



Kandidatutkielma

Tietojenkäsittelytieteen kandiohjelma

Koneoppiminen lääkkeiden kehityksessä

Heikki Pulli

14.12.2021

MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
HELSINGIN YLIOPISTO

Ohjaaja(t)

Leinonen, Matti

Tarkastaja(t)

Prof. Ruotsalainen, Laura

Yhteystiedot

PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

Sähköpostiosoite: info@cs.helsinki.fi

URL: <http://www.cs.helsinki.fi/>

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen kandiohjelma	
Tekijä — Författare — Author			
Heikki Pulli			
Työn nimi — Arbetets titel — Title			
Koneoppiminen lääkkeiden kehityksessä			
Ohjaajat — Handledare — Supervisors			
Leinonen, Matti			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Kandidutkielma	14.12.2021	27 sivua	
Tiivistelmä — Referat — Abstract			
<p>Tässä tekstissä käydään käydään läpi, kuinka koneoppimista voidaan käyttää lääkekehityksessä ja minkälaista dataa voidaan käyttää koneoppimismallien kouluttamisessa. Tekstin alussa käydään nopeasti lääkekehityksen eri vaiheita ja miksi koneoppimista halutaan käyttää lääkekehityksessä. Seuraavissa kappaleissa keskitytään ainoastaan lääkkeiden löytämiseen ja lääkkeiden syntetisoinnin suunnitteluun ja kuinka koneoppimista voidaan hyödyntää näissä vaiheissa. Kappaleissa käydään myös läpi kolmen eri koneoppimismallin toimintaa, joista yksi keskittyy uusien lääkkeiden löytämiseen ja kaksi keskittyvät yhdisteiden syntetisoinnin suunnitteluun. Tekstin alussa kerrotaan myös eri datatyypeistä, joita käytetään kyseisissä koneoppimismalleissa.</p> <p>Koneoppimismallien tarkoituksena on antaa lukijalle ajatus siitä, miten koneoppimista voidaan hyödyntää kyseisissä lääkekehityksen vaiheissa, minkälaisia ongelmia koneoppimismalleilla pyritään ratkomaan ja minkälaisia koneoppimismalleja ja tekniikoita ongelmien ratkomisessa on käytetty.</p>			
<p>ACM Computing Classification System (CCS) Computing Methodologies → Machine Learning → Machine Learning Approaches → Neural Networks, Applied computing → Life and medical sciences → Health informatics</p>			
Avainsanat — Nyckelord — Keywords			
Koneoppiminen, Lääkekehitys			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsingin yliopiston kirjasto			
Muita tietoja — övriga uppgifter — Additional information			

Sisällys

1	Johdanto	1
2	Kemiallinen data	3
2.1	Yhdisteiden kuvaaminen merkkijonoilla	3
2.2	Yhdisteiden kuvaaminen bitteinä, ECFP	4
2.3	Ominaisuusvektorit	4
3	Uusien lääkkeiden löytäminen	5
3.1	Lääkkeiden löytäminen	5
3.2	Virtuaalinen seulonta	5
3.3	Kehitettyjä koneoppimismalleja	6
3.3.1	CMG -malli	6
4	Uusien lääkkeiden syntetisointi	11
4.1	Yhdisteiden syntetisointi	11
4.2	Lääkkeen retrosyntetisoinnin haastavuus	12
4.3	Kehitettyjä apuvälineitä	13
4.3.1	3N-MCTS -malli	13
4.3.2	ICHO -malli	20
5	Yhteenveto	24
	Lähteet	25

1 Johdanto

Uusien lääkkeiden kehittäminen on pitkä ja kallis prosessi. Tähän kuuluu useita vaiheita ja eri vaiheet vievät eri määrän rahaa ja aikaa. Vaiheet ovat sairauden aiheuttajan tunnistaminen, aiheuttajaan vaikuttavan lääkkeen tunnistaminen, lääkkeen optimointi, lääkkeen ominaisuuksien analysointi ja kliiniset testit. Näiden vaiheiden jälkeen lääke joko hyväksytään myyntiin tai sitä joudutaan edelleen jatkokehittämään. Kaikki vaiheet yhdessä ovat tavallisesti vieneet 10–12 vuotta ja niiden hinnaksi on voinut muodostua 1–3 miljardia dollaria [8].

Vaiheiden pitkän keston ja korkean hinnan takia tutkijat ja lääkealan yritykset ovat alkaneet tutkia mahdollisia keinoja, jotka nopeuttaisivat lääkkeiden tutkimista tai lasisivat lääkekehityksen prosessin hintaa. Koneoppimismallit ovat nousseet houkuttelevaksi vaihtoehdoksi ratkaisemaan näitä ongelmia. Yritykset ovat alkaneet selvittää, kuinka eri koneoppimismalleja voidaan käyttää lääketutkimuksen suurimpien pullonkaulojen poistamiseksi [8].

Viimeisen kymmenen vuoden aikana saatavilla olevan laadukkaan datan määrä on kasvanut merkittävästi ja samalla on kehitetty uusia ja tehokkaampia koneoppimismalleja, joita voidaan hyödyntää lääketutkimuksessa [5, 24]. Koneoppimismallien avulla voidaan esimerkiksi karsia kaikista harkinnasta olevista lääkkeistä lupaavimmat kandidaatit, joilla on mahdollisuus päästä testeistä läpi tuotantoon asti [10]. Koneoppimismalleja voidaan myös hyödyntää aivan uusien lääkkeeksi sopivien yhdisteiden etsimisessä [21, 11]. Tähän tarkoitukseen kehitetyt koneoppimismallit kykenevät etsimään lääkkeitä, joilla on haluttu lääkkeelliset ja fysikaaliset ominaisuudet [24]. Kehitetyt mallit ovat näyttäneet, että koneoppimismallit ovat tehokkaita työkaluja, joita voidaan hyödyntää kaikissa lääketutkimuksen eri vaiheissa.

Tässä tekstissä paneudutaan muutamiin koneoppimismalleihin, joita käytetään uusien lääkkeiden tunnistamisessa ja näiden tunnistettujen lääkkeiden syntetisoinnin suunnittelussa. Tekstin tarkoituksena on tutustuttaa lukija koneoppimisen käsitteistöön ja erilaisiin malleihin ja ongelmiin, jotka liittyvät lääkkeiden löytämiseen ja löydettyjen lääkkeiden syntetisoinnin suunnitteluun. Käsiteltävät koneoppimismenetelmät toimivat esimerkkeinä, kuinka koneoppimista voidaan käyttää käsiteltävissä konteksteissa.

Ensimmäisessä kappaleessa tutustutaan eri datatyyppeihin, joita käytetään koneoppimis-

mallien kouluttamisessa. Tämä on lyhyt kappale, jossa käydään vain ne datatyypit läpi, joita myöhemmin selitettävät koneoppimismallit käyttävät. Näiden ymmärtäminen auttaa hahmottamaan myöhemmin tekstissä selitettävien eri mallien toimintaperiaatteita.

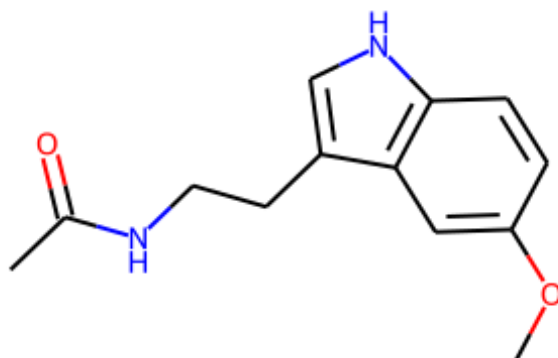
Toisessa kappaleessa käydään läpi, kuinka uusia lääkkeitä etsitään ja löydetään. Kappaleessa käydään pintapuolisesti läpi virtuaalista seulontaa (Virtual Screening), mitä sillä tarkoitetaan ja kuinka sitä hyödynnetään. Tässä kappaleessa käydään myös läpi yhden koneoppimismallin toimintaa, joka on koulutettu löytämään uusia yhdisteitä. Tämä malli mahdollistaa uusien lääkeyhdisteiden nopean löytämisen, joita päästään jatkotutkimaan nopeasti.

Kolmas kappale paneutuu yhdisteiden syntetisointiin ja kuinka sitä voidaan toteuttaa hyödyntämällä koneoppimismalleja. Kappaleessa käsitellään, kuinka syntetisointia voidaan suunnitella retrosynteesianalyysin avulla, mikä tekee yhdisteiden retrosyntetisointianalyysistä hankalaa ja mitä ratkaisuja tähän on kehitetty. Kappale kertoo CASP -menetelmistä (Computer Aided Synthesis Planning) ja kuinka tämä liittyy retrosyntetisointianalyysiin. Kappaleessa käydään myös läpi kahden eri koneoppimismallin toimintaa, jotka on kehitetty auttamaan eri yhdisteiden syntetisoinnin suunnittelussa.

2 Kemiaallinen data

Tässä kappaleessa käydään pintapuolisesti läpi, minkälaista dataa käytetään koneoppimis-
mallien kouluttamisessa. Kappaleessa keskitytään datatyyppeihin, joita käytetään koneop-
pimismalleissa, joiden toimintaa tullaan avaamaan tämän tekstin muissa osissa. Koneop-
pimismallit, joiden toimintaa avataan, käyttävät datatyyppejä, joilla kuvataan pienyhdis-
teitä. Tämän takia kappaleessa keskitytään vain pieniäyhdisteitä kuvaaviin datatyyppei-
hin. Kappaleessa käydään läpi SMILES ja SMARTS datatyypit, mihin niitä käytetään ja
kuinka ne eroavat toisistaan. Kappaleessa myös kerrotaan, mitä ovat yhdisteiden ECFP
-tunnisteet, mitä ominaisuusvektorit ovat ja mitä tietoa ne sisältävät yhdisteistä.

2.1 Yhdisteiden kuvaaminen merkkijonoilla



CC(=O)NCCC1=CNC2=C1C=C(C=C2)OC

Kuva 2.1: Melatoniini yhdiste, kuinka se kuvataan graaffina ja SMILES -merkijonona. Kuva on tuotettu hyödyntämällä RDKit -kirjastoa.

