

**Statistilise metsainventuuri (SMI) ja
maakasutuse, maakasutuse muutuse ja metsan-
duse (LULUCF)
andmehõive ja metoodika
arendamine**

*Metoodika väljatöötamine
metsamaa maakategooria aktuaalsete
piiride kaardi loomiseks (metsamask)
metsamaa pindala ja selle muutumise edasiseks hin-
damiseks*

1 Sissejuhatus: metsamaskid ja mitme definitsiooni väljakutse kaugseires

Kaugseire on kujunenud asendamatuks vahendiks metsaressursside hindamisel ja seires, pakkudes ajakohast ja laiaulatuslikku teavet metsade seisundi ning dünaamika kohta. Üks keskseid tooteid metsanduslikus kaugseires on metsamask (inglise keeles Forest Mask, FM). Metsamask on ruumiline andmekiht, tavaliselt kas raster- või vektorformaadis, mis määratleb metsamaa geograafilise ulatuse, eristades metsa mittemetsaaladest. See on vundamendiks mitmesugustele metsanduslikele analüüsidele, sealhulgas metsa pindala hindamisele, metsade juurdekasvu ja tagavara kalkuleerimisele, raietegevuse ja metsakahjustuste (nt tormikahjud, üraskirüüsted) monitooringule ning elurikkuse uuringutele. Metsamaski täpsus ja kvaliteet on seega kriitilise tähtsusega, kuna see mõjutab otseselt kõikide järgnevate analüüside usaldusväärsust. Näiteks kasvava metsa tagavara hindamisel on täpne metsamask hädavajalik, et välistada analüüsist mittemetsa pikslid ja vältida tulemuste moonutamist. Traditsiooniliselt on metsamaske koostatud ortofotode käsitsi interpreteerimise teel, kuid üha enam kasutatakse selleks satelliitpiltide (nii optiliste kui ka radarandmete) automaatset või poolautomaatset klassifitseerimist, rakendades juhendatud või juhendamata meetodeid. Viimastel aastakümnetel on lisandunud ka laserskaneerimise (LiDAR ehk *Light Detection and Ranging*; ALS – *Airborne Laser Scanning*) andmete kasutamine, mis pakuvad eriti väärtuslikku informatsiooni metsa vertikaalse struktuuri kohta.

Metsamaski tähtsus metsamajanduses ja keskkonnaseires on pidevalt kasvanud. Üha suurenev huvi keskkonnakaitse ja metsade jätkusuutliku majandamise vastu on toonud kaasa vajaduse täpsemate ja usaldusväärsemate meetodite järele metsamuutuste kaardistamisel ja hindamisel. Kaugseire pakub siin efektiivseid lahendusi, võimaldades teavet koguda suurtelt aladelt ja regulaarselt, ilma et oleks vaja ulatuslikke ja kulukaid välitöid igas konkreetses kohas. Metsamaskid on seega väärtuslikud tööriistad metsa omaduste, nagu biomassi akumulatsioon, raadamine, metsa tervislik seisund ja looduslike häiringute ulatus, jälgimiseks. Ka Eestis on digitaalsed lahendused metsamajanduses üha laiemalt kasutusel, lihtsustades ja kiirendades tööprotsesse ning andmeanalüüsi.

Kaugseirepõhiste metsamaskide loomisel kerkib aga esile fundamentaalne probleem: metsa definitsioon ei ole universaalne. Nii siseriiklikul kui ka rahvusvahelisel tasandil eksisteerib mitmeid erinevaid metsa definitsioone, mis varieeruvad peamiste kvantitatiivsete ja kvalitatiivsete parameetrite poolest, nagu minimaalne pindala, puude minimaalne kõrgus, võrastiku minimaalne liitusprotsent ning maakasutuslikud piirangud. See definitsioonide mitmekesisus tekitab olulisi väljakutseid metsa pindala ja muude metsaressurssi iseloomustavate näitajate võrreldavusele erinevate riikide ja organisatsioonide vahel. Samuti mõjutab see

aruandluskohustuste täitmist, näiteks ÜRO Kliimamuutuste Raamkonventsiooni (UNFCCC) või ÜRO Toidu- ja Põllumajandusorganisatsiooni (FAO) suunal. Kui metsamask on loodud ühe konkreetse, jäiga definitsiooni alusel, on selle kasutusala piiratud ning see ei pruugi sobida analüüsideks, mis nõuavad teistsuguste kriteeriumide rakendamist. Seetõttu on üha olulisem arendada metoodikaid, mis võimaldaksid luua paindlikke metsamaske, mida saab kohandada vastavalt erinevatele definitsioonidele. Käesoleva ülevaate peamine eesmärk ongi kirjeldada, kuidas koostada kaugseire andmete põhjal metsamaske, mis suudaksid anda hinnanguid metsa kohta vastavalt mitmesugustele definitsioonidele, pöörates erilist tähelepanu Eestis kasutatavatele ja rahvusvaheliselt tunnustatud lähenemistele. Vajadus selliste paindlike lahenduste järele tuleneb otseselt nii rahvusvahelistest kui ka siseriiklikest aruandluskohustustest ning metsanduspoliitika eesmärkidest, mis võivad tugineda erinevatele metsa määratlustele. Ühtse, jäiga metsamaski kasutamine võib viia metsaressursside ala- või ülehindamiseni sõltuvalt kontekstist, mõjutades seeläbi poliitilisi otsuseid, maakasutuse planeerimist ja rahvusvaheliste kohustuste korrektset täitmist. Paindlikkus metsamaskide loomisel ja rakendamisel võimaldab ühtlustatud lähenemist erinevate nõuete täitmisel, kasutades selleks ühtset ja kvaliteetset kaugseireandmestikku.

2 Metsa definitsioonide mõistmine: parameetrid ja mõjud

Metsa defineerimine on metsandusliku inventuuri ja seire alustala. Erinevad organisatsioonid ja riigid kasutavad metsa määratlemiseks mitmesuguseid parameetreid, mis mõjutavad oluliselt metsa pindala ja teiste metsaressurssi iseloomustavate näitajate hinnanguid. Paindliku metsamaski loomiseks on esmalt vaja mõista neid defineerivaid parameetreid ja nende varieeruvust.

