksi muita, tietokoneella ohjelmoituja hahmoja. Koneoppiminen ilmenee hahmojen käyttäytymisessä. Hahmot oppivat kierrosten aikana ihmispelaajan taktiikan ja alkavat toistaa sitä samassa joukkueessa olevien pelaajien kohdalla ja vastaavasti hyödyntää tietoa vastakkaisella puolella

olevien pelaajien kanssa. Näin luodaan mahdollisimman haastava peli, jossa itse pelaajan pitää muuttaa käyttäytymistään pärjätäkseen oppivaa ohjelmistoa vastaan. (3.)

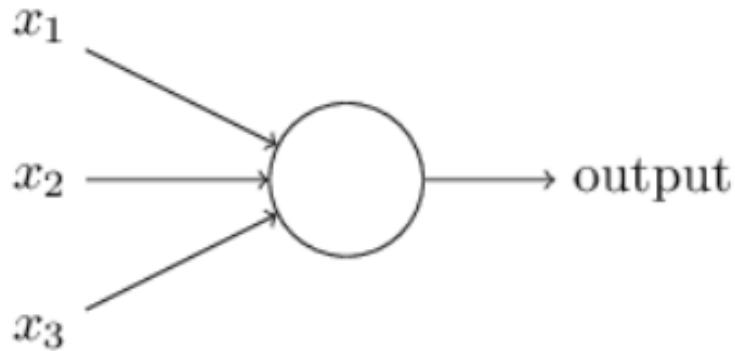
Lääketieteessä tärkein esimerkki on varmasti IBM Watson, älykkäin ihmisen rakentama tekoäly. Watson kykenee lukuisiin erilaisiin tehtäviin, mutta hyödyllisin niistä on sen kyky diagnosoida ihmistä. Tekoälylle on syötetty lukematon määrä dataa erilaisista sairauksista ja niiden oireista, ja se oppii jatkuvasti uusien diagnoosien myötä tunnistamaan ennestään sille tuntemattomia asioita. Watsonissa koneoppiminen on viritetty äärimmilleen ja luotu kone, jolla on enemmän tietoa sairauksista kuin yksikään lääkäri pystyisi muistamaan. Lääketieteellisen osaamisensa lisäksi Watson pystyy muun muassa profiloimaan ihmisiä heidän työhakemuksiensa perusteella. (5.)

3.2 Neuroverkot

Palataan esimerkkiin A-kirjaimen tunnistamisesta. Kuten aikaisemmin todettiin, on kyseisen merkin tunnistaminen tietokoneelle haastavaa johtuen lukemattomista eri tavoista ilmaista kirjainta. Ihminen taas kykenee tunnistamaan erilaiset variaatiot helposti. Ihmisen aivoissa on tunnistukseen erikoistunut sarja: V1, V2, V3, V4 ja V5. Tämän sarjan osista V1, joka sisältää yli 140 miljoona pientä neuronia, on erikoistunut tunnistamaan tämänkaltaisia tapauksia (6). Warren McCulloch ja Walter Pitts osasivat jo vuonna 1943 hyödyntää ihmisaivojen fysiologista rakennetta matemaattisten ongelmien ratkaisemisessa ja esittämisessä (7). Tietokoneiden käyttämät neuroverkot voidaan jakaa kahteen ala-lajiin: perceptroneihin ja sigmoidisiin neuroverkostoihin (6).

Perceptronverkko

Frank Rosenblatt kehitti vuosina 1950-1970 McCullochin ja Pittsin ajatusten innoittamana tavan hyödyntää neuroneita laskennassa. Tästä syntyivät nykyään jo harvemmin käytetyt perceptronin neuronit (kuva 7), joiden teho perustuu tavan yksinkertaisuuteen.



Kuva 7. Perceptronin neuroverkoston malli (6).

Kuten kuvassa 7 voidaan havaita, perceptron perustuu laskentaan, jossa useasta binäärisestä sisääntulosta saadaan tulokseksi yksi binäärinen ulostuloarvo. Kuvan 7 mallissa sisääntulo on merkitty millä tahansa reaaliluvulla x . Matemaattisesti ulostuloarvo pystytään esittämään lukujen x summalla:

$$\sum_{i=0}^n w_i x_i$$

Kuva 8. Perceptronin perusyhtälö.

$$f(x) = \begin{cases} 0, & \sum_{i=0}^n w_i x_i - b \leq 0 \\ 1, & \sum_{i=0}^n w_i x_i - b > 0 \end{cases}$$

Kuva 9. Perceptronin yhtälö. (6).

Kuvien 8 ja 9 kaavoissa kirjain n osoittaa sisään syötettävien tietojen lukumäärää. Jokaiselle sisääntulolle asetetaan vektorimuodossa oleva arvo w , joka kuvaa niin kutsuttua painoarvoa. Tämä tarkoittaa käytännössä sitä, kuinka suuri merkitys sisään syötetyllä luvulla on. Kirjaimella x esitetään sisääntuloarvoa. Alemmassa kaavassa lisänä on kirjain b , eli bias. Bias on valmiiksi annettu yläraja, jonka mukaan tietokone laskee arvoksi 1 tai 0. Perceptronille tyypillistä on, että ulostuloarvo on aina binäärimuodossa. (6.)

Kuten kaavoista voidaan havaita, perceptronin variaatiot toimivat samalla tavalla kuin tietokonearkkitehtuurissa käytetyt logiikkaportit AND, OR, NAND ym. Esimerkkinä voidaan ottaa yleisesti käytetty AND-portti. AND syöttää ulostuloarvon 1, jos ja vain jos sen kaikki sisääntulot ovat arvoltaan 1, muussa tapauksessa ulostulo on 0. Samalla tavalla perceptronin logiikkaa ketjuttamalla saadaan rakennettua neuronpuu, joka syöttää ulos yhden ulostulon ennalta määrättyllä logiikalla.

Ongelmana perceptron verkostojen tavassa pidetään sen herkkyyttä muutoksille. Pienikin muutos sisääntuloissa tai painoarvoissa voi vaikuttaa lopputulokseen radikaalisti. Tämä ongelma korjattiin kehittämällä uusi tapa käsitellä neuroverkkoja. Tätä tapaa kutsutaan yleisesti sigmoidiseksi neuroniksi.

Sigmoidiset neuroverkostot

Sigmoidiset neuroverkostot voidaan kuvata samanlaisella mallilla, kuin perceptronin neuronit on kuvattu kuvassa 7, eroavaisuudet näkyvät kuitenkin sisääntuloarvoissa. Toisin kuin perceptronissa, sigmoidisissa neuroneissa voidaan sisääntuloarvoiksi syöttää mitä tahansa yhden ja nollan välistä. Tämä ominaisuus tekee sigmoidisten neuroverkostojen laskemisesta perceptronia haastavampaa. Algoritmi perustuu Sigmoidin funktioon, joka määritellään seuraavasti:

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}}$$

Funktion avulla voidaan johtaa sigmoidisille neuroneille kuvan 10 mukainen kaava:

$$\frac{1}{1 + e^{(-\sum w_i x_i - b)}}$$

Kuva 10. Sigmoidin yhtälö (6).