Tavallinen tapa kuvata yhdisteitä on kuvata ne yhdellä rivillä merkkijonoina. SMILES (Simplified Molecular Input Line Entry System) on David Weiningerin vuonna 1988 kehittämä tapa kuvata molekyyliä ja yksinkertaisia yhdisteitä [26]. Se kuvaa yhdisteen yhdellä rivillä käyttäen ASCII -merkistöä. Kuvassa 2.1 on esitetty melatoniini yhdiste ja

kuinka se kuvataan SMILES muodossa. Tämän takia se on tietokoneen nopea lukea ja kemistin helppo ymmärtää.

SMARTS (SMILES Arbitrary Target Specification) on SMILES -standardista jatkokehitetty tapa kuvata yhdisteitä. Se mahdollistaa yhdisteiden sisäisten rakenteiden määrittämisen. SMARTS -on yläluokka SMILES -standardista, jonka takia kaikki SMILES -merkkijonoiksi kelpaavat merkkijonot ovat myös kelpoja SMARTS -merkkijonoja. SMARTS -ja SMILES -standardit mahdollistavat myös kemiallisten reaktioiden kuvaamisen ASCII -merkistöllä yhdellä rivillä.

2.2 Yhdisteiden kuvaaminen bitteinä, ECFP

Toinen tavallinen tapa kuvata yhdisteitä on muuttaa ne ECFP (Extended Connectivity FingerPrint) -bittijonoiksi [18]. ECFP -tunniste lasketaan käyttämällä Morganin algoritmia. Algoritmi laskee jokaiselle yhdisteen atomille numeerisen arvon käyttämällä hajautusfunktioita ja iteroimalla ja laskemalla näitä arvoja yhteen se saa laskettua yhdisteen eri osille niitä kuvaavat numeeriset arvot. ECFP -tunnisteita kutsutaan myös Morgan -tunnisteiksi (Morgan Fingerprint). Nämä numeeriset arvot voidaan muuttaa sitten halutun mittaisiksi bittiesityksiksi [18]. ECFP on yksi monista tavoista kuvata yhdisteitä numeerisessa muodossa, mutta se on yksi käytetyimmistä menetelmistä sen nopean ja helpon laskemisen takia. Bittijonojen etuna on se, että lyhyet bittijonot vievät vähän muistia, mutta lyhyen niiden ongelma on, että ne saattavat menettää oleellista tietoa yhdisteestä bittitörmäysten takia. Tämän takia tavallisesti käytetään 4096-bitin mittaisia bittijonoja.

2.3 Ominaisuusvektorit

Yksi tässä tekstissä käsiteltävistä koneoppimismalleista käyttää ominaisuusvektoreita hyödykseen. Ominaisuusvektorit sisältävät tietoa yhdisteen tai molekyylin lääkkeellisistä ominaisuuksista. Ominaisuusvektorit voivat kuvata mitä tahansa yhdisteen tai molekyylin ominaisuutta ja ne voidaan valita tilanteen mukaan. Kyseisessä mallissa ominaisuusvektorit kuvaavat molekyylin PlogP, QED ja DRD2 arvoja [21]. PlogP (Penalized LogP) kuvaa aineen rasvaliukoisuutta, QED (Quantitative Estimate of Drug-likeness) kuvaa tietyn yhdisteen lääkkeenkaltaisuutta ja DRD2 kuvaa, kuinka paljon molekyyli reagoi dopamiini 2 receptoria vastaan [4]. Nämä ovat oleellisia tietoja tutkittaessa uusia mahdollisia lääkedyhdisteitä [21].

3 Uusien lääkkeiden löytäminen

Tässä kappaleessa kerrotaan, mitä asioita liittyy lääkkeiden löytämiseen. Kappaleessa käydään myös läpi, mitä on virtuaalinen seulonta ja kuinka sitä hyödynnetään lääkkeiden etsinnässä. Kappaleessa myös käydään läpi yhden koneoppimismallin toimintaperiaate. Kyseinen koneoppimismalli luo uusia lääkkeitä kuvaavia SMARTS -merkkijonoja, joilla on mahdollisimman lähellä haluttuja ominaisuuksia olevat ominaisuudet.

3.1 Lääkkeiden löytäminen

Yksi ensimmäisistä lääkekehityksen osa-alueista on uusien lääkkeiksi kelpaavien yhdisteiden löytäminen joko uusiin tai jo tunnettuihin tauteihin [8]. Lääkeyhdisteiden löytäminen on kuitenkin ollut tavallisesti hidas prosessi ja uuden toimivan yhdisteen löytäminen on kestänyt useita vuosia [8, 15]. On itse asiassa huomattu, että uusien lääkkeiden löytäminen on muuttunut yhä kalliimaksi ja uusia lääkkeitä tulee markkinoille vähemmän kuin aikaisemmin [15]. Koneoppimista on alettu hyödyntämään tässä vaiheessa prosessiin kuluvan ajan nopeuttamiseksi [24].

Viimeisen kymmenen vuoden aikana saatavilla olevan datan määrä koskien lääkkeitä ja lääkeyhdisteitä on kasvannut merkittävästi kehitettyjen tietopankkien ansiosta [8]. Tietopankkeja ovat esimerkiksi PubChem ja ChEMBL [1, 9]. Molemmat näistä keskittyvät lääkeyhdisteiden tietojen tallentamiseen. PubChem on Yhdysvaltojen terveysviraston (National Institute of Health) ylläpitämä ja ChEMBL on Euroopan bioinformatiikka instituution (European Bioinformatics Institute, EBI) ylläpitämä.

3.2 Virtuaalinen seulonta

Kemiallisten yhdisteiden avaruudella tarkoitetaan kaikkien uniikkien yhdisteiden lukumäärää. On arvioitu, että erilaisia kemiallisia yhdisteitä, jotka voivat esiintyä huoneenlämmössä nesteessä, voi olla välillä $10^{18} - 10^{180}$ [22]. Yhdisteet, jotka täyttävät lääkkeitä vaaditut kriteerit, on puolestaan arvioitu olevan noin 10^{60} [22]. Erillaisten yhdisteiden suuri lukumäärä itsessään esittää tarpeen tehokkaille algoritmeille ja menetelmille, jotka

auttavat karsimaan tästä suuresta määrästä kemiallisia yhdisteitä vain lupaavimmat.

Virtuaalinen seulonta (Virtual Screening) on suosittu lähestymistapa uusien lääkkeiden tai lääkkeeksi kelpaavien yhdisteiden löytämiseksi. Virtuaalinen seulonta käsittää joukon menetelmiä, joissa tietokoneita hyväksi käyttäen karsitaan yhdisteiden avaruudesta vain tietyt kriteerit täyttävät yhdisteet [22]. Yhdisteiden karsimisessa käytetään ohjelmistoja, joissa toteutetut algoritmit suorittavat seulontaa annettujen parametrien avulla.

Koneoppimismenetelmiä voidaan myös käyttää virtuaalisessa seulonnassa [22]. Yleisimpiä koneoppimismenetelmiä ovat tällä hetkellä SVM (Support Vector Machine) ja Random forest -mallit [22]. Neuroverkko ja ensemble -koneoppimismallit ovat myös yleistyneet viime vuosina [21, 17]. Ensemble -mallin ideana on yhdistää useita eri koneoppimismalleja yhdeksi malliksi [19]. Ensemble -mallit ovat tavallisesti parempia ja monipuolisempia toiminnallisuuksissaan kuin yksittäiset koneoppimismallit ja ne myös tuottavat tarkempia tuloksia.

koneoppimismallien avulla löydettyjä yhdisteitä voidaan jatkotutkia ja kehittää edelleen lääkkeiksi.

3.3 Kehitettyjä koneoppimismalleja

Seuraavaksi käydään läpi erään toteutetun koneoppimismallin toimintaperiaate. Kyseinen malli etsii uusia lääkkeitä käyttämällä hyödyksi sille annettua prototyyppi-yhdistettä, prototyypin kemiallisia ominaisuuksia ja uudelta lääkkeeltä haluttuja kemiallisia ominaisuuksia. Kemialliset ominaisuudet on koodattu ominaisuusvektoreihin.

3.3.1 CMG -malli

CMG (Controlled Molecule Generator) on koneoppimismalli, joka etsii uusia molekyyilejä, jotka perustuvat syötteenä annettuun molekyyliin ja joilla on ennalta määritellyt halutut ominaisuudet [21]. CMG -mallin toimintaperiaatteen ajatuksena on muokata prototyyppi-yhdistettä siten, että se maksimoi halutut ominaisuudet, jotka on määriteltä sen syötteenä saamassa ominaisuusvektorissa. CMG -mallin toimintaa rajoittaa kuitenkin prototyyppi-yhdiste ja syötteenä annettu samanlaisuusluku, jonka alle uuden luodun yhdisteen ja prototyyppi-yhdisteen välinen tanimoto -indeksi ei saa mennä. Tanimoto -indeksi kuvaa, kuinka paljon molekyylit muistuttavat toisiaan rakenteellisesti [14]. Tanimoto -indeksin arvot ovat välillä $[0, 1]$.

CMG -malli eroaa aikaisemmin kehitetyistä malleista siten, että se pystyy optimoimaan annetun molekyylin useampaa ominaisuutta samanaikaisesti. Aikaisemmin kehitetyt mallit ovat pystyneet optimoimaan vain yhtä molekyylin ominaisuutta.

CMG -malli muodostuu kolmesta verkosta, joista kaksi verkkoa toimivat rajoiteverkkoina. Ensimmäinen verkko on MTN (Molecule Translation Network) ja rajoiteverkot ovat PPN (Property Prediction Network) ja SPN (Similarity Prediction Network). Kuvassa 3.1 näkyy CMG -mallin rakenne ja toimintaperiaate.