2.1 Peamised defineerivad parameetrid

Metsa definitsioonid põhinevad tavaliselt järgmistel kvantitatiivsetel ja kvalitatiivsetel kriteeriumitel:

- **Minimaalne pindala (ha):** See on alampiir, millest suuremat puittaimestikuga kaetud maa-ala loetakse metsaks. See parameeter aitab välistada väga väikesed puude grupid või üksikud puud.
- **Minimaalne puude kõrgus (m):** Määrab alampiiri puude kõrgusele, mida antud alal arvestatakse metsa moodustavate puudena. See aitab eristada metsa madalamakasvulisest põõsastikust või noorendikest, mis ei ole veel saavutanud metsa tunnuseid.
- **Minimaalne võrastiku liitus/katvus (%):** See on protsentuaalne näitaja, mis kirjeldab, kui suurt osa maapinnast peavad puuvõrad katma, et ala kvalifitseeruks metsaks. See parameeter iseloomustab puistu tihedust.
- **Maakasutus:** Paljud metsa definitsioonid välistavad alad, mis on küll puudega kaetud, kuid on peamiselt põllumajanduslikus (nt viljapuuaiad, energiavõsaistandikud põllumaal) või linnalises (nt pargid, kalmistud, kui need ei ole spetsiaalselt metsana määratletud) kasutuses. Mõned definitsioonid, sealhulgas Eesti oma, hõlmavad ka alasid, mida aktiivselt majandatakse puidu tootmise eesmärgil või puittaimkatte säilitamiseks, isegi kui need ajutiselt ei vasta kõrguse või liituse kriteeriumidele (nt raiesmikud, kus oodatakse metsa uuenemist).

2.2 Eesti siseriiklik metsa definitsioon

Eesti Metsaseaduse kohaselt on mets ökosüsteem, mis koosneb metsamaast, sellel kasvavast taimestikust ja seal elunevast loomastikust. Praktikas on metsamaa defineerimisel kasutusel järgmised kriteeriumid:

- Metsamaaks loetakse maa-ala, mille pindala on vähemalt 0,1 hektarit.
- Sellel alal peavad kasvama puittaimed, mille kõrgus on vähemalt 1,3 meetrit.
- Nende puittaimede võrad peavad katma vähemalt 30% sellest maa-alast.
- Lisaks loetakse metsamaaks ka sellist ala, mis ei vasta ajutiselt eelnimetatud kriteeriumidele (nt raiesmik), kuid mida majandatakse puidu tootmise või puittaimkatte säilitamise eesmärgil.

2.3 Peamised rahvusvahelised metsa definitsioonid

Rahvusvahelisel tasandil on mitmeid olulisi metsa definitsioone, millest kaks peamist on FAO ja UNFCCC poolt kasutatavad.

- **FAO (ÜRO Toidu- ja Põllumajandusorganisatsioon):** FAO definitsiooni kohaselt on mets maa-ala, mis on suurem kui 0,5 hektarit, kus kasvavad puud, mille kõrgus on (või potentsiaal saavutada) vähemalt 5 meetrit *in situ*, ning puuvõrade liitus (või ekvivalentne täius) on üle 10%. See definitsioon ei hõlma maad, mis on valdavalt põllumajanduslikus või linnalises kasutuses. Oluline on märkida, et FAO definitsioon hõlmab ka noori looduslikke puistuid ja kõiki rajatud metsakultuure, mis ei ole veel saavutanud nõutud tihedust või kõrgust, kuid millelt oodatakse nende saavutamist. Samuti kuuluvad siia ajutiselt puudeta alad (nt raie või looduslike põhjuste tõttu), millelt oodatakse looduslikku või kunstlikku taastumist metsaks.
- **UNFCCC (ÜRO kliimamuutuste raamkonventsioon):** UNFCCC lubab Kyoto protokolli lisas I nimetatud riikidel (Annex I countries) valida oma metsa definitsiooni parameetrid teatud kindlaksmääratud vahemikes:
 - Minimaalne pindala: 0,05–1,0 hektarit.
 - Minimaalne puude kõrgus *in situ*: 2–5 meetrit.
 - Minimaalne võrastiku liitus: 10–30 protsenti. Lisaks ei tohi maa olla peamiselt põllumajanduslikus või muus mittemetsanduslikus kasutuses. Ka ajutiselt raadatud alad, millelt oodatakse metsa taastumist, loetakse metsa hulka.
- **FRA (Forest Resources Assessment) definitsioonid:** FAO globaalse metsaressurside hindamise (FRA) raames kasutatakse spetsiifilisi kriteeriume. Statistilise Metsainventuur SMI on alates 2005. aastast kasutanud FRA metsa ja muu metsamaa (OWL – Other Wooded Land) klassifikatsioone paralleelselt siseriikliku definitsiooniga, et tagada andmete rahvusvaheline võrreldavus. FRA 2020 aruandluses Eesti kohta ei kasutatud metsa pindala hindamisel enam riiklikke definitsioone, vaid lähtuti FRA kriteeriumidest.

2.4 Definitsiooni valiku mõju metsa pindalale ja omadustele

Erinevate parameetrite (pindala, kõrgus, liitus) künnisväärtuste valik toob kaasa märkimisväärsed erinevusi hinnatavas metsa pindalas ja seega ka teistes metsaressurssi iseloomustavates näitajates. Näiteks Eesti riikliku metsa definitsiooni (minimaalne kõrgus 1,3 m, minimaalne pindala 0,1 ha, minimaalne liitus 30%) võrdlemisel FAO definitsiooniga (minimaalne kõrgus 5 m, minimaalne pindala 0,5 ha, minimaalne liitus 10%) on ilmne, et Eesti definitsioon on mitmes aspektis rangem. See tähendab, et sama kaugseireandmestiku põhjal klassifitseeritakse FAO definitsiooni järgi metsaks tõenäoliselt ka nooremad ja üksikud puud, mis Eesti definitsiooni järgi veel metsaks ei kvalifitseeru. Sellest tulenevalt võib FAO definitsiooni kohane metsa pindala olla suurem kui Eesti definitsiooni kohane. Need erinevused mõjutavad otseselt metsamajanduslikke otsuseid, maakasutuspoliitikat ja rahvusvahelist aruandlust.

Maakasutuse kriteeriumid on samuti olulised. Nende eesmärk on eristada metsa teistest puittaimestikuga kaetud aladest, mis ei täida metsa ökoloogilisi või majanduslikke

funktsioone samal viisil, näiteks viljapuuaiad, energiavõsaistandikud põllumaal või pargid linnades (kui need ei ole spetsiaalselt metsana määratletud). Maakasutuse aspekti arvestamine nõuab sageli kaugseireandmete kombineerimist teiste andmekihtidega, nagu maakasutuskaardid või katastriandmed, et metsamaski korrektselt piiritleda. Ilma sellise integratsioonita võivad kaugseireandmed üksi (nt ainult spektraalinfo või kõrgusmudel) viia valeklassifitseerimiseni, kus näiteks suur viljapuuaed loetakse metsaks.