Kuten kuvan 10 kaavasta voidaan havaita, pohjautuvat sigmoidiset neuronit perceptronin tapaan laskea output arvo summaamalla input valmiiksi määriteltyyn painoarvoon. Kuten perceptronissa, sigmoidin yhtälössä x = sisääntulon arvo, b = bias, eli etukäteen asetettu raja-arvo ja w = inputin painoarvo. Lisäksi esiintyy sigmoidin yhtälössä neperin luku e . Eroavaisuus kuitenkin liittyy mahdollisuuteen käyttää muitakin kuin binääriarvoja. Sig-

moidin funktio on lineaarinen, nouseva funktio. Tätä hyödyntämällä pystytään mallintamaan algoritmi, jossa pienet arvonmuutokset sisääntuloarvoissa saavat aikaan pieniä arvomutoksia ulostuloarvoissa, tämän mahdollistaa funktion derivoituvuus.

3.3 Koneoppimisesimerkki

Päästäkseen syvemmälle neuroverkostoihin ja ymmärtääkseen niiden toiminnan on hyvä tarkastella konkreettista esimerkkiä. Tässä luvussa esitellään yksinkertainen Pythonilla kirjoitettu koneoppimista hyödyntävä esimerkkitapaus. Kyseessä on kuvan tunnistusta tekevä neuroverkosto, ja testidata on ladattu internetistä. Koska itse työ käsittelee pankkialalle suunnatun tekoälyn suunnittelua, on vaikea saada oikeanlaista testidataa, jotta pystyttäisiin tekemään tarpeeksi osuva esimerkki. Idea kuitenkin kuvan tunnistuksessa ja asiakastiedon käsittelyssä sama. Molemmat tavat hyödyntävät matriiseja, joihin tieto on muutettu numeeriseen muotoon, jotta algoritmi pystyy laskemaan todennäköisyyksiä ja tekemään päätöksiä. Esimerkissä on käytetty Anaconda-ohjelmaa ja Theano-nimistä, Python-kielelle tehtyä ohjelmaa. Theanon ladattiin Anaconda Prompt-ikkunassa, joka on verrattavissa windowsin command prompt-ikkunaan, käyttämällä komentoja *"conda install -c conda-forge theano"* ja *"conda install -c conda-forge/label/broken theano"*.

Oikeiden kirjastojen käyttäminen koneoppimisessa on tärkeää. Pythonille on luotu lukemattomia erilaisia matematiikkaan ja tiedonkäsittelyyn liittyviä kirjastoja. Tässä esimerkissä on käytetty esimerkikoodin 1 mukaisia kirjastoja.

```
import gzip
import os
import numpy as np
import theano
import theano.tensor as T
import random as rd
from theano.tensor.signal import pool
from theano.tensor.nnet import conv2d
import six.moves.cPickle as pickle
```

Esimerkkikoodi 1. Koodissa käytetyt kirjastot.

Näistä kirjastoista erityisesti Numpy on laajasti käytetty koneoppimisessa. Numpy sisältää muun muassa tehokkaan objektin, jonka avulla pystytään käsittelemään N-ulottuvuudellista listaa. Lisäksi kirjastossa on useita funktioita, jotka käsittelevät lineaarista algebraa ja Fourierin muunnosta. (12.)

Esimerkki alkaa datan lataamisella. Kuten edellä on mainittu, käytetään tässä ohjelma pätkässä internetistä saatua esimerkki materiaalia, oikean datan puuttumisen johdosta. Tiedon lataaminen tapahtuu esimerkikoodin 2 esittämällä tavalla.

```
def load_data(dataset, train_size=6000):
    data_dir, data_file = os.path.split(dataset)
    if data_dir == "" and not os.path.isfile(dataset):
        new_path = os.path.join(os.path.split(__file__)[0], dataset)
        if os.path.isfile(new_path) or data_file == 'mnist.pkl.gz':
            dataset = new_path

    if (not os.path.isfile(dataset)) and data_file == 'mnist.pkl.gz':
        from six.moves import urllib
        origin = (
            'http://www.iro.umontreal.ca/~lisa/deep/data/mnist/mnist.pkl.gz'
        )
        print('Downloading data from %s' % origin)
        urllib.request.urlretrieve(origin, dataset)
```

Esimerkkikoodi 2. Tiedon lataaminen.

Suurin osa koodista on lähtötilanteen määrittämistä. Kuten aina ohjelmoissa, kaikki tiedot tulee ensin alustaa ja alku arvot on asetettava oikein. Esimerkkikoodi 3 sisältää konvoluutiokerroksen määrittämisen.

```
def convLayer(rng, data, image_size, filter_size, pool_size=None, activation=T.tanh):
    conv_output = conv2d(
        input=data,
        filters=W,
        border_mode='half',
        filter_shape=filter_size,
        input_shape=image_size)

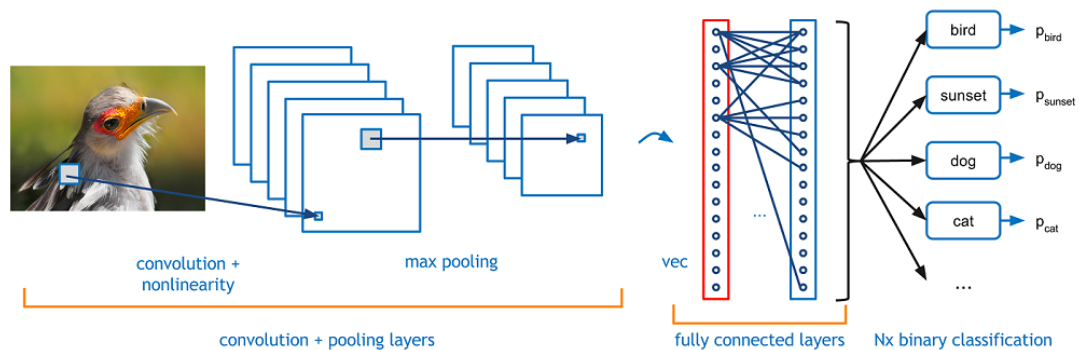
    bias_shuffled = bias.dimshuffle('x', 0, 'x', 'x')
    output = activation(conv_output + bias_shuffled)
    if pool_size:
        output = pool.pool_2d(input=output, ws=pool_size, ignore_border=True)

    return output, [W, bias]

def fullyConnected(data, dim_in, dim_out):
```

Esimerkkikoodi 3. Konvoluutiokerroksen määrittäminen.

Kuvassa 11 esitetään konvoluutioverkon toimintaa käytännössä. Siinä kuvaa tarkastellaan pikseli kerrallaan ja yritetään luoda vektorimuotoinen kokonaiskuva. Mitä pienemmissä osissa tunnistus tehdään, sitä todennäköisempää on, että kuvantunnistus tehdään virheettömästi. Jokaisesta pikselin sisältämästä tiedosta luodaan vektorimuotoinen arvo, joista yhdessä muodostetaan matriisi, jota tietokone käsittelee. Täysin yhdistetty neuroverkosto tarkoittaa, että jokainen neuroni on yhteydessä toisiinsa. (14.)



Kuva 11. Konvoluutioverkon toiminta.

Esimerkkikoodissa 4 on esitetty vektorien ja biasin alustaminen. Varsinkin vektorin alustaminen on tärkeää kohinan vähentämiseksi.

```

W = theano.shared(
    value=np.asarray(
        rng.normal(loc=0, scale=0.1, size=(dim_in, dim_out))),
    name='W',
    borrow=True)

bias = theano.shared(
    value=np.zeros((dim_out,)),
    name='b',
    borrow=True)

classes = T.nnet.softmax(T.dot(data, W) + bias)
y_pred = T.argmax(classes, axis=1)
return y_pred, classes, [W, bias]

def train_test_conv_net(learning_rate, num_epochs,
                        num_filters, batch_size, activation,
                        train_set_x, valid_set_x, test_set_x,
                        lambda_cof, valid_set_y, test_set_y,
                        filter_size=(5,5), pool_size=(2,2)):
    params = fc_layer_params + layer1_params

```

Esimerkkikoodi 4. Vektorien ja biasin alustaminen.