CMG:n kehittäjät lähestyvät molekyylin ominaisuuksien optimointiongelmia merkkijonon käännös - ja luontiongelmana. CMG:lle opetetaan, kuinka syötteenä annettu yhdistemerkkijono käännetään sellaiseksi yhdistemerkkijonoksi, jolla on lähimpänä haluttuja ominaisuuksia olevat ominaisuudet. CMG tulkitsee ensin annetut merkkijonot hyödyntäen DN:ää (Deep Network) jonka jälkeen se luo uusia molekyyliyhdisteitä hyödyntäen tätä tulkintaa ja haluttua ominaisuusvektoria. Koska ominaisuudet annetaan vektorina, niin CMG pystyy optimoimaan useampaa ominaisuutta samanaikaisesti. CMG hyödyntää myös etukäteen koulutettuja rajoiteverkkoja (Constraint Network), jolloin vältytään luomasta mahdottomia yhdisteitä. CMG -malli käyttää näitä verkkoja hyödyksi käyttämällä muokattua Beam Search -algoritmia.

Beam Search -algoritmi saa syötteenä kaikki CMG -mallin kehittämät yhdisteet, pisteyttää ne ja palauttaa yhdisteen, joka sai parhaimman pisteytyksen. Algoritmi käyttää PPN -verkkoa ennustaessaan kehitetyn molekyylin ominaisuuksia ja SPN -verkkoa laskiessaan, kuinka samankaltainen kehitetty yhdiste on CMG -mallin syötteenä saadun yhdisteen kanssa. Näiden tietojen avulla algoritmi laskee kehitetyille yhdisteille numeeriset arvot ja palauttaa yhdisteen, joka saa parhaimman arvon.

Data, jolla mallit koulutetaan, on peräisin ZINC -tietokannasta ja DRD2 -dataseiteistä. ZINC on ilmainen tietokanta, joka sisältää tietoa lääkeyhdisteistä, ja jota voidaan käyttää virtuaalisessa seulonnassa [23]. ZINC tietokantaa ylläpitää Irwin ja Shoichet laboratoriot Kalifornian yliopistossa. DRD2 -datasetti sisältää tietoa yhdisteistä ja kuinka nämä yhdisteet reagoivat dopamiini 2 receptorin kanssa. DRD2 -datasetti luotiin Olivecrona et al. tutkimuksen yhteydessä ja sitä käytetään samalla tavalla CMG -mallissa [21].

Data sisältää kaiken kaikkiaan 257 565 molekyyliä, joista luodaan pareja. Parit muodostetaan siten, että kahden molekyylin välinen tanimoto -indeksi on yli 0.4. Näitä (X, Y) pareja muodostetaan kaiken kaikkiaan 10 827 615 kappaletta. Lisäksi kaikille molekyyleille lasketaan niiden eri kemiallisten ominaisuuksien arvot, jotka ovat PlogP, QED ja DRD2. Nämä kuvaavat eri ominaisuuksia, joita malli pyrkii optimoimaan halutulla tavalla.



Kuva 3.1: CMG:n toimintaperiaate kuvattuna

[21]

MTN -verkko

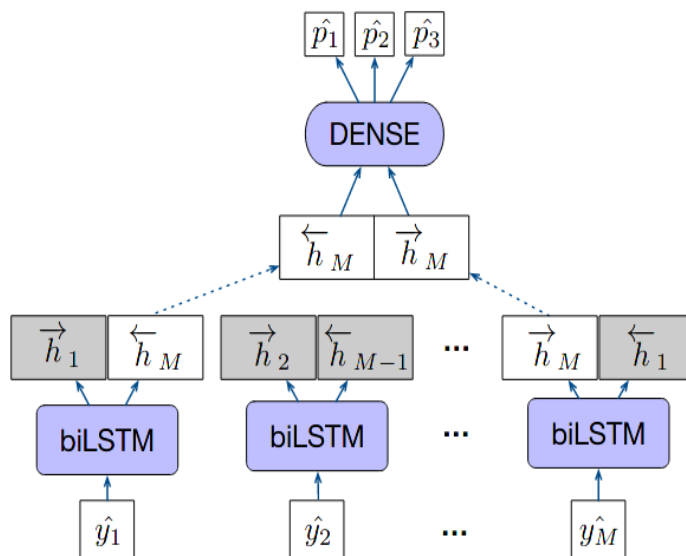
MTN -verkko perustuu aikaisemmin kehitettyyn Transformer -malliin [25]. Transformer -malli eroaa aikaisemmin yleisistä NLP -malleista siten, että se perustuu täysin attention -perusteiseen lähestymistapaan. Transformer -malli noudattaa encoder-decoder arkkitehtuuria. Encoder muuntaa ensin saamansa syötteen jokaisen sanan todennäköisyysjakaumaksi ja nämä todennäköisyysjakaumat syötetään decoder -osiolle. Tämän todennäköisyysjakauman ja aikaisemman sanan avulla se määrittää seuraavan sanan tulosteeseen.

MTN eroaa Transformer -mallista kahdella tavalla. Toisin kuin Transformer -malli, joka käsittelee sanoja ja niistä muodostettuja lauseita, MTN -verkko käsittelee yksittäisiä merkkejä ja niistä muodostettuja molekyyliä. Lisäksi MTN:än piilotettuun kerrokseen on lisätty tietoa kemiallisista ominaisuuksista.

PPN -verkko

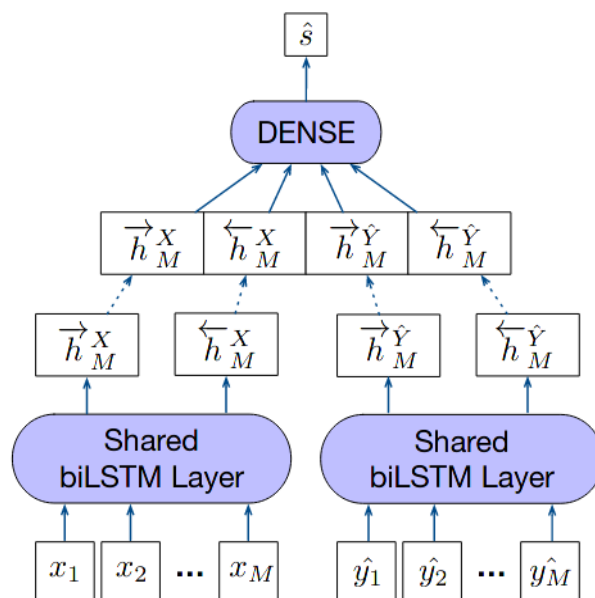
PPN -verkon rakenne ja toiminta periaate näkyvät kuvassa 3.2. PPN -verkko ottaa syötteenä ennustetun molekyylin merkkijonon (y_i). Tämä merkkijono muunnetaan piilotetuiksi vektoreiksi hyödyntäen Bidirectional Long short-term memory (BiLSTM) -kerrosta. BiLSTM on rakenteeltaan samanlainen kuin LSTM taso, mutta koska se käsittelee syötteen sekä alku-loppu suunnassa että loppu-alku suunnassa se oppii tietyn merkin kontekstin molemmissa suunnissa. BiLSTM -kerros muodostaa vektoreita oikealta vasemmalle ja vasemmalta oikealle suunnissa. Näistä vektoreista valitaan molempien suuntien viimeiset vektorit. Nämä vektorit yhdistetään ja yhdistevektori syötetään täysin yhdistetylle neuroverkolle. Tämä verkko sisältää kaksi piilotettua tasoa. Verkko antaa tuloksena kyseisen yhdisteen ominaisuuksien ennusteen.

PPN koulutetaan käyttäen 257 565 eri molekyyliä. Nämä molekyylit jaetaan sattumanvaraisella valinnalla koulutus -ja testiryhmiin suhteessa 8:2.



Kuva 3.2: CMG -mallin PPN -neuroverkon rakenne
[21]

SPN -verkko



Kuva 3.3: CMG -mallin SPN -neuroverkon rakenne
[21]

SPN verkon rakenne ja toimintaperiaate on kuvattu kuvassa 3.3 SPN ottaa syötteenä ennustetun molekyylin merkkijonon (y_i) ja alkuperäisen molekyylin merkkijonon (x_i). Nä-

mä merkkijonot syötetään kahdelle eri biLSTM -tasolle, toinen käsittelee alkuperäisen molekyylin merkkijonot ja toinen ennustetun molekyylin merkkijonot. Nämä biLSTM -kerrokset toimivat samalla periaatteella kuin PPN:ässä ja palauttavat näin ollen neljä eri vektoria, kaksi ennustetulta merkkijonolta ja kaksi alkuperäiseltä merkkijonolta. Nämä vektorit yhdistetään ja tämä yhdistevektori annetaan täysin yhdistetylle verkolle, joka sisältää kaksi piilotettua tasoa. Tämä verkko toimii binääriluokittelijana ja palauttaa tiedon, onko kaksi syötemolekyyliä samanlaisia määritellyn samanlaisuusrajan mukaan.

SPN -verkonkouluttamiseen käytetään osajoukkoa 10 827 615 eri parista. Tästä määrästä valitaan kymmenen prosentin osajoukko, joka on 997 773 molekyylin suuruinen. Tämä joukko jaetaan samalla tavalla koulutus -ja testiryhmiin kuin PPN -verkossa.

4 Uusien lääkkeiden syntetisointi

Tässä kappaleessa käydään läpi, miten lääkkeiden syntetisoinnin suunnittelua voidaan tehostaa koneoppimismallien avulla. Kappaleessa keskitytään syntetisoinnin suunnitteluun käyttämällä retrosyntetisointimenetelmää. Kappaleessa kerrotaan, mikä tekee yhdisteen retrosyntetisoinnista vaikeaa ja kuinka sitä voidaan tehostaa hyödyntämällä eri koneoppimismalleja. Kappaleessa käydään läpi kahden koneoppimismallin toimintaa, jotka suorittavat halutun yhdisteen retrosyntetisointia.

4.1 Yhdisteiden syntetisointi

Yhdisteen syntetisoinnin suunnittelulla tarkoitetaan prosessia, jossa määritellään, kuinka haluttu yhdiste voidaan tuottaa synteettisesti saatavilla olevista lähtöaineista [6]. Retrosynteesianalyysillä tarkoitetaan puolestaan menetelmää, jonka avulla löydetään halutun yhdisteen tuottamiseen tarvittavat lähtöaineet. Retrosynteesianalyysi toimii siis toiseen suuntaan kuin syntetisointi. Retrosynteesissä yhdiste pilkotaan rekursiivisesti pienempiin lähtöaineisiin kunnes jäljellä on vain saatavilla olevia lähtöaineita [7].