Järgnev tabel annab ülevaate peamiste metsa definitsioonide parameetritest.

Tabel 1. Peamiste metsa definitsioonide parameetrite võrdlus

Definitsiooni Allikas	Minimaalne Pindala (ha)	Minimaalne Puude Kõrgus (m)	Minimaalne Võrastiku Lii-tus (%)
Eesti Riiklik (Metsaseadus)	0,1	1,3	30
FAO (ÜRO Toidu- ja Põllumaj.)	0,5	5	10
UNFCCC (ÜRO Kliimamuutuste Raamk.)	0,05–1,0	2–5	10–30
FRA (kasutatud Eesti SMI poolt alates 2005)	Sarnane FAO-le	Sarnane FAO-le	Sarnane FAO-le

See tabel illustreerib selgelt, kui suures ulatuses võivad metsa definitsioonid varieeruda. Paindliku metsamaski loomise võimekus, mis suudab neid erinevaid künniseid ja tingimusi arvesse võtta, on seega äärmiselt oluline.

3 Kaugseire andmeallikad metsamaski koostamiseks

Metsamaski koostamiseks kasutatakse mitmesuguseid kaugseireandmeid, millest igaühel on oma spetsiifilised omadused, eelised ja puudused. Optimaalse tulemuse saavutamiseks kombineeritakse sageli erinevat tüüpi andmeid.

3.1 Optilised satelliitandmed (nt Sentinel-2, Landsat)

Optilised sensorid mõõdavad Päikeselt peegeldunud kiirgust Maa pinnalt erinevates elektromagnetilise spektri lainepikkuste vahemikes. Levinumad optilised satelliidid metsanduslikes rakendustes on Euroopa Kosmoseagentuuri (ESA) Sentinel-2 ja USA Geoloogiateenistuse (USGS) Landsat seeria satelliidid. Sentinel-2 pakub multispektraalseid andmeid ruumilise lahutusega 10, 20 ja 60 meetrit ja teoreetilise korduskülastusajaga umbes 5 päeva ekvaatoril (tihedam kõrgematel laiuskraadidel). Landsati andmeseeria ulatub tagasi 1970. aastatesse, pakkudes väärtuslikku ajalist perspektiivi metsa muutuste uurimiseks.

- **Eelised:** Optilised andmed on laialdaselt kättesaadavad, paljud neist (sh Sentinel-2 ja Landsat) tasuta. Need sobivad hästi vegetatsiooniindeksite (nt NDVI, EVI) arvutamiseks, mis on tundlikud taimkatte aktiivsusele ja biomassile. Multispektraal- ja eriti hüperspektraalandmed võimaldavad teatud määral eristada ka puuliike ning hinnata metsa tervislikku seisundit ja fenoloogilisi muutusi.
- **Puudused:** Optiliste sensorite peamine piirang on nende tundlikkus ilmastikutingimustele, eriti pilvkattete ja atmosfääri aerosoolidele. Pilved ja nende varjud takistavad Maa pinna vaatlemist, mis vähendab andmete tegelikku ajalist kättesaadavust ja võib tekitada lünki aegridades. Optilised andmed annavad peamiselt informatsiooni metsa horisontaalse struktuuri ja võrastiku pealmise kihi kohta. Tiheda võrastiku korral võib signaal küllastuda, mistõttu on raskem saada teavet metsa vertikaalse struktuuri või alusmetsa kohta.
- **Eeltöötlus:** Enne kasutamist vajavad optilised satelliidiandmed mitmeid eeltöötamise etappe, sealhulgas geomeetriline korrektsioon (georeferentseerimine), radiomeetriline kalibratsioon (digitaalsete väärtuste teisendamine füüsikalisteks kiirgustiheduse väärtusteks), atmosfäärikorrektsioon (atmosfääri mõjude eemaldamine, et saada pinnapeegelduse väärtused) ja mõnikord ka topograafiline korrektsioon (maastiku reljeefist tingitud valgustustingimuste varieeruvuse vähendamiseks). Oluline on ka pilvede, pilvevarjude ja muude artefaktide korrektne maskeerimine.

3.2 Aerolaserskanneri (ALS / LiDAR) andmed

ALS on aktiivne kaugseire tehnoloogia, mis kasutab laservalgust kauguste mõõtmiseks. Süsteem saadab välja laserimpulsse ja mõõdab nende tagasipeegeldumise aega ja intensiivsust, luues Maa pinnast ja sellel asuvatest objektidest (sh vegetatsioonist) tiheda kolmemõõtmelise (3D) punktipilve. Metsanduses kasutatakse peamiselt lennukitelt (ALS) või droonidelt kogutud LiDAR-andmeid.

- **Eelised:** LiDAR võimaldab erakordselt täpselt mõõta puude kõrgust, võrastiku

vertikaalset struktuuri, üksikute puude võrade kuju ja maapinna reljeefi ka tiheda taimkatte all. Sellest saab tuletada ülitäpseid digitaalseid maapinnamudeleid (DTM – Digital Terrain Model) ja digitaalseid pindmudeleid (DSM – Digital Surface Model). Nende kahe mudeli vahe annab võrastiku kõrgusmudeli (CHM – Canopy Height Model), mis näitab vegetatsiooni kõrgust maapinnast. LiDAR on vähem tundlik ilmastikutingimustele kui optilised sensorid (kuigi tugev vihm või udu võivad mõjutada) ja suudab tänu laserkiire laiale jäljele saada võrastikust mitu peegeldust.

- **Puudused:** LiDAR-andmete kogumine on tavaliselt kulukam ja ajamahukam kui satelliitkaugseire, eriti kui on vaja katta suuri alasid. Andmemahud on samuti väga suured, nõudes spetsiifilist tarkvara ja arvutusvõimsust töötlemiseks.
- **Eeltöötlus:** LiDAR-punktipilvede eeltöötlus hõlmab sageli müra (valepunktide, nt lindude või atmosfääriosakeste peegelduste) eemaldamist, maapinna punktide klassifitseerimist (et eraldada need vegetatsiooni ja hoonete punktidest DTM-i loomiseks) ning punktipilve normaliseerimist (maapinna reljeefi mõju eemaldamine, et saada iga punkti kõrgus maapinnast).