Koneoppimisen perustana on sen kyky oppia valmiiksi annetusta tiedosta ja soveltaa sitä halutulla tavalla. Esimerkkikoodi 5 näyttää, miten opettaminen tapahtuu.

```

updates = sgd(cost, params, learning_rate)

train_model = theano.function([index],
                               cost,
                               updates=updates,
                               givens={
                                   x: train_set_x[index * batch_size: (index + 1) * batch_size],
                                   y: train_set_y[index * batch_size: (index + 1) * batch_size]})

epoch = 0
cost_arr = np.array([])
valid_score_arr = np.array([])
valid_score_arr = np.append(valid_score_arr, 1)

print('***** Training model *****')
while (epoch < num_epochs):
    epoch = epoch + 1
    print("Training in epoch: %d / %d\n" % (epoch, num_epochs),
          end='\r')
    for index in range(n_train_batches):
        # cost of each minibatch
        cost_ij = train_model(index)
        cost_arr = np.append(cost_arr, cost_ij)

    # Computing loss on each validation mini-batch after each epoch
    valid_losses = [valid_model(i) for i in range(n_valid_batches)]
    valid_score_arr = np.append(
        valid_score_arr,
        np.mean(valid_losses))

```

Esimerkkikoodi 5. Algoritmin opettaminen.

Datan lataaminen Theanosta, suodattaminen ja muut ohjelmalle tärkeiden parametrien lataaminen tapahtui esimerkkikoodissa 6. Kaikkein alimmat rivit tekevät varsinaisen neuroverkoston prosessoinnin.

```

datasets = load_data('mnist.pkl.gz')
train_set_x, train_set_y = datasets[0]
valid_set_x, valid_set_y = datasets[1]
test_set_x, test_set_y = datasets[2]
learning_rate = 0.01
activation = T.tanh
filter_size = [5, 5]
pool_size = [2, 2]
num_epochs = 5
num_filters = [20, 20]
batch_size = 200

train_test_conv_net(learning_rate, num_epochs,
                    num_filters, batch_size, activation,
                    train_set_x, valid_set_x, valid_set_x,
                    valid_set_y, valid_set_y, test_set_y,
                    filter_size=filter_size,
                    pool_size=pool_size)

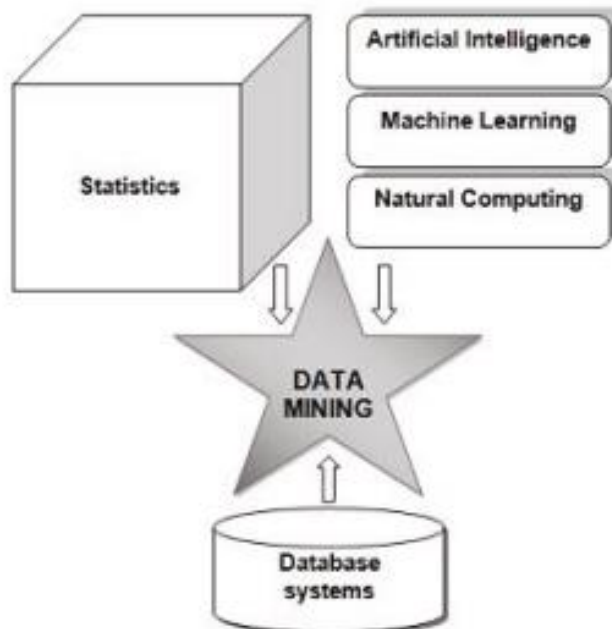
```

Esimerkkikoodi 6. Suodattaminen ja neuroverkoston käsitteleminen.

4 Tiedonlouhinta

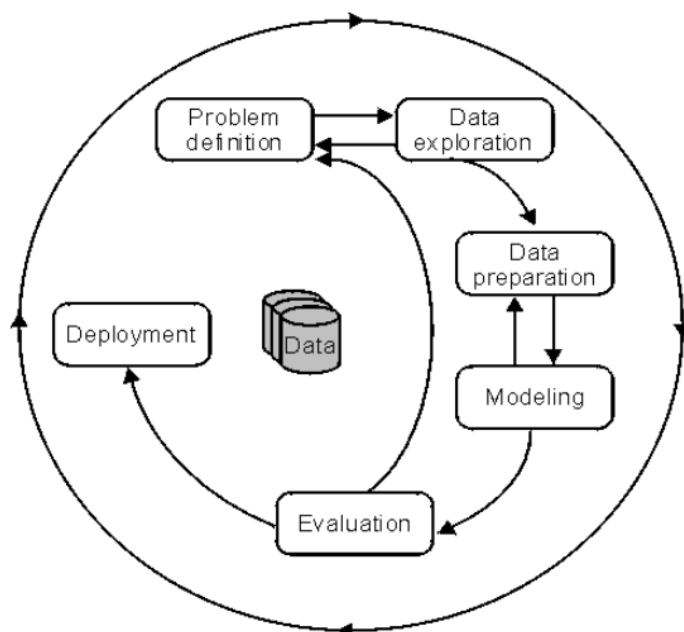
On tärkeää ymmärtää, että koneoppiminen ja neuroverkot ovat vain pieni osa suurempaa kokonaisuutta. Synapsit ja niiden sisältämät algoritmit olisivat turhia ilman dataa, jota käsitellä. Tässä kohtaa kuvaan astuu tiedonlouhinta. Tiedonlouhinnan kehittymisen katsotaan alkaneen jo vuonna 1763, kun Bayesin teoreema julkaistiin. Tätä voidaan kutsua tilastotieteen ja tiedonlouhinnan synnyinvuodeksi, sillä teoreema mahdollistaa monimutkaistenkin realiteettien ymmärtämisen todennäköisyyksien avulla. Vuonna 1805 Adrien-Marie Legendre ja Carl Friedrich Gauss kehittivät regressioanalyysin, joka on edelleen yksi tärkeimmistä työkaluista tiedonlouhinnassa. Seuraavat urauurtavat askeleet tiedonlouhinnan kehityksessä otettiin vasta 1900-luvun puolella, neuroverkostojen syntyessä ja myöhemmin tietokoneiden kehittyessä. Tiedonlouhinnan merkitys on kasvanut räjähdysmäisesti viimeisten vuosikymmenien aikana, ja nykypäivänä voidaan sanoa datan olevan rahaakin arvokkaampaa valuuttaa. (9.)

Ennen kuin syvennyttään itse tiedonlouhintaan prosessina, tarkastellaan miten koneoppiminen, tekoäly, tilastotiede ja tiedonlouhinta liittyvät yhteen. Tarkastelemalla kuvaa 12 voidaan havaita jokaisen palasen olevan yhteydessä toisiinsa. Koneoppimista käytetään yhtenä tiedonlouhinnan rakennuspalikkana muiden osien rinnalla.



Kuva 12. Tiedonlouhinnan osa-alueet (9).