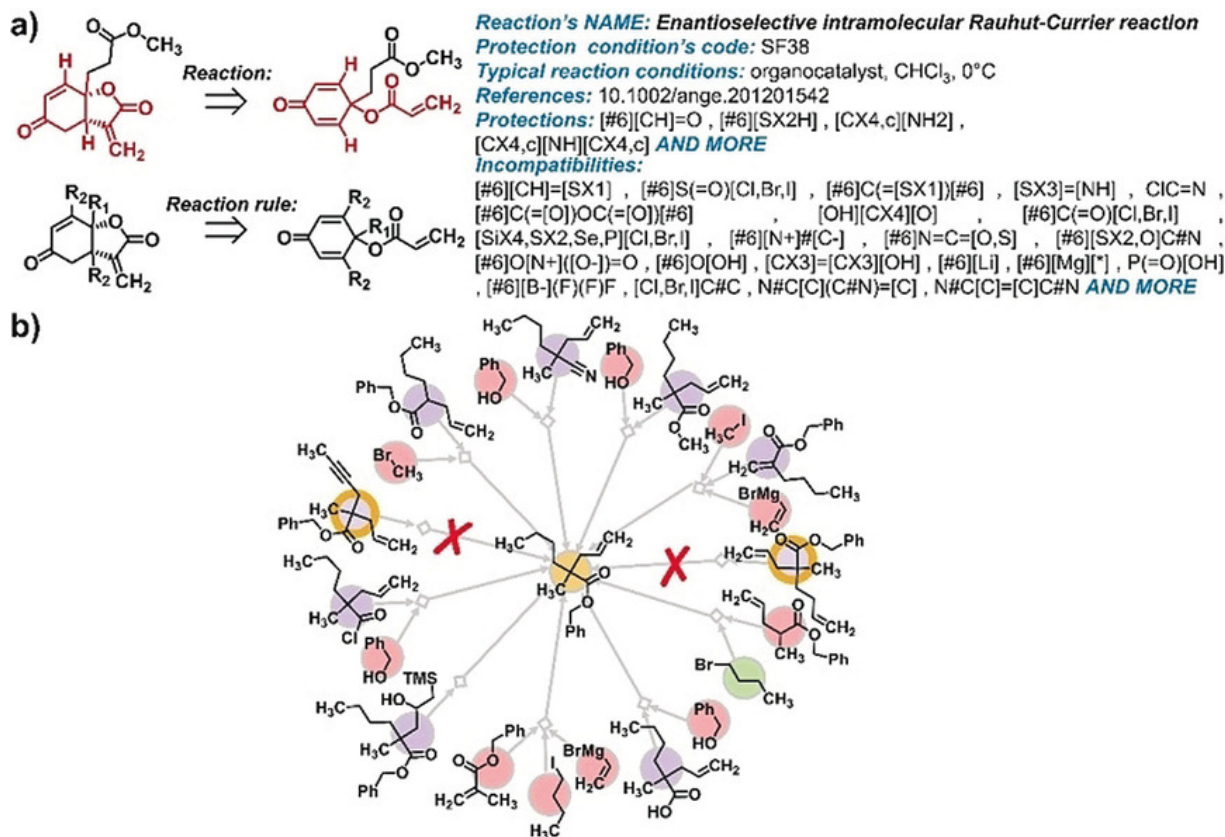
Tavallisesti yhdisteen retrosyntetisointi on vaatinut suorittavalta kemistiltä usean vuoden kokemusta ja tietoa saatavilla olevista lähtöaineista ja eri reaktioista. Retrosyntetisointia on pyritty automatisoimaan eri CASP -menetelmien avulla (Computer-Aided Synthesis Planning). CASP on jo 1960 -luvulla kehitetty ajatus, ja ensimmäisiä yrityksiä toteuttaa retrosyntetisointia tekoälyn avulla oli E.J Coreyn LHASA -projekti [6]. LHASA perustui heuristisiin algoritmeihin, joissa kemistit käsin koodasivat, miten eri lähtöaineet reagoivat keskenään ja mikä on reaktion lopputuote [16]. Tämä on kuitenkin osoittautunut toivotomaksi yritykseksi valtavan datamäärän takia.

Kehitys koneoppimismenetelmissä on kuitenkin tarjonnut uuden lähestymistavan CASP -menetelmien kehityksessä. Sen sijaan, että kemistit kehittäisivät heuristisia malleja, uudet koneoppimismallit koulutetaan saatavilla olevan datan avulla. Tämä on todettu merkittävästi paremmaksi lähestymistavaksi toteuttaa [6].

Koneoppimismallien käyttö ja koulutus ei ole kuitenkaan täysin ongelmaton lähestymistapa. Ongelmaan liittyvää dataa ei välttämättä ole saatavilla ja datan hankkiminen voi

olla kallista [2]. Ei ole julkista, kaikille avointa tietopankkia, joka sisältäisi tietoa kemiallisista reaktioista, vaan ainoastaan kaupallisia tietopankkeja on kehitetty, kuten Reaxys ja SciFinder [2].

4.2 Lääkkeen retrosyntetisoinnin haastavuus



Kuva 4.1: (a) esimerkki reaktiosäännöstä, (b) esimerkki mahdollisista reaktioista, kuinka voidaan luoda haluttu yhdiste (keskellä). Kuvaa on yksinkertaistettu ja se sisältää vain yksitoista mahdollista reaktiota, jotka tuottavat halutun yhdisteen. Värikoodaukset tarkoittavat: keltainen — haluttu yhdiste, punainen — saatavilla oleva yhdiste, vihreä — kirjallisuudesta tunnettu yhdiste, liila — tuntematon yhdiste

[3]

Retrosyntetisoinnin tekee hankalaksi se, että tutkittaessa kuinka haluttu yhdiste voidaan muodostaa, tulee käydä läpi satojatuhansia mahdollisia kemiallisia muunnoksia, joiden avulla voidaan luoda haluttu yhdiste [20]. Kemiallisella muunnoksella tai myöhemmin käytetyllä termillä muunnossääntö tarkoitetaan yleisesti kaikkia mahdollisia reaktioita tai reaktiotyyppejä, joita kemiassa tunnetaan. Tämä ongelma toistuu rekursiivisesti, kun yhdiste pilkotaan yhdisteisiin, jotka keskenään reagoidessaan muodostavat alkuperäisen yh-

disteen. Kuvassa 4.1 näkyy, kuinka monella tavalla yksinkertainen yhdiste voidaan muodostaa. Pienille ja yksinkertaisille yhdisteille tämä muunnostenavaruus on pienempi, mutta yhdisteen koon kasvaessa eri tapojen määrä muodostaa haluttu yhdiste kasvaa eksponentiaalisesti.

Tästä johtuen tarve prosessia yleistäville koneoppimismalleille on iso.

4.3 Kehitettyjä apuvälineitä

Seuraavissa luvuissa avataan kahden kehitetyn koneoppimismallin toimintaa. Nämä mallit on toteutettu suorittaman retrosynteesianalyysiä, mutta niiden toimintaperiaate perustuu eri lähestymistapoihin.

3N-MCTS -malli on kokonaisempi malli, koska se palauttaa retrosyntetisointipolun, joka kertoo, kuinka haluttu yhdiste voidaan syntetisoida laboratoriossa. Se suodattaa jokaisessa retrosyntetisointivaiheessa kaikista mahdollisista muunnoksista vain ne, jotka ovat toteutettavissa. Monte Carlo -puuhakualgoritmi etsii näiden suodatettujen muunnosten avulla retrosyntetisointipolun.

ICHO -malli palauttaa parhaan muunnossäännön, jonka avulla haluttu yhdiste voidaan muodostaa. ICHO -mallin toiminta on verrattavissa 3N-MCTS -mallissa käytettyihin koneoppimismalleihin, jotka yhdessä suodattavat mahdollisista muunnoksista vain parhaimmat, joiden avulla yhdiste voidaan muodostaa. ICHO -malli käyttää sekä samanalaista reaktiodataa kuin 3N-MCTS, mutta se myös hyödyntää ammattikemistien luomia heuristisia sääntöjä.

4.3.1 3N-MCTS -malli

3N-MCTS on koneoppimismalli, jonka on kehittänyt Marwin et al. tutkimusryhmä. Se etsii retrosynteesipolkuja yksinkertaisempiin ja saatavilla oleviin lähtöaineisiin [20]. Kun retrosynteesipolku on varmennettu ja todettu toimivaksi, niin syötteenä annettu yhdiste on mahdollista syntetisoida laboratoriossa.

3N-MCTS -malli muodostuu kolmesta eri koneoppimismallista ja Monte Carlo -puuhakualgoritmista (Monte Carlo Tree Search, MCTS). Neuroverkot on koulutettu avustamaan puuhakualgoritmia etenemäänärkevimpään suuntaan, kun hakualgoritmi etsii syntetisointipolkuja ja tarkistamaan, onko ehdotettu reaktio mahdollinen kyseiselle molekyylille.

Neuroverkot ovat hakupuun laajentumisen suuntaa ohjaava verkko (Expansion Policy Network, EPN), MCTS:än rollout toimintoa tukeva Rollout -verkko (Rollout Policy Network, RPN) ja verkko, joka tarkistaa, onko syntetisointipolku toteutettavissa (In-scope Filter Network, IFN).