3.3 Andmete fusioon ja integreerimine

Kuna igal andmetüübil on oma tugevused ja nõrkused, siis sageli parima tulemuse metsamaski ja teiste metsanduslike parameetrite hindamisel annab erinevate andmeallikate kombineerimine ehk fusioon. Näiteks LiDAR-andmetest saadud täpne kõrgusinfo (CHM) koos optiliste satelliitandmete (nt Sentinel-2) spektraalse informatsiooniga võib oluliselt parandada metsaalade eristamise ja klassifitseerimise täpsust võrreldes kummagi andmetüübi eraldi kasutamisega. Optilised andmed annavad teavet vegetatsiooni tüübi ja seisundi kohta, samas kui LiDAR täpsustab selle vertikaalset struktuuri. Eestis on SMI andmete integreerimine Sentinel-2 ja ALS andmetega näidanud paljulubavaid tulemusi metsa kasvava tagavara enustamisel, kasutades selleks masinõppe meetodeid.

Andmetüüpide valikul ja kombineerimisel on oluline silmas pidada, et ükski kaugseire andmetüüp ei ole universaalselt parim kõikide metsa definitsiooni parameetrite (kõrgus, liitus, pindala, struktuur) tuletamiseks. Optimaalne lahendus peitub sageli andmete fusioonis, kus kasutatakse ära iga andmetüübi spetsiifilisi tugevusi vastavalt konkreetse definitsiooni nõuetele. Näiteks kui definitsioon rõhutab puude kõrgust, on LiDAR-andmed asendamatud. Kui oluline on eristada erinevaid puuliike või hinnata lehestiku seisundit, on multispektraal- või hüperspektraalandmed väärtuslikud. Ilmastikukindluse ja struktuuriinfo saamiseks tihedates metsades võib SAR olla kasulik. Seega, paindliku metsamaski loomiseks, mis suudab arvestada erinevate definitsioonide parameetritega, tuleb hoolikalt valida ja potentsiaalselt kombineerida erinevaid kaugseire andmeallikaid.

Sõltumata valitud andmeallikast või hilisemast klassifitseerimismeetodist, on andmete eeltötluse kvaliteet kriitilise tähtsusega lõpliku metsamaski täpsusele. Igal andmetüübil on spetsiifilised eeltötluse etapid, mille eesmärk on eemaldada sensori-, atmosfääri- ja

maastikumõjud ning teisendada algandmed füüsikaliselt tähenduslikeks väärtusteks (nt pinnapeegeldus, radari tagasihajumise koefitsient, normaliseeritud punktikõrgused). Vigane või ebapiisav eeltöötlus (nt puudulik atmosfäärikorreksioon optilistel pildidel, ebakorrektne georefereerimine või haavelmüra halb eemaldamine SAR-andmetel) levib vigadena edasi kogu analüüsi ahelas ja viib paratamatult ebatäpse metsamaskini. Näiteks pilvede ja nende varjude korrektne maskeerimine optilistel satelliitpildidel on hädavajalik, et vältida nende alade valeklassifitseerimist. Seega on hoolikas ja korrektne andmete eeltöötlus vundament usaldusväärse metsamaski loomiseks, mis suudab erinevaid definitsioone adekvaatselt hinnata.

4 Andmetöötlus tarkvara

Projekti eesmärk on määratleda alad, mis kvalifitseeruvad metsamaaks. Metsamaa defineerimine on keerukas, kuna see on maakasutuse kategooria, mis ei sõltu ainult puittaimestikust (maakattest), vaid eelkõige maa-ala kasutusotstarbest. Projekti nõuete kohaselt oli vajalik optimeerida mahukate andmete analüüsi ning tagada saadava tulemuse maksimaalne täpsus.

Sellest lähtuvalt viidi läbi sisendandmete, vajaliku tarkvara ja kasutatavate algoritmide põhjalik analüüs. Vaatluse all olid nii Python kui ka tarkvarapakett R. Optimaalse andmetöötluse ja kogu andmetöötlusprotsessi tervikliku lahenduse leidmiseks osutus hetkel parimaks valikuks R.

Töö käigus analüüsiti erinevate protseduuride andmetöötluskiirust ja mälukasutust. Oluline kriteerium oli andmetöötluse teostatavus ka väiksemal lauaarvutil, vältides vajadust eraldi klasteri järele. Kuna andmemahud on erakordselt suured, on kriitilise tähtsusega andmetöötlusprotsess jagada etappideks selliselt, et tavapärane arvuti suudaks vajaliku andmestiku edukalt töödelda.

4.1 Andmetöötlus paketid

Antud projekti andmed on väga erinevates vormingutes, mistõttu tuleb ka neid andmeid lugeda erinevate pakettidega. Andmete lugemiseks ja kirjutamiseks on kasutatud järgnevaid pakette:

- Rasterandmed – terra (versioon 1.8-86)
- Vektorandmed – sf (versioon 1.0-23)
- LiDARi andmed – lidR (versioon 4.2.2)

5 Andmestikud

5.1 Sentinel-2 andmed

Kaugseire andmetest on kasutusel Sentinel-2 andmed ning need laaditakse alla Copernicus-e andmeportaalist. See võimaldab kasutada kogu Sentinel-2 aegrida. Käesoleva projekti raames ei ole kasutatud Eesti EstHUB süsteemi, kuna avalikult on seal saadaval vaid viimase kahe aasta andmed ning eraldi ei ole ette projekti raames nähtud EstHUB andmetöötluskeskonna kasutust, kuna kokku tuleb kombineerida erinevaid andmeid ning enamust nendest ei ole EstHUBis saada. Seetõttu moodustab Sentinel-2 andmestik väikse osa kogu andmetöötlustest ning otstarbekam on vajalikud andmed alla laadida Copernicus-e andmeportaalist.

Copernicus võimaldab andmete automaatselt alla laadimist ning kerge vaevaga on võimalik sama andmete alla laadimise protseduuri kohandada ka EstHUB-i jaoks, aga kuna nendel on päringute parameetrid pisut erinevad, siis varasemate andmete jaoks on antud projekti käigus keskendunud Copernicus andmeportaalile. Kui edasine töötlus on keskendunud vaid viimasele kahele aastale, siis on võimalik kogu süsteem üles ehitada ka EstHUBist andmete laadimisele.

Tuleb arvestada, et mõlemas süsteemis peab kasutaja olema registreeritud ning andmete alla laadimine eeldab ka kasutaja autentimise funktsiooni.

5.2 LiDAR-i andmed

Metsa mõõtmete tuvastamiseks kasutatakse LiDAR-i andmeid, mis laaditakse alla Maa- ja Ruumiameti geoportaalist. Nendest andmetest kasutatakse ainult tava ja metsanduslikku lendu.