Tiedonlouhinta voidaan siis jakaa useaan osaan: yksittäinen prosessi sisältää tilastotietoa, tietokannan ja mahdollisesti tekoälyn, koneoppimista tai luonnollista laskentaa kuten kuvasta 13 voidaan havaita. Näistä koneoppimista esimerkiksi käytetään tiedon käsittelyyn ja mallintamiseen. Tietokanta taas sisältää itsessään kaiken datan, jota tiedonlouhinnalla haetaan. (10.)



Kuva 13. Tiedonlouhintaprosessi (11).

Facebookilla oli pelkästään Yhdysvalloissa vuonna 2017 yli 12,1 miljoona käyttäjää (15). Määrä on huima, ja sivustoa voidaan edelleen suosionsa laskusta huolimatta pitää yhtenä laajimmalle levinneistä ja käytetyimmistä sosiaalisista medioista. Yksi suosion laskuun liittyvistä syistä on Facebookin yhä röyhkeämmäksi menevä kohdennettu mainos ja yksityisten tietojen urkkiminen sekä kyseisen datan eteenpäin myyminen. Kaikki tämä tapahtuu tiedonlouhinnan avulla.

Ongelman määrittäminen

Ongelman määrittämiseksi kutsutaan tiedonlouhinnan ensimmäistä ja tärkeintä vaihetta. Tätä vaihetta voidaan kutsua myös datan visualisoinniksi, sillä siinä maalataan tarkka kuva halutusta tiedosta, jotta se pystyttäisiin erottamaan lukemattomista biteistä dataa. Suurista tietokannoista on mahdotonta etsiä dataa, jota ei ole määritelty tarpeeksi kattavasti heti alussa. Jos otetaan esimerkiksi edellä mainittu Facebook, tämä vaihe tarkoittaa

kohteen valintaa. Liittyessään Facebookiin käyttäjä antaa sivustolle luvan kerätä ja käyttää sille annettuja tietoja mielensä mukaisesti. Tilin luomisen jälkeen Facebook voi luvallisesti jakaa ja myydä tietoa, jonka se käyttäjästä kerää. Käytännössä se tarkoittaa, että tilin luomisen yhteydessä käyttäjän yksityisyys katoaa.

Datan etsiminen

Datan etsiminen nimensä mukaisesti sisältää itse "louhinnan". Siinä hyödynnetään eri tietokantoja visualisoidun datan löytämiseksi. Facebookissa dataa etsitään lukemattomilla eri tavoilla ja kanavilla. Paras esimerkki on vilkaista Facebook Messengerin käyttöoikeuksia. Muutama päivitys sitten ilmestyi uusi ominaisuus, joka rajoitti Facebookin viestien lukuoikeutta niin, että puhelimella ei viestejä kykene näkemään tai lähettämään ilman Messenger-ohjelmaa. Käytännössä tämä tarkoittaa, että käyttäjän yrittäessä painaa viestipainiketta Facebook ohjaa suoraan Messengerin lataussivuille. Ladatakseen ohjelman käyttäjän on kuitenkin hyväksyttävä mm kuvan 14 käyttöehdot.

Kalenteri

- read calendar events plus confidential information

Yhteystiedot

- find accounts on the device
- read your contacts
- modify your contacts

Tekstiviesti

- read your text messages (SMS or MMS)
- receive text messages (MMS)
- receive text messages (SMS)
- send SMS messages
- edit your text messages (SMS or MMS)

Puhelin

- directly call phone numbers
- reroute outgoing calls
- read call log
- read phone status and identity
- write call log

Kamera

- take pictures and videos

Mikrofoni

- record audio

Muu

- download files without notification
- receive data from Internet
- view network connections
- create accounts and set passwords
- change network connectivity
- full network access

Kuva 14. Messengerin käyttöehdot (16).

Toisin sanoen, ladatessaan ohjelman puhelimelleen, käyttäjä luovuttaa viimeisenkin yksityisyyden ja antaa Facebookille oikeudet puhelimeensa. Tähän Facebook käyttää apunaan tiedonlouhintaa. Se nauhoittaa mm. käyttäjän puhetta mikrofonin kautta ja poimii yksittäisiä sanoja etsien usein keskusteluissa toistuvia termejä ja kerää yleisimpiä hakusanoja sekä verkkosivuja, joilla käyttäjä vieraillee. Moni on oppinut tämän kantapään kautta. Lukemattomat käyttäjät kertovat tilanteista, joissa he ovat keskustelleet aiheesta x melkein pä ohimennen, Messenger on poiminut pääsanat ja erinäiset sovellukset ovat alkaneet mainostaa aihealueeseen liittyviä tuotteita. (17.)

Datan valmistaminen

Kolmannessa vaiheessa data käsitellään valmiiksi mallintamiseen. Käsittelyn tuloksena saadaan yhteen koottua dataa, jota on siivottu poistamalla mahdollista ”meteliä”, jota löydettyssä datassa usein ilmenee. Tämä valmistelu on hyvin kriittistä seuraavaa vaihetta ajatellen, sillä sotkuisen datan käsittely aiheuttaa virheitä ja vaikeuttaa mallintamista.

Mallintaminen

Mallintamisessa kuvaan astuu koneoppiminen tai mahdollinen tekoäly. Se sisältää algoritmeja, joiden tehtävänä on käsitellä tietoa ja tehdä päätöksiä tai vaihtoehtoisesti opettaa tekoälyä. Erilaisia koneoppimisen malleja on neuroverkkojen lisäksi muun muassa päätöspuu.

Arviointi ja käyttöönotto

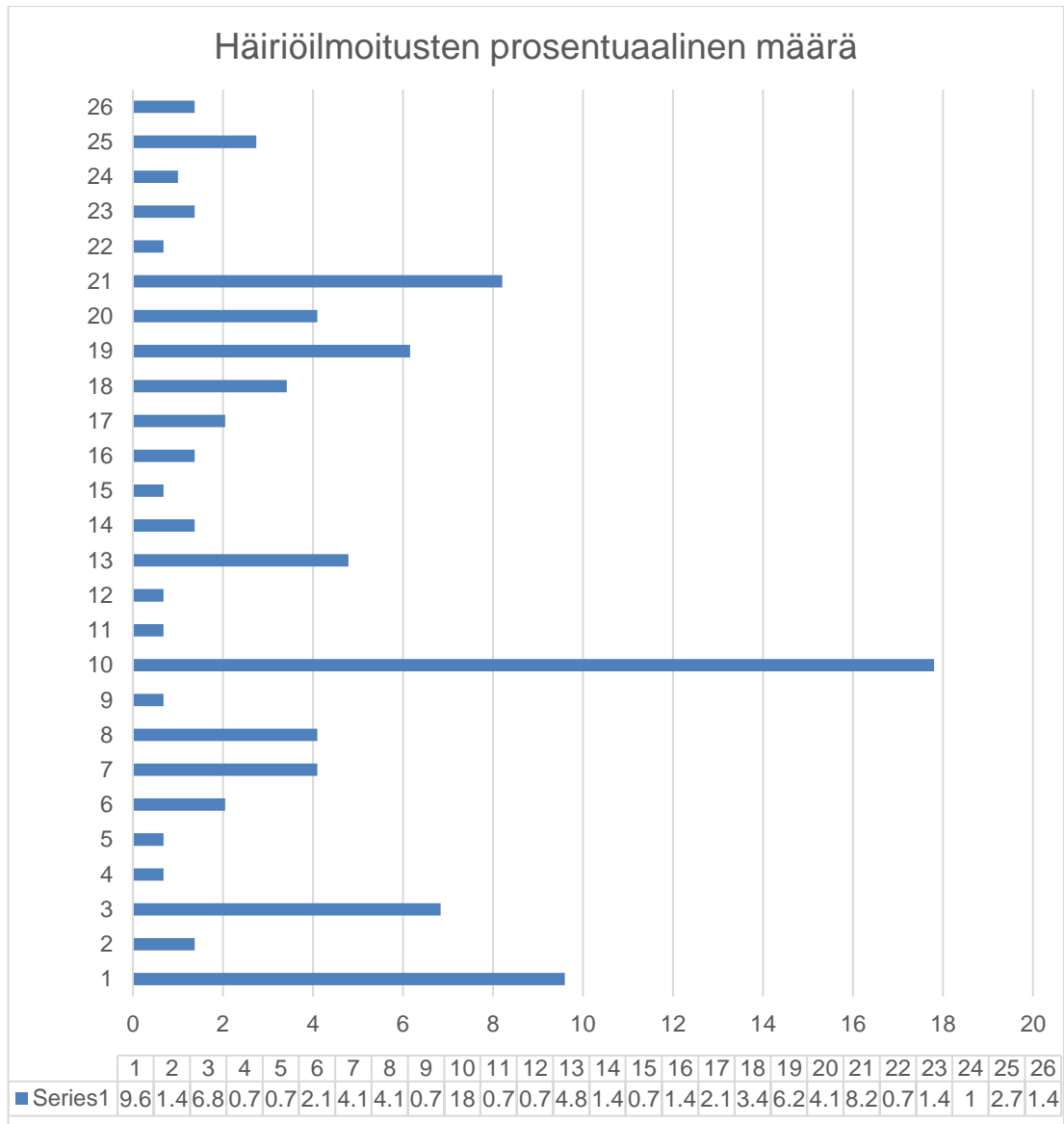
Viides ja kuudes vaihe tapahtuu tiedonkäsittelyyn erikoistuneen työntekijän toimesta. Siinä saatu data arvioidaan ja päätetään, täyttääkö se vaaditut kriteerit. Jos data on käytökelpoista, se voidaan ottaa hyötykäyttöön. Facebook esimerkissä tämä vaihe tarkoittaa