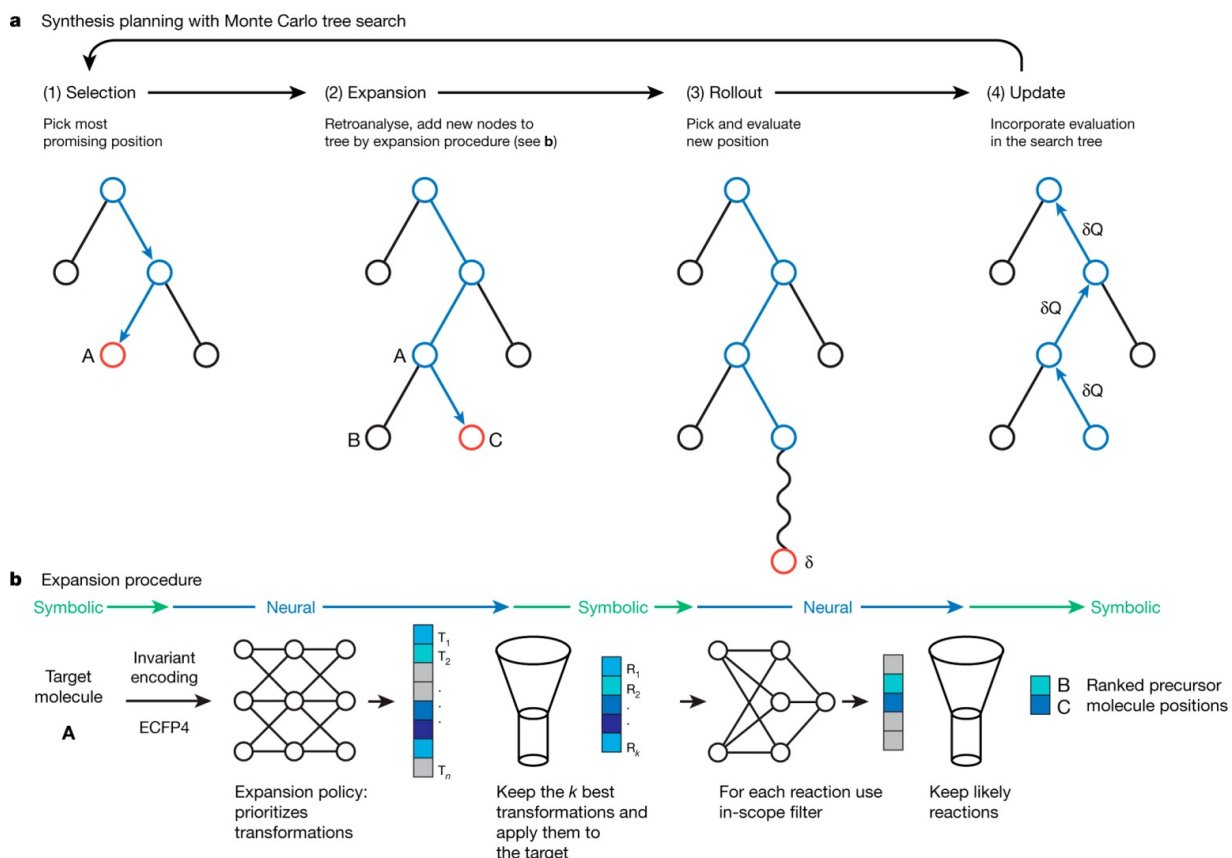
Data, jolla neuroverkot koulutetaan, on peräisin Reaxys -tietokannasta. Reaxyksen omistaa Elsevier kustantamo. Reaxys -tietokannan sisältämä data koostuu muunnossäännöistä (Transformation Rule), jotka kertovat, mitkä lähtöaineet reagoivat keskenään, mikä reaktio on kysessä ja mikä on reaktion lopputuote. Näitä sääntöjä käytetään mallien kouluttamisessa. Tietokanta sisältää yli 12.4 miljoonaa sääntöä. Mallien kouluttamisessa käytetyt säännöt sisältävät vain yksivaiheisia kemiallisia reaktioita ja reaktiossa on mukana vain yhdestä kolmeen lähtötuotetta. Eri mallien kouluttamiseen käytetään tietokannasta eri kriteerein suodatettua dataa. Kaikki koneoppimismallit koulutetaan käyttämällä ADAM -optimoijaa. ADAM -optimoija on algoritmi, jota käytetään neuroverkon kaarien painojen optimoinnissa neuroverkon koulutusvaiheessa. ADAM -optimoija perustuu stokastiseen gradienttiin, mutta koska algoritmi on helppo toteuttaa ja se on laskennallisesti tehokas, niin se soveltuu neuroverkkojen optimointiin, jotka sisältävät paljon kaaria. ADAM -optimoija hyödyntää AdaGrad:in ja RMSProp:in toiminnallisuutta [13].

3N-MCTS toimintamalli

3N-MCTS -mallin IFN ja EPN on kehitetty toimimaan yhdessä. Tutkittaessa puun tilaa S_i jokainen molekyyli syötetään EPN:älle ja se tulostaa, mitkä säännöt voivat muodostaa annetun yhdisteen. Tästä säännöstä saadaan selville, mitkä lähtöaineet voivat muodostaa annetun yhdisteen. Nämä säännöt syötetään IFN:älle, joka suodattaa valituista reaktioista toteutettavissa olevat vaihtoehdot. 3N-MCTS -mallin suoritusjärjestys on kuvattu kuvassa 4.2. Tätä toiminnallisuutta iteroidaan mallissa seuraavassa järjestyksessä.

(1) Ensimmäisessä vaiheessa algoritmi valitsee seuraavan lupaavimman tilan puusta kunnes puun lehti on saavutettu. Jos lehdessä käydään ensimmäisen kerran valintavaiheen aikana, niin lehti arvostellaan simuloimalla hakualgoritmia d askelta eteenpäin samalla muodostaen synteesipolkua (Rollout). Jos lehdessä käydään useamman kuin yhden kerran valintavaiheen aikana, niin mahdolliset säännöt, jotka muodostavat lehden, tutkitaan ja lisätään lehden lapsiksi (Expansion).

(2) Toisessa vaiheessa lupaavien tilojen lapset tutkitaan. Tällöin etsitään lupaavimmat säännöt, jotka muodostavat kyseisessä tilassa olevan yhdisteen.



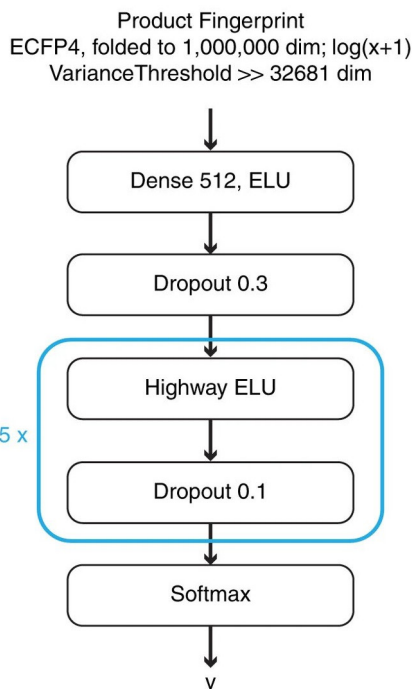
Kuva 4.2: 3N-MCTS:än toimintaperiaate kuvattuna

[20]

(3) Kolmannessa vaiheessa tarkistetaan lehden tila. Jos lehti on ‘todistettusti toimiva’, niin algoritmi palauttaa luvun, mikä on suurempi kuin yksi, jolloin lehteä suositellaan käytettävän synteesipolussa. Muussa tapauksessa lehdelle suoritetaan rollout, jolloin RPN antaa rekursiivisesti uusia sääntöjä niin kauan, kunnes lehti on pilkottu lähtöaineisiin tai kunnes suurin sallittu syvyys d on saavutettu.

(4) Viimeisessä vaiheessa lehtien arvot päivitetään. Jos lähtöaineet löydetään rolloutin aikana, niin lehti saa palkinnoksi arvon 1. Jos kaikkia lähtöaineita ei löydetty, annetaan lehdelle osittainen palkinto. Lehti saa arvon -1, jos yhtään lähtöainetta ei löytynyt.

Saattaa kuitenkin olla, että synteesipolkua ei voida luoda. Joko synteesipolun tutkimiseen menee liian kauan aikaa tai synteesipolku sisältää liian monta vaihetta yhdisteen syntetisoimiseen.

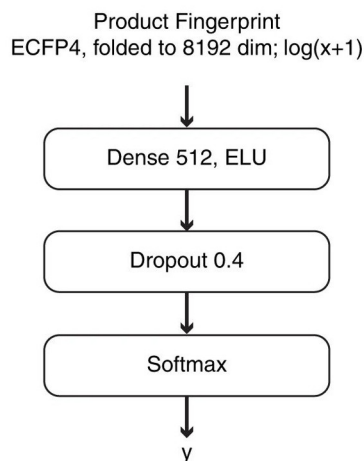
Expansion Policy**Kuva 4.3:** Kuvassa expansion policy -neuroverkon rakenne.

[20]

EPN -verkko

EPN muodostuu täysin kytketystä tasosta ja viidestä highway -tasosta. Molemmissa tasossa on käytetty aktivointifunktiona ELU -funktiota (Exponential Linear Unit). Täysin kytketyssä tasossa suoritetaan dropout -toiminto, jonka jälkeen vain 30 prosenttia ensimmäisen tason solmujen tuloksista otetaan huomioon seuraavalla tasolla. Dropout -toiminolla tarkoitetaan, että vain tietty osa kaikista edellisen tason palauttamista arvoista huomioidaan [12]. Dropout -toiminto suoritetaan mallin koulutusvaiheessa ja pois jätettävät arvot valitaan sattumanvaraisesti. Dropout -toiminnolla pyritään välttämään ylitulointia (Overfitting). Tämän jälkeen jokaisen highway -tason jälkeen suoritetaan dropout -toiminto, jossa otetaan huomioon vain 10 prosenttia edellisen tason arvoista. EPN -verkon viimeinen taso on softmax -taso, joka palauttaa todennäköisyysjakauman. Tämä jakauma kertoo kuinka todennäköisesti muunnossääntö a muodostaa tuotteen m .

EPN -verkon kouluttamiseen valittiin datasta vain reaktiokeskus. Datasta suodatettiin pois sellaiset säännöt, jotka ilmenivät datassa alle kolme kertaa ennen vuotta 2015. Lopullinen reaktiomäärä, jolla EPN koulutettiin, oli 301 671 reaktiota.

RPN -verkko**Rollout Policy**

Kuva 4.4: Kuvassa Rollout policy -neuroverkon rakenne.

[20]

RPN muodostuu yhdestä täysin kytketystä tasosta. Samalla tavalla kuin EPN, RPN käyttää myös ELU -funktioita aktivointifunktiona. RPN -tasossa suoritetaan dropout -toiminto, jonka jälkeen vain 40 prosenttia kaikista ensimmäisen tason arvoista tulkitaan. RPN -verkon viimeinen taso on myös softmax -taso, joka palauttaa todennäköisyysjakauman.