5.3 Eesti Topograafia andmekogu

Metsa puhul on tegemist maakasutuse kategooriaga, mistõttu tuleb arvestada ka sellega, et kõik puud ei ole alati metsamaa. Sellest tulenevalt tuleb eristada mitte-metsamaal kasvavad puud. Projekti käigus eraldi meetodit nende tuvastamiseks välja ei töötata ning selleks kasutatakse ETAK andmestikust alade filtreerimist erinevate kihtide kaupa. Need andmed saab alla laadida kas Maa- ja Ruumiameti geoportaalist

Alad, millel metsamaad ei tuvastata on järgmised:

- õuealad - ETAK kiht e_302_ou_a
- hooned - ETAK kiht e_401_hoone_ka
- teed – ETAK kiht e_501_tee_a
- vooluveekogud – ETAK kiht e_203_vooluveekogu_a

- seisuveekogud ETAK kiht e_202_seisuveekogu_a

Lisaks vaadeldakse ka joonobjekte, millel arvestatakse selle laiussega vähemalt 3 m:

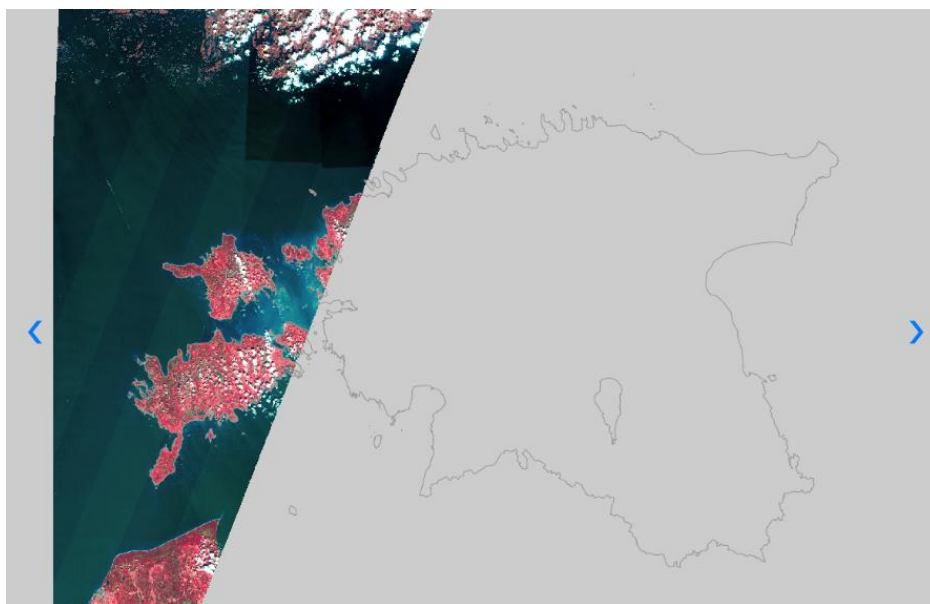
- elektriliinid – ETAK kiht e_601_elektriliin_j
- vooluveekogud – ETAK kiht e_203_vooluveekogu_j
- teed – ETAK kiht e_501_tee_j

6 Satelliidiandmete töötlus

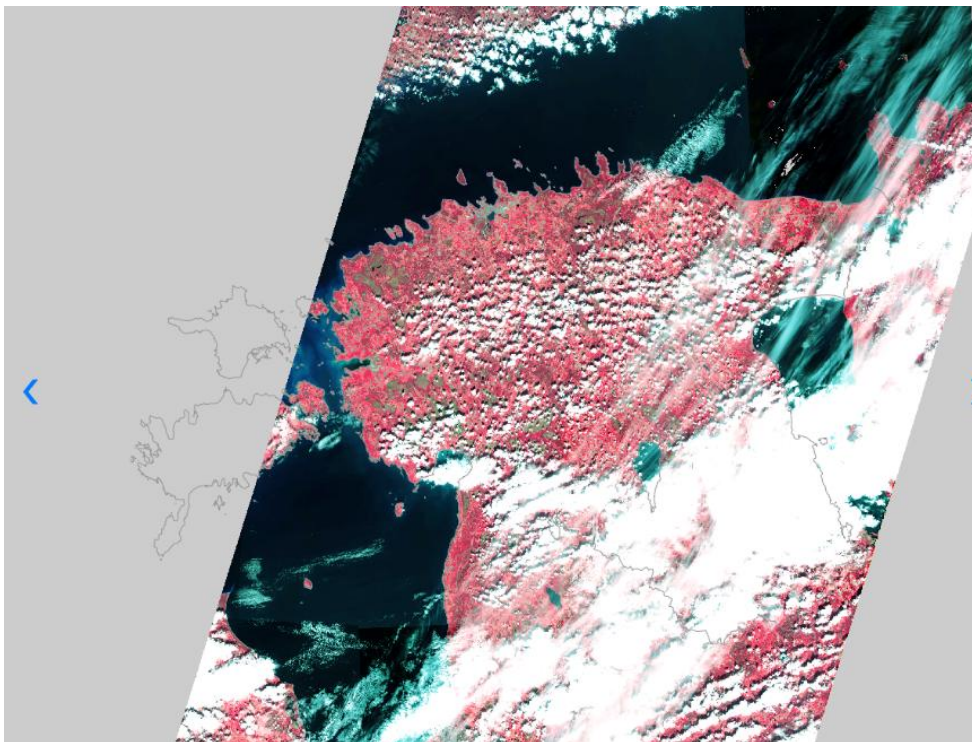
6.1 Metoodika üldine põhimõte

Tuleb arvestada, et LiDARi andmeid ei ole ühelgi aastal kogu Eesti kohta ning seetõttu on vajalik tuvastada ka konkreetse aasta kohta kogu metsamaa ala. Selleks saab kasutada satelliidiandmeid.