tiedon luovuttamista eteenpäin. Kun kerätty data annetaan yritysten haltuun, ne kykenevät kohdentamaan markkinoinnin juuri käyttäjälle sopivaksi.

Insinööriyön yritykselle toteutetussa ratkaisussa tiedonlouhinta esiintyy ratkaisevassa roolissa asiakkaiden profiloinnissa. Pankki-alan yritykset omistavat useita eri kohdejärjestelmiä, joihin asiakkaiden tietoja käsitellään, varsinkin jos asiakkaalla on keskitetty asiakkuus. Tiedonlouhinnan avulla kyettäisiin keräämään kaikki tarvittava data yhteen koneoppimista hyödyntävien algoritmien käsiteltäväksi. Mitä enemmän tietoa löytyy, sitä varmemmaksi muuttuu tiedonkäsittelyvaihe.

5 Markkinointiaineisto yritykselle

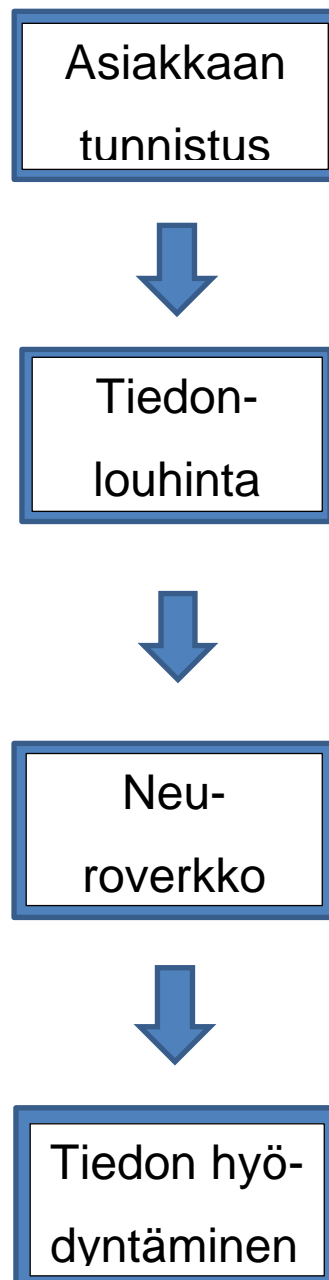
Insinööri työhön kuului oleellisena osana markkinointi raportin (liite 1) kirjoittaminen ja sen pohjalta tehdyn diaesityksen (liite 2) tekeminen. Nämä dokumentit kohdennettiin potentiaaliselle asiakkaalle, ja siksi ne sisältävät vain vähän teknologista tietoa ja keskittyvät enemmän tekoälyn markkinointi hyötyihin. Markkinointi raportin kohdalla oli tärkeintä esittää ongelma, johon tekoäly toisi ratkaisun. Tämä ongelma oli ohjelmistorobotiikan epävarmuus ja kustannukset yritykselle. Työhön lisätty graafi sisälsi robottien tuottamia häiriöilmoituksia ylläpidolle. Näitä prosentteina esitettyjä lukuja avataan raportissa keskittyen niiden ylimääräisiin kustannuksiin. Kuvassa 14 havainnollistetaan näitä prosentuaalisia lukuja häiriöilmoituksissa prosessia kohden.



Kuva 15. Robottien aiheuttamat häiriöt rajatulla kolmen kuukauden jaksolla.

Graafista on poistettu robottien nimet ja korvattu ne juoksevalla numeroinnilla arkaluontoisen tiedon vuoksi. Kuvaajasta pystytään kuitenkin helposti huomaamaan ongelmalliset tapaukset. Tietysti on otettava huomioon, etteivät robotit ole keskenään täysin verrattavissa, sillä häiriöiden määrään vaikuttavat vahvasti aikataulut ja resurssit. Tämä tarkoittaa, että osa roboteista ajaa päivittäin ja mahdollisesti useamman kerran päivän aikana, ja lisäksi ne saattavat prosessoida työtehtäviään samanaikaisesti usealla koneella, jolloin virheitä syntyy huomattavasti useammin kuin robotilla joka käyttää vain yhtä konetta. Kuvaaja kuitenkin auttaa selkeyttämään tilannetta visuaalisesti ja näyttämään hyviä kehityskohteita, kuten työn oli tarkoituskin tehdä.

Ohjelmistorobotiikan avulla rakennetut ratkaisut korvattaisiin hyödyntämällä tiedonlouhintaa ja koneoppimista. Kuva 16 esittää prosessin etenemistä.



Kuva 16. Markkinaideaa kuvaava kaavio.