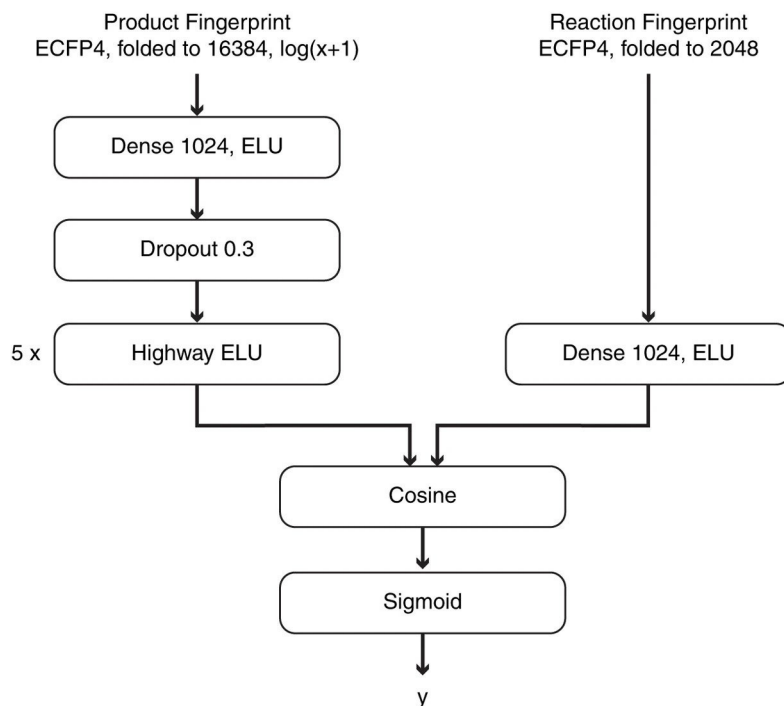
RPN -verkon kouluttamiseen valittiin datasta vain reaktiossa muuttuneet atomit ja liitokset (Reaktiokeskus) ja lähimmät vierekkäiset atomit. Datasta suodatettiin pois sellaiset säännöt, jotka ilmaantuivat alle 50 kertaa ennen vuotta 2015. Lopullinen reaktiomäärä, jolla RPN koulutettiin, oli 17 134 reaktiota.

IFN -verkko

3N-MCTS -mallissa IFN toimii binääriluokittelijana. IFN -verkko kertoo, onko sille annettu yhdiste mahdollista muodostaa annetulla muunnossäännöllä.

IFN -neuroverkko muodostuu kahdesta haarasta. Toinen haara saa syötteenä muodostettavan yhdisteen tunnisteen ja toinen haara saa muunnossäännön tunnisteen.

Haara, joka saa syötteenä yhdisteen tunnisteen, muodostuu yhdestä täysin kytketystä tasosta jonka jälkeen suoritetaan 30 prosentin dropout -toiminto. Tämän jälkeen data kulkee viiden highway -tason läpi. Sekä täysin kytketty että highway -tasot käyttävät

In Scope Filter**Kuva 4.5:** Kuvassa In scope filter -neuroverkon rakenne.

[20]

ELU -funktioita aktivointifunktiona.

Haara, joka saa syötteenä muunnossäännön tunnisteiden, muodostuu vain yhdestä täysin kytketystä tasosta. Tämä taso käyttää myös ELU -aktivointifunktiota.

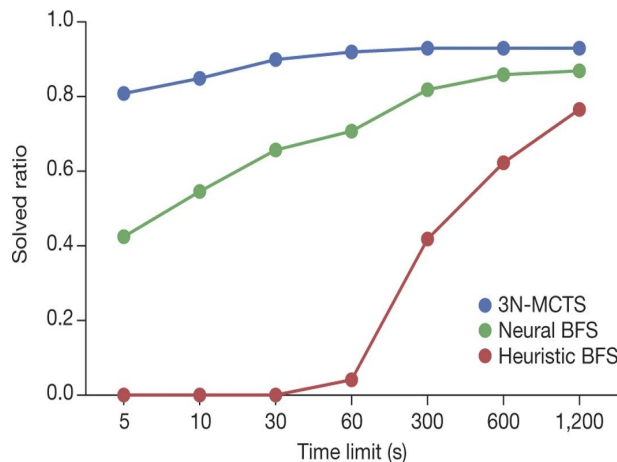
Näiden haarojen jälkeen saaduista tuloksista lasketaan niiden kosinismankaltaisuus. Saatua tulos syötetään sitten sigmoid -funktiolle, jonka tulos kertoo, pystyykö kyseinen muunnossääntö muodostamaan halutun yhdisteen.

IFN -verkko koulutetaan sekä onnistuneiden että epäonnistuneiden sääntöjen avulla. Koska epäonnistuneita sääntöjä ei tallenneta tietokantaan, niin kyseinen data generoidaan. Data generoidaan niin, että jos reaktiossa



lähtöaineet A ja B muodostavat reaktiossa lopputuotteen C, niin lopputuotteita D, E, F, jne. ei muodostu näistä. IFN -verkon kouluttamista varten luotiin 100 miljoonaa epäonnistunutta reaktiota ja 10 miljoonaa epäonnistunutta reaktiota luotiin testaamista varten.

3N-MCTS -mallin suorituskky



Kuva 4.6: 3N-MCTS:än suorituskky verrattuna $SMILES^{3/2}$ -menetelmään hyödyntävään BFS -menetelmään ja 3N-MCTS:än neuroverkkoja hyödyntävään BFS -menetelmään. Kuvaaja kertoo, kuinka monelle prosentille annetuista yhdisteistä menetelmä kykenee löytämään retrosyntetisointipolun annetussa ajassa. Annetulla ajalla tarkoitetaan aikaa, jonka menetelmä voi korkeintaan käyttää yhden yhdisteen retrosyntetisointipolun löytämiseen.

[20]

3N-MCTS -mallin suorituskkyä verrattiin BFS -menetelmään (Best-First Search -Method), joka käyttää heuristista käsinkoodattua $SMILES^{3/2}$ -kustannefunktioita 4.6. Tämä funktio antaa parhaimman tuloksen muunnossäännölle, joka jakaa halutun yhdisteen mahdollisimman hyvin yhtä suuriin lähtötuotteisiin. Tästä menetelmästä luotiin myös versio, joka käyttää muokattua $SMILES^{3/2}$ -kustannefunktioita. Muokattu funktio käyttää 3N-MCTS -mallin EPN -ja IFN -neuroverkkoja parhaimman muunnossäännön selvittämiseksi.

Mallit koulutettiin samoilla muunnossäännöillä ja testeissä käytettiin samaa aikaan perustuvaa jakoa. Mallit koulutettiin muunnossäännöillä, jotka ilmenivät ennen vuotta 2015, ja malleja testattiin vuoden 2015 jälkeen ilmenneillä muunnossäännöillä.

Mallien tehtäväksi annettiin löytää halutulle yhdisteelle retrosyntetisointipolku annetussa ajassa. Testaamiseen käytettiin 497 eri yhdistettä.

3N-MCTS -malli suoriutui kahta muuta algoritmia paremmin jokaisessa eri aikakategoriassa. 3N-MCTS -malli löysi viidessä sekunnissa 80 prosentille yhdisteistä retrosyntetisointipolun, kun taas muokattu $SMILES^{3/2}$ -menetelmä löysi retrosyntetisointipolun vain 40:lle prosentille yhdisteistä ja alkuperäinen ei löytänyt polkuja lainkaan. Kun aikaa kasvatettiin 60 sekuntiin per yhdiste, 3N-MCTS -malli löysi 92 prosentille yhdisteistä polun, muokattu $SMILES^{3/2}$ löysi 71 prosentille yhdisteistä ja alkuperäinen versio vain

4 prosentille yhdisteistä polun.

3N-MCTS -mallia testattiin myös sokkotestillä 45 yliopistotason kemistille. Testin tarkoituksena oli selvittää, suosivatko henkilöt ennemmin kirjallisuuteen perustuvaa retrosyntetisointipolkua vai 3N-MCTS -mallin antamaa polkua. Sokkotestissä 57 prosenttia henkilöistä valitsi 3N-MCTS -mallin muodostaman polun ja loput valitsivat kirjallisuuteen perustuvan polun.

4.3.2 ICHO -malli

On myös yleistä, että retrosynteesipolkuja suunnitelevat mallit hyödyntävät heurustiikka [3]. Badowski et al. ryhmä on kehittänyt koneoppimismallin, joka ehdottaa tietylle yhdisteelle parasta mahdollista muunnossääntöä, jonka avulla yhdiste voidaan muodostaa. Kehitetty ICHO -malli hyödyntää neuroverkkoja ja ammattikemistien muodostamia heuristisia sääntöjä pisteyttäessään muunnossääntöjä. ICHO-malli on verrattavissa 3N-MCTS -mallin EPN -ja IFN -verkkojen toimintaan.

ICHO -verkon (Instytut Chemii Organicznej) kouluttamisessa käytettiin dataa 1.4 miljoonasta reaktiosta ja niiden lopputuotteista. Data saadaan julkaistuista artikkeleista ja patenteista. Reaktiolta kuitenkin vaaditaan, että ainakin yksi reaktio per lopputuote löytyy myös Chematican Expert-Coded säännöstöstä. Datasta suodatettiin myös pois suojaryhmien säännöt. Chematica on retrosyntetisointia suunnitteleva ohjelmisto ja tietokanta, joka sisältää dataa kemiallisista reaktioista. Chematican omistaa Merck KGaA inc. ja sen nimi nykyään on Synthia.