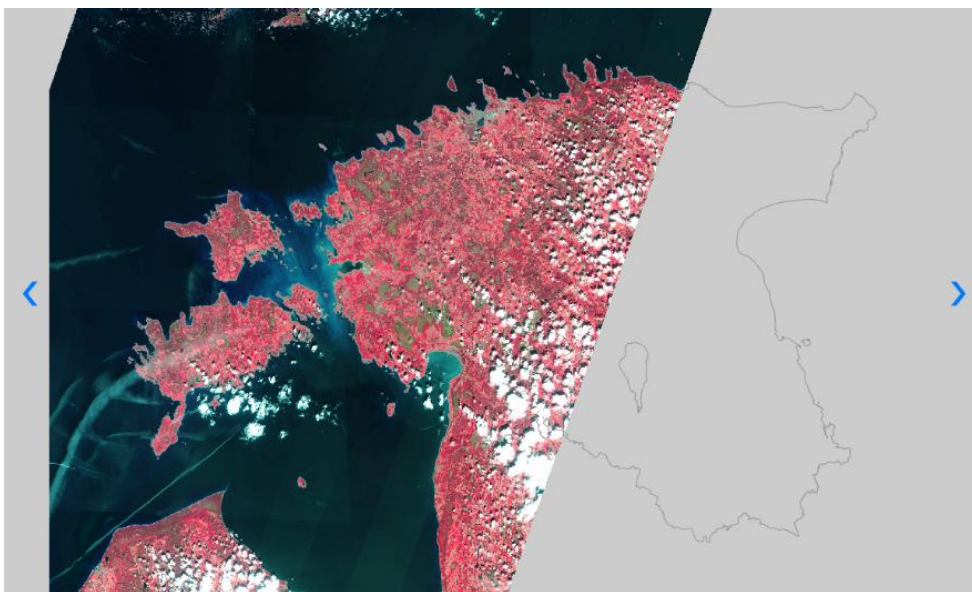
Käesolev töö on keskendunud Sentinel-2 andmetele, mis laaditakse alla Copernicus andmeportaalist. Tuleb arvestada, et üks päev ei kata kogu riiki ning satelliitpiltide puhul esineb pilvi, mis ei võimalda kogu pilti alati kasutada. Satelliit lendab orbiidil ning need orbiidid on nummerdatud. Antud töö puhul kasutatakse neid satelliitpilte, mis on orbiitidelt 36, 136 ja 79. Need on orbiidid, mis katavad vähemalt poolt Eestis ning võimaldavad piisavalt usaldusväärselt arvutada metsasuse tõenäosuse kaarti. Satelliitpildi nimes kajastub see vastavalt *_R036_*, _R136_* ja *_R079_*



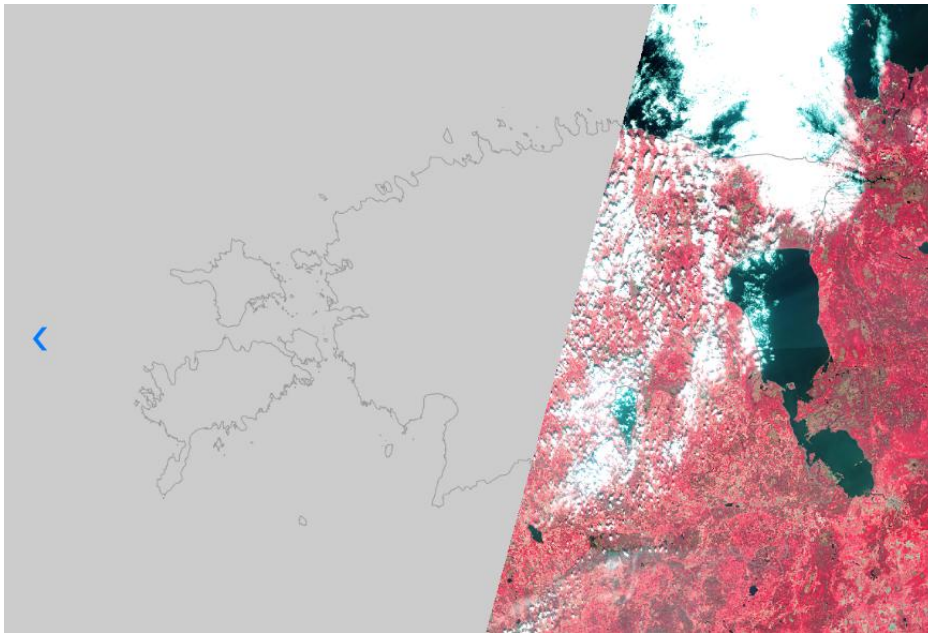
Joonis 1. Orbiidi 122 piltide kattuvus Eestiga.



Joonis 2 Orbiidi 36 piltide kattuvus Eestiga.



Joonis 3 Orbiidi 79 piltide kattuvus Eestiga.



Joonis 4 Orbiidi 136 piltide kattuvus Eestiga.



Joonis 5 Orbiidi 93 piltide kattuvus Eestiga.

Satelliitpiltide töötlemiseks kasutame paketti terra, mis võimaldab lugeda R-is andmestike gruppide kaupa (subdataset). Funktsioon rast võimaldab ette anda ka parameetrit subds ning sellele vastavalt järjekorra numbrit. Parameetriga subds = 1 loetakse sisse kihid B4, B3, B2, B8 ehk kõik 10 m piksliga kihid.

Andmetöötlemise kiirendamiseks on koostatud vastavus EPK10T ruutude ja Sentinel-2 kaardi-lehtede süsteemile, et vältida topelt arvutusi ning tööd oleks võimalik korraldada paralleelselt EPK10T ruutude alusel. See on optimaalne andmemaht, et ei koormaks mälu üle.

Satelliitpiltide puhul loetakse sisse esmalt satelliidiandmed ning samuti sellele pikslile vastav treeningandmestiku kiht. Käesoleva töö käigus kasutati treeningandmeteks SMI proovitükkide andmeid, kus

Käesoleva töö käigus katsetati erinevaid masinõppe lahendusi pikslite klassifitseerimiseks, aga arvutusmahult osutus kõige otstarbekamaks kasutada RandomForest funktsiooni pakettis ranger.

Satelliitpiltide puhul tuleb arvestada, et piltide valik ei ole alati sobilik, sest vegetatsiooniperioodi jooksul pikslite väärtused varieeruvad ning erinevad maakasutuse kategooriad võivad olla spektraalselt väga sarnased.

Klassifitseerimise puhul võib küll esineda treeningandmetes ka mõningasi muutusi, aga kokkuvõttes marginaalsed muutused aitab see siiski tuvastada. Kui õpetusandmestikus on vähesel määral nõ muutunud piksleid, siis see mõjutab marginaalselt klassifitseerimise tõenäosust, aga muutunud pikslile omistatakse uue klassi tõenäosus.

6.2 Sentinel-2 andmete kogumine ja filtreerimine

Metoodika algab vajalike Sentinel-2 satelliitpiltide nimekirja koostamisega, kasutades Copernicus Data Space'i kataloogi (endine Sentinel Copernicus Open Access Hub).