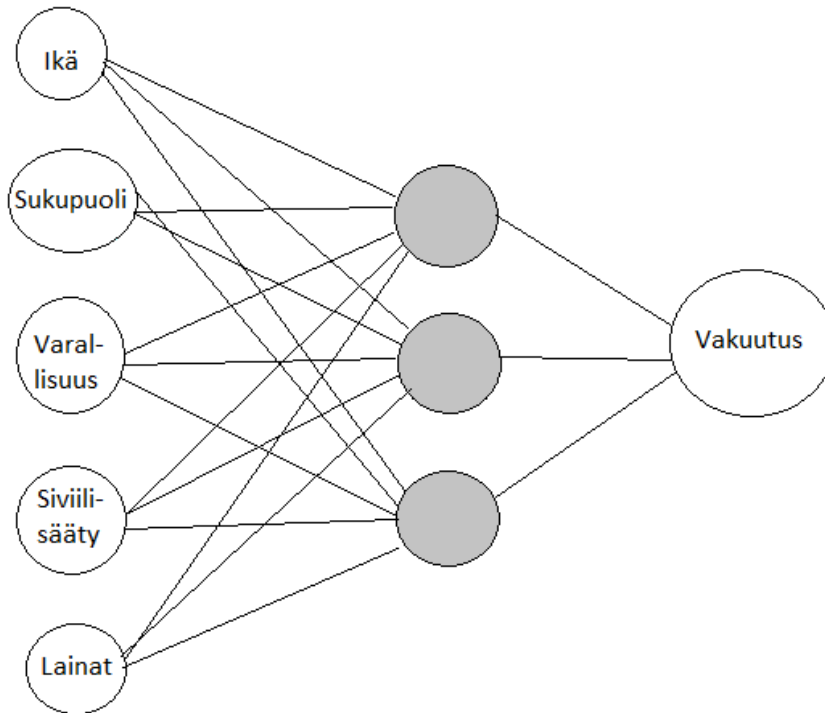
Ensimmäinen kohta eli asiakkaan tunnistaminen on tärkeä yrityksen näkökulmasta. On hyvä tunnistaa potentiaalinen asiakas ennen resurssien käyttämistä. Asiakas tunnustetaan ja paikannetaan seuraamalla aktiivisuutta. Paikannus voi tapahtua monella eri tavalla. Esimerkiksi ohjelmistorobotiikka on edelleen varsin tehokas tapa seurata ihmisten

liikkeitä ja ilmoittaa yrityksille halutusta aktiivisuudesta. Tämä tarkoittaa, että esimerkiksi pankkialalla juuri lainan ottaneet asiakkaat ovat ensisijaisia louhintakohteita ja potentiaalisia uusia asiakkaita, sillä he ovat todennäköisimmin valitsemassa vakuutuksia tai ovat kiinnostuneita uusista vakuutuksista. Tämä tarkoittaa, että ottaessaan lainaa autoon, moni on valmis myös valitsemaan hiukan kalliimman autovakuutuksen, varsinkin jos yritys tarjoaa tuotteen tarjouksena tai tekee oston muuten helpoksi.

Tiedonlouhintaa tapahtuu jatkuvasti, vaikka ihminen ohjaisikin tuloksia enemmän paikannetun asiakkaan suuntaan. Tiedonlouhinnassa etsitään kaikki mahdollinen tieto asiakkaasta ja data valmistellaan sopivaksi myöhempää käyttöä varten. Tässä kohtaa on huomioitava tekoälyn laskentatehokkuus ohjelmistorobotiikkaan verrattuna. Kuten jo aikaisemmin on mainittu, asiakkaiden tiedot ovat hajautettuna usein lukemattomiin eri järjestelmiin ja tietokantoihin. Robotin kyky lukea näitä on hyvä, mutta monet seikat hidastavat tuotannossa prosessoivaa robottia, jonka esimerkiksi selainpohjaisia järjestelmiä lukiessa on avattava selain näkyviin päästäkseen käsiksi sinne piilotettuun dataan. Tiedonlouhinnalla taas päästään suoraan kiinni tietokantaan, ilman erillisiä, ihmissilmälle näkyviä vaiheita.

Kun on kerätty kaikki mahdollinen tieto ja muutettu se käsiteltäväksi oikeaan matriisimuotoon pystytään tiedot antamaan neuroverkoston käsiteltäväksi. Toisissaan kiinni olevat neuronit käsittelevät tiedon, kuten aivot käsittelevät dataa, ja tekevät päätöksen aikaisemmin opitun tiedon mukaisesti.

Kuvasta 16 voidaan nähdä yksinkertaistettu malli verkoston toiminnasta. Siinä alkuarvoja ovat ikä, sukupuoli, varallisuus, siviilisääty ja mahdolliset lainat. Nämä alkuarvot esiintyvät matriiseina (verrattavissa koneoppimista käsittelevän luvun 3 esimerkkikoodin matriiseihin), joista neuroverkosto laskee todennäköisyyden eri vakuutuksille. Esimerkkinä on tilanne, jossa nuori on ottanut itselleen opintolainaa tavallista enemmän ja nostanut sen kevään lopussa. On todennäköistä, että nuori harkitsee vaikkapa ulkomaanmatkaa, jolloin kannattaa tarjota halpa matkavakuutus.



Kuva 17. Esimerkki neuroverkoston rakenteesta.

Kun päätös on tehty, tekoäly antaa ulos lopputuloksen. Tästä eteenpäin yrityksen omat työntekijät voivat tarjota asiakkaalle lopputulosta olettaen tuotteen menevän kaupaksi paremmin kuin ihmisen tekemän arvauksen perusteella. Kaiken kaikkiaan tämän tyyppinen ratkaisu parantaisi yrityksen myyntiä lisäämällä kauppvoja ja säästämällä resursseja turhalta ja hedelmättömältä työltä. Vaihtoehtoisesti myös ohjelmistorobotiikalla on potentiaalia jatkaa tiedon eteenpäin viemistä, sillä tämänkaltainen työ ei vaadi robotilta vaikeaa päättelyä tai haastavia kohdejärjestelmiä.

6 Yhteenveto

Insinööriyössä esiteltiin mahdollinen vaihtoehto ohjelmistorobotiikalle pankkialan yrityksessä edesauttamaan sen myyntiä ja tuomaan yritys lähemmäs nykyaikaa. Tekoäly on vallannut suuria markkina-alueita ympäri maailman, ja sen ote yritysmaailmassa tiukentuu. Pysyäkseen teknologian tasalla ja tarjotakseen asiakkailleen parhaan mahdollisen palvelun on yritysten yritettävä tehokkaasti etsiä uusia ratkaisuja ja mahdollisuuksia.

Tiedonlouhintaa ja neuroverkkoja hyödyntävä tekoäly kykenisi laskemaan potentiaalisia lisämyyntejä asiakkaiden keskuudesta etsimällä kaiken mahdollisen tiedon heistä ja luomalla käyttäytymismalleja. Tämänkaltaisella teknologialla virhemarginaali pienentyisi ja myyntiä saataisiin lisättyä. Tällä hetkellä kyseinen yritys hyödyntää ohjelmistorobotiikkaa tiedon keräämiseen ja käsittelemiseen, mutta tämä tapa on hidas ja virheherkkä. Yhden robotin prosessointi aika saattaa kestää useita tunteja, jopa päivän, kun taas tekoäly kykenee laskemaan samat tehtävät muutamissa minuuteissa. Ajan säästön lisäksi myös taloudellinen hyöty on otettava huomioon. Poistamalla epävakaa robotit ja hyödyntämällä kaupanteossa useaa teknologiaa tavoitteista riippuen yritys säästyy ylläpito- ja kehityskustannuksilta robottien osalta.