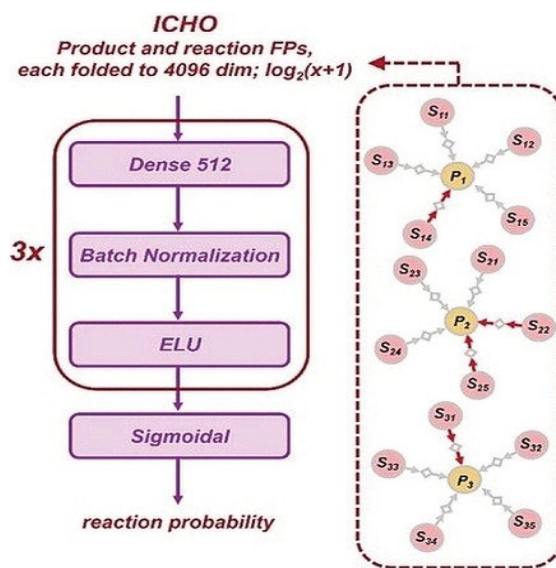
ICHO:sta kehitettiin myös malli, jonka kouluttamisessa käytettiin lisäksi vektoreita, jotka sisälsivät dataa kemiallisesti intuitiivisista reaktiopiirteistä, joita on käytetty aikaisemmissa tutkimuksissa. Tätä mallia kutsutaan ICHO+ malliksi, ja sitä käytetään vertailemaan ICHO- mallin suoriutumista.

Kehitettyssä ICHO -mallissa on kaksi tärkeää asiaa.

(1) Se laskee reaktioiden ilmaantumistodennäköisyyttä uudelleen perustaen kuinka usein ne ilmaantuvat ammattilaisten käyttämissä reaktioissa. Tämä laskenta tapahtuu mallin koulutusvaiheessa. Käytännössä tämä tarkoittaa, että koulutusvaiheessa mallilla on tieto, kuinka monta kertaa tietty reaktiosääntö ilmeni datassa ja kuinka monta kertaa kyseistä reaktiota käytettiin luomaan jokin lopputuote. Malli laskee näiden välisen suhteen, jolloin se pystyy määrittämään, kuinka usein kyseistä reaktiota kannattaa käyttää. Jos esimerkiksi reaktio on mainittu datassa kymmenen kertaa ja samassa datassa kyseistä reaktiota

käytetään kymmenen kertaa luomaan jokin lopputuote, niin malli käyttää tätä reaktiota, koska se luokitellaan ‘Helpoksi ja varmaksi toteuttaa’. Jos suhde taas on pieni, niin reaktio luokitellaan ‘vaikeaksi toteuttaa’.

(2) Toiseksi se kykenee antamaan suuremman kuin nollatodennäköisyyden reaktiolle, joka ei ilmaantunut koulutusvaiheessa. Tämä sen takia, että mallille koulutetaan jatkuva funktio, joka antaa todennäköisyyden jokaiselle reaktiotyypille perustuen, kuinka reaktio muuttaa lopputuotteen lähtötuotteiksi. Eli jos jokin reaktion sormenjälki tai heuristinen kuvaus on samankaltainen toisen reaktion kanssa, joka ilmenee koulutusdatassa, niin reaktiolle voidaan antaa todennäköisyys sen käytölle.



Kuva 4.7: ICHO -mallin rakenne.

[3]

ICHO -malli muodostuu neljästä kerroksesta, joista kolme kerrosta ovat täysin kytkettyjä neuroverkkoja ja viimeinen taso on sigmoid -taso 4.7. Samalla tavalla kuin 3N-MCTS -mallissa, ICHO -malli käyttää myös ELU -aktivointifunktiota. ICHO -malli käyttää Batch normalization -algoritmia koulutusvaiheessa. Batch Normalization -algoritmin idea on standardoida koulutusvaiheessa kerrokselle tuleva syöte, jolla verkon painoja muutetaan.

ICHO -mallia vertailtiin aikasemmin mainitun 3N-MCTS -mallin kanssa, jota kutsutaan artikkelissa SW -mallina. Kyseisestä SW -mallista kehitettiin myös heuristista dataa hyödyntävä malli SW+. Lisäksi luotiin SW -malli, joka ei käy läpi kaikkia mahdollisia reaktioita, vaan valitsee säännöt niistä reaktioista, jotka eivät aiheuta konfliktia ja jotka johtavat vain tiettyyn lopputuotteeseen. Tätä mallia kutsutaan SW2 -malliksi ja tästä luodus-

ta heuristisesta mallista käytetään nimeä SW2+. Lisäksi vertailumalliksi luodaan täysin heuristinen *SMALLER* -malli, joka arvioi reaktioita sen mukaan, kuinka paljon ne rakenteellisesti yksinkertaistavat lopputuotetta. Tämä malli suosii reaktioita, jotka pilkkovat molekyylin keskeltä eli puolittavat sen. Tämä on haluttu lähestymistapa retrosynteesiin, koska se minimoi syntetisointivaiheiden määrän.

ICHO -mallin suorituskyky

ICHO -mallia testataan eri tavalla kuin 3N-MCTS -mallia, jossa mitataan, kuinka monelle yhdisteelle malli kykenee löytämään toimivan retrosyntetisointipolun tietyssä ajassa. ICHO -mallia testataan siten, että jokaiselle mallille annetaan yhdiste, ja malli antaa tulosteena listan muunnossäännöistä, jotka on asetettu paremmuusjärjestykseen niille annetun pisteytyksen mukaan. Muunnossäännön pisteytys kertoo, kuinka paljon malli suosittelee käytettävän kyseistä muunnossääntöä. Näistä muunnossäännöistä laskettiin, kuinka suuri osa niistä pohjautuu johonkin aikaisempaan muunnossääntöön, joka on esiintynyt kirjallisuudessa. Mitä suurempi osa ehdotetuista muunnossäännöistä löytyy kirjallisuudesta, sitä paremmin malli toimii.

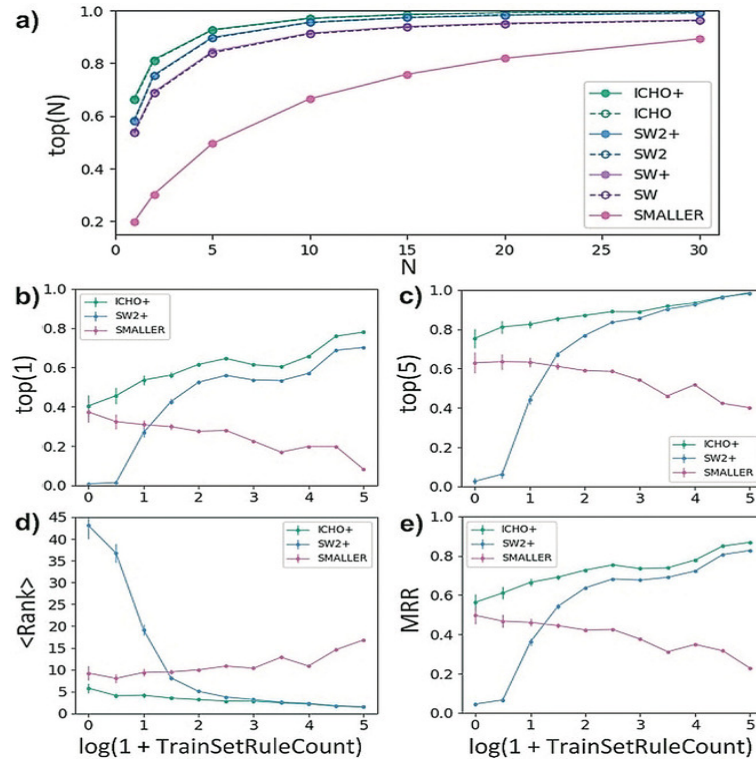
Kuvaajaan 4.8 on piirretty vain eri mallien heuristiikkaa hyödyntävät versiot, koska ne pärjäsivät testeissä ei-heuristisia versioita paremmin. Lisäksi kuvaajaan on piirretty *SMALLER* -malli vertailuksi.

Nähdään, että ICHO+ -malli pisteyttää kirjallisuuteen perustuvat muunnossäännöt paremmin kuin muut mallit, vaikkakin eroavaisuudet ovat vähäiset.

Kuvaajan 4.8 kuvaajista b ja c nähdään, että pienelläkin koulutusaineistolla ICHO+ -malli sijoittaa kirjallisuuden muunnossäännöt korkeammalle kuin muut mallit ja mitä enemmän koulutusaineistoa käytetään mallin kouluttamiseen sitä suurempi osa korkeimmin sijoitetuista muunnossäännöistä perustuvat kirjallisuuteen. Mielenkiintoinen huomio, mikä kuvaajasta nähdään on, että *SMALLER* -malli sijoittaa kirjallisuuteen perustuvat säännöt pienemmällä koulutusaineistolla korkeammalle kuin suuremmalla aineistomäärällä. SW2+ -malli taas toimii päinvastaisesti. Mitä enemmän koulutusaineistoa se saa, sitä korkeammalle se sijoittaa kirjallisuuteen perustuvat säännöt.

Kuvaajat d ja e eivät ole ristiriidassa kuvaajien a, b ja c kanssa. Kuvaajasta d nähdään, että nopeasti koulutusaineiston kasvaessa tutkittavat muunnossäännöt siirtyvät nopeasti korkeille sijoille SW2+ -mallissa ja ICHO+ -malli sijoittaa ne korkealle jo pienellä data-määrällä. *SMALLER* -malli sen sijaan sijoittaa tutkittavat muunnossäännöt koulutusai-

neiston kasvaessa huonommalle sijalle. Kuvaaja e myös näyttää, että ICHO+ ja SW2+-mallien MRR (Mean Reciprocal Rank) -pisteet kasvavat koulutusaineiston kasvaessa, kun taas *SMALLER*-mallin pisteytys laskee. MRR -pisteytyksellä tarkoitetaan kaikkien yhdisteiden sijoitusten käänteislukujen keskiarvoa. Tällöin kuvaajan arvo lähestyy arvoa yksi, jos malli sijoittaa kaikki tutkittavat muunnossäännöt korkeille sijoille.



Kuva 4.8: ICHO -neuroverkon ja muiden testattujen koneoppimismallien suorituskyky kuvattuna. Kuvaajat a-c kertovat, kuinka suuri osa n -määrästä parhaiten pisteytettyjä muunnossääntöjä oli jostain kirjallisuudesta, esim. raportti. Esimerkiksi $\text{top } 1 = 1$ tarkoittaa, että malli antaa parhaimman pisteytyksen aina kirjallisuuteen perustuvalla muunnossäännöllä. Kuvaaja d kertoo, mille sijalle kukin malli sijoittaa tietyn muunnossäännön. Tässä tapauksessa $\text{rank} = 1$ tarkoittaa, että malli sijoittaa tietyn muunnossäännön aina sijalle yksi. Kuvaaja e kertoo keskiarvon kirjallisuuteen perustuvien muunnossääntöjen käänteisestä sijoituksesta. Tämä tarkoittaa, että jos malli sijoittaa kaikkien kirjallisuuteen perustuvat muunnossäännöt ensimmäiseksi, niin MRR saa arvon yksi.