- Ajavahemik: Pildid valitakse vahemikust 1. mai kuni 30. september määratud aasta kohta.
- Geograafiline ulatus (BBox): Andmed piiratakse Eesti ala katvale ristkülikule.
- Pilvisuse piirang: Lubatakse pildid, mille pilvisus on vahemikus 0% kuni 30%.
- Tootetüüp: Kasutatakse S2MSI2A toodet, mis on atmosfääri poolt korrigeeritud (Level-2A).
- Orbiidi filtreerimine: Andmekvaliteedi ja järjepidevuse tagamiseks filtreeritakse välja pildid, mis on saadud kindlatelt suhtelistelt orbiidinumbrilt (36, 136, 79).
- Kvaliteedikontroll: Iga kuupäeva kohta peab olema üle 5 pildi, et tagada piisav geograafiline katvus treeninguks ja analüüsiks.
- Nimekirja loomine: Filtreeritud andmetest koostatakse CSV-fail, mis sisaldab allalaadimiseks vajalikku infot (nimi, GUID, kuupäev, ruut).

6.3 Sentinel-2 andmete allalaadimine

Järgmises etapis laaditakse kättesaadud nimekirja alusel puuduvad Sentinel-2 paketid (ZIP-failid) Copernicus Data Space'ist alla.

- API/Klassi kasutamine: Allalaadimiseks kasutatakse eelduslikult defineeritud copsi R6 klassi, mis haldab autentimist ja juurdepääsuloa (tokenite) uuendamist.
- Andmete päring: Kasutatakse httr2 paketti, et esitada päringud Copernicus Download API-sse, kasutades iga toote GUID-d.
- Salvestamine: Iga ZIP-fail salvestatakse spetsiifilisse kausta struktuuri.
- Puuduva kontroll: Allalaadimine toimub ainult siis, kui vastavat faili kohapeal veel ei eksisteeri.

6.4 RandomForest mudeli treenimine

Klassifitseerimiseks kasutatakse **Random Forest (Juhuslik Mets)** masinõppe algoritmi. Mudel treenitakse eraldi iga unikaalse ja sobiva Sentinel-2 pildistuspäeva kohta, et arvestada fenoloogiliste erinevustega.

Random Forest valiti järgmistel põhjustel, mis sobivad metsamaa kaugseireks:

- **Müra ja vigade taluvus:** Kuna satelliitpildidel võib esineda atmosfäärihäireid või ebatäpseid piksliväärtusi (müra), on paljudest otsustuspuudest koosnev ansambel-mudel stabiilsem kui üksikud mudelid.
- **Mittelineaarsus:** Seosed spektraalkanalite ja metsa omaduste vahel ei ole alati lineaarsed. Random Forest suudab tõhusalt modelleerida keerulisi ja mittelineaarseid seoseid ilma andmete eelneva normaliseerimise vajaduseta.
- **Korreleeruvad tunnused:** Sentinel-2 kanalid on omavahel tugevas korrelatsioonis (nt B2 ja B3). Random Forest tuleb multikolineaarsusega toime paremini kui paljud teised klassifitseerijad.

Õpetusandmestiku ettevalmistamine ja tasakaalustamine

- **Lähteandmed:** Kasutatakse statistilise metsainventuuri (SMI) andmeid.
- **Ajaline filter:** Kasutatakse treeningpunkte, mis on kogutud klassifitseeritava aasta ja eelneva nelja aasta jooksul ehk ühe SMI perioodi (5 aasta) kõik proovitükid. See tagab, et kogu Eesti on ruumiliselt ühtlaselt kaetud ja esindatud piisava hulga punktidega.
- **Ruumiline filter:** Selleks, et vältida metsa serva lähedal olevaid proovitükke, kasutatakse ETAK puittaimestiku kihti esmaseks filtreerimiseks, kus välja jäetakse need proovitükid, mis asuvad sellele piirile lähemal kui 20 m ehk kahe piksli jagu satelliitpildil.
- **Klasside tasakaalustamine:** Maakategooriate pindalad on väga varieeruvad ning selleks, et need ei oleks mudeli koostamisel väga tasakaalust väljas, kasutatakse vaid kahte klassi: metsamaa ja mitte-metsamaa. Kuna Eestis on metsasus ca 50%, siis on

need kaks klassi võrdselt esindatud ning mudeli tulemus ei ole kallutatud ebavõrdsete klasside hulga tõttu.

Mudeli hüperparameetrid

Mudeli seadistamisel on eesmärgiks vähendada ülesobitamist (overfitting) ja tagada mudeli üldistusvõime erinevatel kuupäevadel. Rakendatakse järgmisi printsiipe:

- **Puude arv ($n_estimators$):** Kasutatakse piisavalt suurt arvu puid (käesolevas töös 30), et tagada mudeli stabiilsus ja vähendada ennustuste dispersiooni. Pärast teatud piiri (umbes 100 puud) mudeli täpsus oluliselt ei suurene, kuid arvutusressurss kasvab, seega on valitud optimaalne vahemik.
- **Puu maksimaalne sügavus (max_depth):** Puude sügavust ei piirata liiga rangelt, et võimaldada mudelil õppida peeneid erinevusi spektraalsignatuurides, kuid vajadusel kasutatakse piiranguid koos *min_samples_split* parameetriga, et vältida üksikute erindite (outliers) päheõppimist.
- **Tunnuste valik ($max_features$):** Igas sõlmes (split) kaalutakse alamhulka tunnustest (tavaliselt ruutjuur tunnuste koguarvust). See tagab, et puud on omavahel vähem korreleeritud (dekorreleerimine), mis omakorda tõstab ansambli täpsust, eriti olukorras, kus mõned kanalid (nt B11 ja B12) on märgalade tuvastamisel domineerivad.
- **Kriteerium:** Puude hargnemisel kasutatakse **Gini ebapuhtuse (Gini impurity)** indekssi, mis on standardne ja arvutuslikult efektiivne meetod binaarse klassifitseerimise optimeerimiseks.

Mudeli andmeteks kasutatakse järgmisi spektraalkanaleid.

Nähtava valguse kanalid (10 m): B2, B3, B4

Need 10-meetrise eraldusvõimega ribad (Sinine, Roheline, Punane) moodustavad klassifitseerimise "vundamendi".

- **Ruumiline täpsus:** Kuna need on kõrgeima lahutusega (10 m), aitavad need täpselt paika panna metsaeraldiste piirid, teed, sihid ja legendikud.
- **Klorofüllü neeldumine:** Eriti oluline on punane kanal (B4), kus toimub tugev klorofüllü neeldumine. See on baasiks taimkatteindeksite (nagu NDVI) arvutamisel, mis näitab biomassi üldist hulka.