Lähteet

1. What is RPA? Verkkoaineisto. UiPath. <https://www.uipath.com/automate/robotic-process-automation..> Luettu 14.1.2018.
2. What is robotic process automation? 2018. Verkkoaineisto. <https://irpaa.com/what-is-robotic-process-automation/>. Luettu 10.4.2018
3. Bell, Jason. 2012. Machine Learning. E-kirja. Wiley.
4. Letters pattern. 2015. Verkkoaineisto. Dribble. https://cdn.dribbble.com/users/1002282/screenshots/2389627/shot-3_1x.png. Luettu 30.10.2017.
5. Ferrucci, David ym. 2010. The AI Behind Watson. Verkkoaineisto. AI Magazine. <http://www.aaai.org/Magazine/Watson/watson.php>. Fall/2010. Luettu 31.10.2017.
6. Nielsen, Michael. 2017. Neural Networks and Deep Learning. E-kirja.
7. McCulloch, Warren S. Pitts, Walter. 5/1943. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of mathematic and biophysics.
8. Sailer, Ben. How to write white papers people actually wants to read. 2017. Verkkoaineisto. <https://coschedule.com/blog/how-to-write-white-papers/> Luettu 23.3.2018.
9. Li, Ray. 2016. History of Datamining. Verkkoaineisto <https://www.kdnuggets.com/2016/06/rayli-history-data-mining.html>. Luettu 20.3.2018.
10. Gorunescu, Florin. 2011. Data Mining. E-kirja.
11. The datamining process. Verkkoaineisto. IBM. https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.5.0/com.ibm.im.easy.doc/c_dm_process.html. Luettu 20.3.2018.
12. Numpy. 2017. Verkkoaineisto. Numpy. <http://www.numpy.org/>. Luettu 10.4.2018.
13. The evolution of rpa past – present. Verkkoaineisto. UiPath. <https://www.uipath.com/blog/the-evolution-of-rpa-past-present-and-future>. Luettu 14.1.2018.
14. Beginner’s guide to understanding convolutional neural networks. 2016. Verkkoaineisto. GitHub. <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>. Luettu 10.4.2018.

15. Wagner & Molla. 2018. Verkkoaineisto. Recode. <https://www.recode.net/2018/2/12/16998750/facebooks-teen-users-decline-instagram-snap-marketer>. Luettu 10.4.2018.
16. Play kauppa. Verkkoaineisto. Google inc. <https://play.google.com/store/apps/details?id=com.facebook.orca>. Luettu 13.4.2018.
17. Facebook voi huomaamatta kerätä puhelimesta yhteystietoja. 2016. Verkkoaineisto. Aamulehti. <https://www.aamulehti.fi/kotimaa/facebook-voi-huomaamatta-kerata-puhelimesta-yhteystietoja-etenkin-messenger-on-tosi-ovela-23856059/>. Luettu 4.5.2018

Tekoäly vakuutuskauppojen tukena

Sisällys

1	Johdanto	1
2	Tämän hetkinen tekniikka yrityksessä	2
3	Tekoälyn hyödyntäminen	3

1 Johdanto

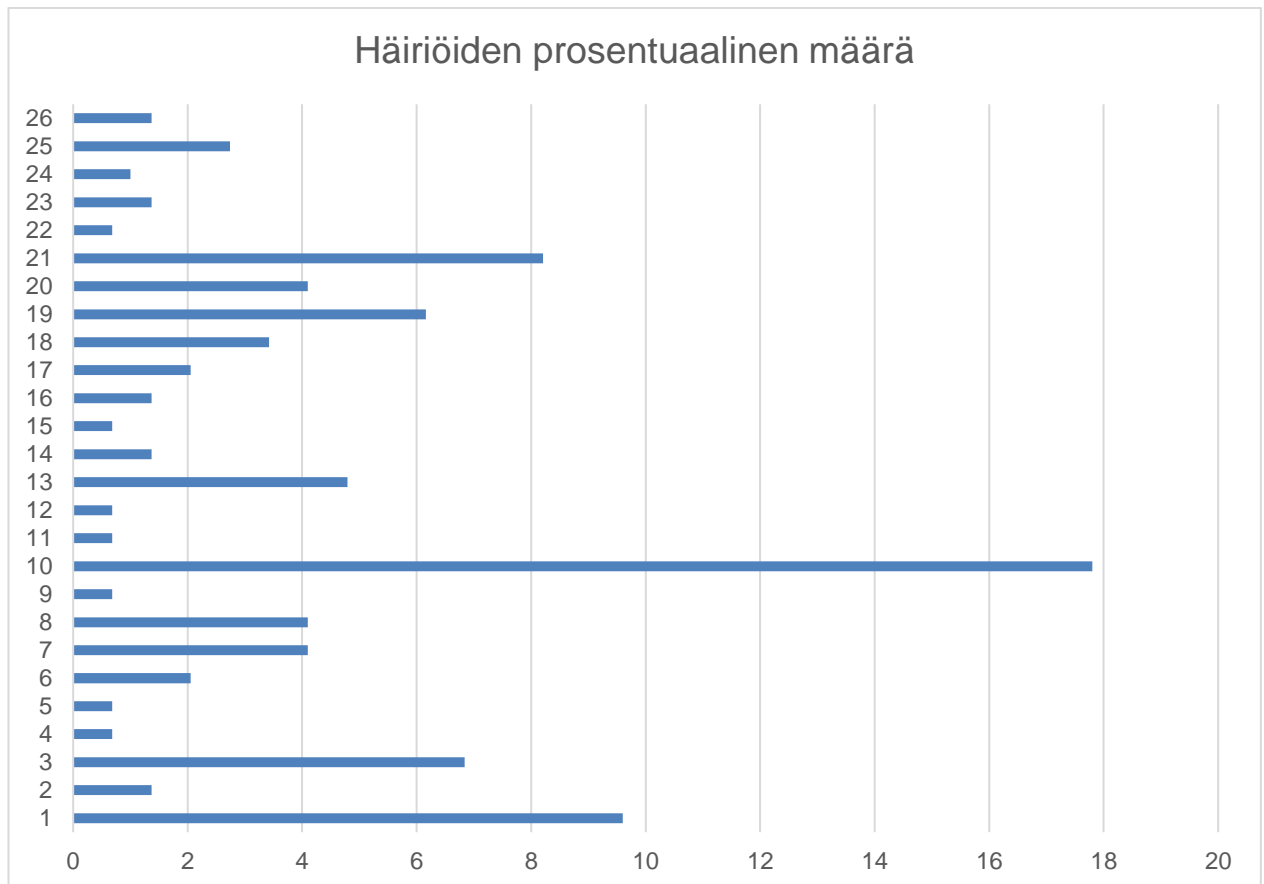
Teknologian kehittyessä yritysten on yhä tärkeämpää pysyä mukana vastatakseen kysyntään tehokkaasti ja kyetäkseen tarjoamaan kilpailukykyisiä vaihtoehtoja. Tavalliset myyntiä edistävät keinot eivät enää riitä, vaan on etsittävä jatkuvasti uusia tapoja ja mahdollisuuksia.

Tekoälyä on käytetty jo vuosia erilaisissa tehtävissä, erityisesti suuremmat supermarketit ovat valjastaneet sen esimerkiksi arvioimaan asiakkaan ostotottumuksia ja näin kohdentamaan mainontaa juuri oikeaksi. Jos supermarketit pystyvät hyödyntämään tätä, miksei myös pankki ja finanssi alan yritykset?