5 Yhteenveto

Tässä tekstissä käytiin läpi yleisiä asioita, mitä liittyy uusien lääkkeiden löytämiseen ja syntetisoinnin suunnitteluun ja kuinka koneoppimista voidaan hyödyntää näissä vaiheissa. Lisäksi tekstissä käytiin läpi kolmen eri koneoppimismallin toimintaperiaatteet.

Johdannossa kerrottiin, mitä vaiheita kuuluu lääkekehitykseen ja miksi koneoppimista halutaan käyttää osana lääkekehityksen prosessia. Tämä asettaa ongelman, miksi koneoppimista halutaan käyttää lääkekehityksen prosesseissa. Vaikkakin käyttökohteita koneoppimiselle on jokaisessa lääkekehityksen vaiheessa, niin tekstissä keskityttiin vain ainostaan lääkkeiden löytämiseen koneoppimismenetelmillä ja lääkkeiden syntetisoinnin suunnitteluun koneoppimismenetelmillä. Nämä käytiin sen takia, että ne ovat kliinisten testien jälkeen ne vaiheet, joihin kuluu eniten resursseja, ja joiden nopeuttaminen voi nopeuttaa myös kliinisiä testejä. Nämä käytiin myös sen takia, koska ongelmat ja konseptit, joita lääkkeiden löytämisessä ja syntetisoinninsuunnittelussa käytiin läpi, on suhteellisen helppoja ymmärtää ilman syvällisempää kemiaan tai lääkkeisiin liittyvää tietoa, jolloin teksti pystyi keskittymään enemmän kehitettyihin koneoppimismalleihin.

Johdannon jälkeen käytiin myös, mitä eri datatyyppejä käytetään tekstissä käytävien koneoppimismallien kouluttamisessa. Tämän kappaleen sisällön käyminen auttaa ymmärtämään, miten koneoppimismallit toimivat ja mitä ongelmaa ne ratkovat.

Myöhemmissä kappaleissa käytiin, mitä asioita ja taustaa liittyy lääkkeiden löytämiseen ja syntetisoinnin suunnitteluun. kappaleissa myös kerrottiin, mitä ongelmia näihin vaiheisiin kuuluu ja miten koneoppimisella voidaan ratkoa näitä ongelmia.

Kappaleissa käytiin läpi myös kolmen koneoppimismallin toimintaperiaatteet, kuinka ne toimivat ja kuinka ne ratkovat aikaisemmin esitettyjä ongelmia. Vaikkain kirjallisuudesta löytyy useita eri koneoppimiselle ratkomaan kappaleissa kerrottuja ongelmia, niin nämä mallit valittiin sen takia, koska ne on toteutettu eri tavalla kuin aikaisemmat mallit hyödyntäen uusimpia koneoppimistekniikoita. Yhdisteiden syntetisointiin keskittyviä koneoppimismalleja käytiin kaksi kappaletta, koska haluttiin nostaa esiin, kuinka syntetisoinninsuunnittelua voidaan suorittaa usealla eri tavalla. lisäksi haluttiin nostaa esiin, kuinka hyvin läpi käydyt koneoppimismallit suoriutuvat tehtävistään verrattaen muihin malleihin.

Lähteet

- [1] *About PubChem*. URL: <https://pubchemdocs.ncbi.nlm.nih.gov/about>.
- [2] A. F. de Almeida, R. Moreira ja T. Rodrigues. "Synthetic organic chemistry driven by artificial intelligence". eng. *Nature reviews. Chemistry* 3.10 (2019), s. 589–604. ISSN: 2397-3358.
- [3] T. Badowski, E. P. Gajewska, K. Molga ja B. A. Grzybowski. "Synergy Between Expert and Machine-Learning Approaches Allows for Improved Retrosynthetic Planning". *Angewandte Chemie International Edition* 59.2 (2020), s. 725–730. DOI: <https://doi.org/10.1002/anie.201912083>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201912083>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201912083>.
- [4] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan ja A. L. Hopkins. "Quantifying the chemical beauty of drugs". eng. *Nature chemistry* 4.2 (2012), s. 90–98. ISSN: 1755-4330.
- [5] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev ja A. Walsh. "Machine learning for molecular and materials science". eng. *Nature (London)* 559.7715 (2018), s. 547–555. ISSN: 0028-0836.
- [6] C. W. Coley, W. H. Green ja K. F. Jensen. "Machine Learning in Computer-Aided Synthesis Planning". eng. *Accounts of chemical research* 51.5 (2018), s. 1281–1289. ISSN: 0001-4842.
- [7] E. J. Corey. "Robert Robinson Lecture. Retrosynthetic thinking—essentials and examples". eng. *Chemical Society reviews* 17 (1988), s. 111–133. ISSN: 0306-0012.
- [8] S. Ekins, A. C. Puhl, K. M. Zorn, T. R. Lane, D. P. Russo, J. J. Klein, A. J. Hickey ja A. M. Clark. "Exploiting machine learning for end-to-end drug discovery and development". eng. *Nature materials* 18.5 (2019), s. 435–441. ISSN: 1476-1122.
- [9] A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magariños, J. P. Overington, G. Papadatos, I. Smit ja A. R. Leach. "The ChEMBL database in 2017". *Nucleic Acids Research* 45.D1 (marraskuu 2016), s. D945–D954. ISSN: 0305-1048. DOI: [10.1093/nar/gkw1074](https://doi.org/10.1093/nar/gkw1074). eprint: [https://](https://doi.org/10.1093/nar/gkw1074)

- academic.oup.com/nar/article-pdf/45/D1/D945/8846762/gkw1074.pdf. URL: <https://doi.org/10.1093/nar/gkw1074>.
- [10] K. M. Gayvert, N. S. Madhukar ja O. Elemento. "A Data-Driven Approach to Predicting Successes and Failures of Clinical Trials". eng. *Cell chemical biology* 23.10 (2016), s. 1294–1301. ISSN: 2451-9456.
- [11] S. Harel ja K. Radinsky. "Accelerating Prototype-Based Drug Discovery Using Conditional Diversity Networks". Teoksessa: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, s. 331–339. ISBN: 9781450355520. DOI: 10.1145/3219819.3219882. URL: <https://doi-org.libproxy.helsinki.fi/10.1145/3219819.3219882>.
- [12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever ja R. R. Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors*. 2012. arXiv: 1207.0580 [cs.NE].
- [13] D. P. Kingma ja J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [14] G. Maggiora, M. Vogt, D. Stumpfe ja J. Bajorath. "Molecular similarity in medicinal chemistry". eng. *Journal of medicinal chemistry* 57.8 (2014), s. 3186–3204. ISSN: 1520-4804.
- [15] B. H. Munos ja W. W. Chin. "How to revive breakthrough innovation in the pharmaceutical industry". eng. *Science translational medicine* 3.89 (2011), s. 89cm16–89cm16. ISSN: 1946-6234.
- [16] D. A. PENSAK ja E. J. COREY. "LHASA—Logic and Heuristics Applied to Synthetic Analysis". Teoksessa: *Computer-Assisted Organic Synthesis*. Luku 1, s. 1–32. DOI: 10.1021/bk-1977-0061.ch001. eprint: <https://pubs.acs.org/doi/pdf/10.1021/bk-1977-0061.ch001>. URL: <https://pubs.acs.org/doi/abs/10.1021/bk-1977-0061.ch001>.
- [17] M. Popova, O. Isayev ja A. Tropsha. "Deep reinforcement learning for de novo drug design". eng. *Science advances* 4.7 (2018), eaap7885–eaap7885. ISSN: 2375-2548.
- [18] D. Rogers ja M. Hahn. "Extended-Connectivity Fingerprints". eng. *Journal of chemical information and modeling* 50.5 (2010), s. 742–754. ISSN: 1549-9596.
- [19] L. Rokach. "Ensemble-based classifiers". eng. *The Artificial intelligence review* 33.1-2 (2009), s. 1–39. ISSN: 0269-2821.

- [20] M. H. S. Segler, M. Preuss ja M. P. Waller. "Planning chemical syntheses with deep neural networks and symbolic AI". eng. *Nature (London)* 555.7698 (2018), s. 604–610. ISSN: 0028-0836.
- [21] B. Shin, S. Park, J. Bak ja J. C. Ho. "Controlled Molecule Generator for Optimizing Multiple Chemical Properties". Teoksessa: *Proceedings of the Conference on Health, Inference, and Learning*. CHIL '21. Virtual Event, USA: Association for Computing Machinery, 2021, s. 146–153. ISBN: 9781450383592. DOI: 10.1145/3450439.3451879. URL: <https://doi.org/10.1145/3450439.3451879>.
- [22] C. Sotriffer, R. Mannhold, H. Kubinyi ja G. Folkers. *Virtual Screening: Principles, Challenges, and Practical Guidelines*. eng. Vol. 48. Methods and principles in medicinal chemistry. Weinheim: John Wiley ja Sons, Incorporated, 2011. ISBN: 9783527326365.
- [23] T. Sterling ja J. J. Irwin. "ZINC 15 - Ligand Discovery for Everyone". *Journal of Chemical Information and Modeling* 55.11 (2015). PMID: 26479676, s. 2324–2337. DOI: 10.1021/acs.jcim.5b00559. eprint: <https://doi.org/10.1021/acs.jcim.5b00559>. URL: <https://doi.org/10.1021/acs.jcim.5b00559>.
- [24] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer ja S. Zhao. "Applications of machine learning in drug discovery and development". eng. *Nature reviews. Drug discovery* 18.6 (2019), s. 463–477. ISSN: 1474-1776.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser ja I. Polosukhin. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].
- [26] D. Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". *Journal of Chemical Information and Computer Sciences* 28.1 (1988), s. 31–36. DOI: 10.1021/ci00057a005. eprint: <https://pubs.acs.org/doi/pdf/10.1021/ci00057a005>. URL: <https://pubs.acs.org/doi/abs/10.1021/ci00057a005>.