2 Tämän hetkinen tekniikka yrityksessä

Tällä hetkellä ohjelmistorobotiikka on tärkeä apuväline yrityksen myynnissä, mutta se aiheuttaa paljon omia kustannuksia ja ongelmia. Monet tehtävät ovat ulkoistettuna roboteille, kuten tarjousten lähettäminen, mahdollisten ostajien tunnistaminen ja asiakkaiden tietojen kerääminen eri kohdejärjestelmistä. Jokaista tehtävää varten on rakennettu yksi tai useampi robotti, joka toistaa sille valmiiksi opetettuja tehtäviä. Tällaisen robottien massa on kuitenkin osaltaan haastavaa ylläpitää niiden epävakaaisuuden ja omien erityispiirteiden vuoksi. Vuoden aikana ylläpidolle tulee pelkästään vakuutus puolen roboteista merkittävä määrä korjauspyyntöjä ja pienkehityksiä, mitkä itsessään jo tuovat kustannuksia yritykselle.

Tietysti monessa tehtävässä robotiikka on edelleen kannattava ratkaisu, mutta monet tehtävät kuten tiedon keruu ja potentiaalisten asiakkaiden etsiminen voitaisiin toteuttaa valjastamalla tekoäly ja koneoppiminen myynnin parantamiseksi. On huomioitava, että myös tämän kaltainen ratkaisu tuottaa omaa työtään ylläpidollisesti, mutta useamman robotin sijasta käsiteltävänä olisi yksi ohjelma, jonka opettamisesta ja huoltamisesta pitäisi huolehtia.



Kuvaajassa on esitetty diagrammilla ylläpidossa olevien vakuutusrobottien aiheuttamat häiriötiketit noin vuoden ajalta (Otos on vajavainen, sillä siitä on poistettu muun muassa eläkkeelle jääneet prosessit) kuvaajan x-akseli esittää häiriöiden lukumäärää prosentteina. Kuten kaaviosta voidaan todeta, osa prosesseista aiheuttaa huomattavasti enemmän ongelmia. Lisäksi rakenteilla on ollut jo pitkään prosesseja, joiden tehtävänä on kerätä tietoa, kyseiset prosessit eivät kuitenkaan ole edelleenkaan valmistuneet ylläpitoon valmiiksi. Yhtenä suurena ongelmana kyseisisissä prosesseissa voidaan pitää eri kohdejärjestelmissä navigoimisen haasteet. Nämä ovat ongelmia, jotka kuitenkin pystyttäisiin korjaamaan luomalla nykypäiväisempi, älykkäämpi ohjelma.

3 Tekoälyn hyödyntäminen

Rakentamalla tekoälyn profiloimaan ja etsimään mahdollisia asiakkaita voitaisiin epävaakaan robotiikan korvaamisen lisäksi edistää myyntiä. Tarkoituksena olisi, että asiakkaiden tietoja etsittäisiin hyödyntämällä tiedonlouhintaa. Määrittämällä asiakkaan algoritmit voisivat etsiä validin tiedon suurestakin massasta huomattavasti robotiikkaa nopeammin, eikä kohdejärjestelmissä navigoiminen ja pinta-automaatio hidastaisi prosessia.

Löydettyään tarpeellisen tiedon, käyttäisi tekoäly valmiiksi mallinnettua kaavaa päätösten tekemisessä. Itse opettaminen tapahtuisi syöttämällä kaupaksi menneitä tapauksia mahdollisimman suuri massa, jolloin tekoäly pystyy itsenäisesti määrittämään neuroneissa piilevät vakiot ilman ihmisen opastusta. Tämä tarkoittaa käytännössä sitä, että tekoälylle olisi rakennettu fully connected neuroverkosto, joka muuttaisi datan matriiseiksi. Matriisien avulla tekoäly pystyisi todennäköisyyksiä hyödyntäen etsimään kaikki tuotteet, jotka asiakas todennäköisimmin ostaisi. Esimerkkinä voidaan käyttää tapausta, jossa tekoäly on aikaisemmin huomannut, että 70 prosenttia tietyn tuotteen ottajista olisi miehiä. Kun tekoälylle syöttää asiakkaan joka on sukupuoleltaan mies, se asettaa kyseisen tuotteen painoarvoksi 0,7. Näin tekoäly jatkaa asettamalla ehtoja ja niiden painoarvoja tuottaen lopulta valmiin laskukaavan, jonka perusteella päätös tehdään. Mitä lähempänä ulos saatava output on 1:stä, sitä todennäköisemmin asiakas on valmis ostamaan tuotteen (output on aina välillä 0-1).

On huomioitava, että mitä enemmän tekoäly saa tapauksia käsiteltäväksi, sitä varmemmaksi se tulee. Tämä tarkoittaa, että on hyvä opettaa sitä jatkuvasti lisää, jotta tekoäly oppisi löytämään lisää pieniä, melkein huomaamattomia yhteyksiä asiakkaiden väliltä.

Yhteenveto

Korvaamalla osan robotiikasta tekoälyllä pystyttäisiin yrityksen myyntiä lisäämään huomattavasti. Jatkuvasti oppiva tekoäly kehittyisi iän myötä paremmaksi, toisin kuin robotiikka, joka on yhä epästabiilimpi ja virheherkempi, mitä vanhemmaksi se tulee.



Tekoäly vakuutusalaalla

Helmikuu 2018

© CGI Group Inc.

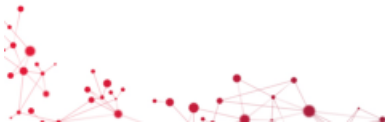
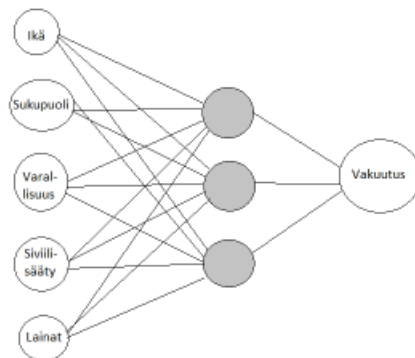
CGI

Experience the commitment®

Toimintamalli

- Tekoälylle syötetään riittävä määrä dataa kaupaksi menneistä vakuutustarjouksista
 - Mitä enemmän algoritmille antaa vanhoja tapauksia, sitä varmemmaksi sen toimintavarmuus muuttuu
- Tekoäly tunnistaa datasta toistuvia kaavoja hyödyntäen koneoppimisessa käytettäviä hermoverkostoja
- Algoritmi etsii opittuja kaavoja asiakkaista ja tunnistaa näin potentiaalisia vakuutuksien ostajia.





3

CGI

Toimintamalli

- Tekoäly etsii yhtäläisyyksiä vakuutusten ostajista ja näiden perusteella asettaa vakiot algoritmiin
 - Esim. 70% tietyn autovakuutuksen ostajista ovat miehiä, algoritmillemme annetaan sukupuolen kohdalle miehillä painoarvoksi 0,7 ja naisille 0,3
- Mitä lähempänä algoritmin antama output on ykköstä, sitä varmemmin asiakas valitsisi kyseisen vakuutuspaketin



4

CGI

Hyödyt

- Potentiaalisten asiakkaiden löytäminen helpottuu
 - Myynnin kasvu
- Virhe marginaali pienenee
 - Tekoäly oppii jatkuvasti, joten osumatarkkuus paranee jokaisen uuden tapauksen myötä
- Osa epävakaita ohjelmistoroboteista voidaan korvata
 - Tiedon keruu robotteja ei enää tarvita, sillä tekoäly hyödyntää tiedonlouhintaa datan keruussa



5

CGI

Hyödyt verrattuna ohjelmistorobotiikkaan

- Huomattavasti parempi laskentateho
- Ei yhtä herkkä virheille
- Kohdejärjestelmien ja käsiteltävän datan lisääntyminen ei hidasta tekoälyn toimintaa
- Kyky kehittyä



6

CGI