Lähi-infrapuna ja "Punane Serv" (NIR & Red Edge): Puittaimestiku eristamine

See on analüüsi kõige kriitilisem osa. Tavalise RGB pildi peal on nii heinamaa kui ka mets "roheline", kuid spektraalselt on nad väga erinevad just selles vahemikus.

Miks on vaja mitut NIR kanalit (B5–B8A)?

- **Punane serv (Red Edge - B5, B6, B7):** Need 20 m kanalid katavad kitsa spektriosa, kus taimede peegeldus tõuseb järsult punasest (madal peegeldus) lähi-infrapunani (kõrge peegeldus).
 - **Erinevus roht- ja puittaimede vahel:** Puittaimestik (metsal) ja rohttaimestik (põllud, niidud) on raku ehitus ja lehtede asetus erinev. See muudab "punase serva" tõusu nurka ja asukohta. Kasutades kanaleid B5, B6 ja B7, saab luua indekseid (nt NDRE), mis suudavad eristada tihedat metsa madalast rohus või võsast paremini kui tavaline NDVI.
- **Lai vs Kitsas NIR (B8 vs B8A):**
 - **B8 (10 m):** See on laiem spektriga ja parema ruumilise lahutusega. See on vajalik biomassi detailseks kaardistamiseks (metsa tihedus).
 - **B8A (20 m):** See on kitsam riba, mis on spetsiifiliselt disainitud vältima veeauru mõju atmosfääris. Kombineerides B8A ja punase serva kanaleid, saab täpsemini hinnata lehepinnaindeksit (LAI) ja eristada lehtpuid okaspuudest.

Kokkuvõtvalt: Kombinatsioon kanalitest B5–B8A võimaldab eristada **liigilist koosseisu ja struktuuri** (nt kas tegemist on noore võsa, küpse metsa või heinamaaga), mida ainult ühe NIR kanali abil teha ei saaks.

Lühilaine infrapuna (SWIR): B11 ja B12 märgalade tuvastamiseks

SWIR kanalid (Short-Wave Infrared) on 20 m eraldusvõimega ja reageerivad peamiselt veesisaldusele taimedes ja pinnases, samuti ligniini ja tselluloosi sisaldusele.

- **Märgalade ja veerežiimi eristamine:** Vesi neelab SWIR kiirgust väga tugevalt (peegeldus on madal).
 - **Kuiv mets vs märg mets:** Terve ja kuiv okasmets peegeldab SWIR kanaleid teatud määral tagasi. Soostunud metsade või lagedate märgalade puhul on pinnas ja taimestik veest küllastunud, mistõttu B11 ja B12 väärtused langevad märgatavalt madalamale.
- **Struktuuri tuvastus (B11/B12):** Lisaks niiskusele aitavad need kanalid eristada vanu metsi (rohkem varjusid ja keerulisem struktuur, mis neelab SWIR-i) noorendikest (ühtlasem peegeldus).

SCL (Scene Classification Layer): Kvaliteedikontroll

SCL ei ole otsene satelliidikanal, vaid eeltöötuse tulemus (mask), mis on metsaanalüüsis hädavajalik.

- **Andmete puhtus:** Metsa kaugseire on kasutu, kui analüüsitakse kogemata pilvi või nende varje metsa pähe. SCL kiht võimaldab enne klassifitseerimist eemaldada

pikslid, mis on vigased, pilves või pilvevarjus, tagades, et B2–B12 kanalite info on tõepoolest maapinnalt pärit.

6.5 Päevapõhine treenimine

Metsamaa ja taimkatte klassifitseerimisel on kriitilise tähtsusega arvestada taimestiku ajalist dünaamikat. Kuigi satelliitpiltide piksliväärtuste mediaani (mediaanpildi) arvutamine on levinud viis pilvevabade mosaiikide loomiseks, on klassifitseerimismudelite täpsuse huvides parem treenida mudel igale sobivale päevale eraldi ja arvutada mediaan tulemustest (ennustustest). Sellel on kolm peamist põhjust:

- **1. Taimkatte fenoloogiline muutumine**

Vegetatsiooniperioodi jooksul ei ole kanali väärtused staatilised. Taimkatte spektraalne signatuur muutub pidevalt:

- **Kevad:** Kiire klorofülliteke, "punane serv" (Red Edge) on järsk, kuid lehestik võib olla hõredam.
- **Suvi:** Maksimaalne biomass ja lehepinnaindeks, kuid võimalik kuivusstress (mõjutab B11/B12 kanaleid).
- **Sügis:** Klorofüllide vähenemine, niiskuse suurenemine.

Probleem mediaan satelliitpildiga: Kui me võtame terve suve piltide peale mediaani, siis me "tasandame" need erinevused. Me loome **süntheetilise spektraalse signatuuri**, mida looduses ühelgi ajahetkel ei pruugi eksisteerida. Näiteks, kui märgala on mais vee all (madal peegeldus) ja juulis kuiv (kõrge peegeldus), siis mediaanpilt näitab "pool-niisket" olukorda, mis võib mudeli segadusse ajada ja liigitada ala valesti.

- **2. Spektraalsete suhete säilitamine (Füüsikaline korrektsus)**

Satelliitpildi iga piksel üksikul päeval esindab füüsikalist reaalsust – konkreetset päikese nurka, atmosfääri seisundit ja taimkatte hetkeseisu.

- **Üksikute päevade mudelid** kasutavad puhtaid füüsikalisi seoseid (nt suhe B4 ja B8 vahel konkreetsetel päeval).
- **Mediaanpilt** võib kombineerida erinevate päevade andmeid (nt punane kanal mai-kuust ja NIR kanal augustist). See lõhub kanalitevahelised loomulikud korrelatsioonid, mis on vajalikud just peente erinevuste (nt lehtpuu vs okaspuu või märgala vs mets) tuvastamiseks.

- **3. Tulemuste mediaan vähendab juhuslikku müra ("Ensemble" efekt)**

Lähenedamine, kus igale päevale koostatakse eraldi mudel ja lõpuks võetakse ennustuste mediaan, toimib nagu **ekspertide konsiilium**:

- Kui meil on 10 sobivat päeva, siis meil on 10 "hinnangut" igale pikslile.
- Mõnel päeval võib tulemust mõjutada kerge vihm, öine vihm (märjad lehed) või varjud. See võib tekitada üksikus mudelis vea.
- Võttes **tulemuste mediaani**, filtreerime need ajutised anomaaliad välja. Kui 8 päeva 10-st ütleb, et tegemist on männikuga, ja 2 päeva (müra tõttu) pakub midagi muud, siis mediaan annab tulemuseks õige klassi (männik).

Koostades mudeli igale päevale eraldi, säilitame me taimestiku sesoonse info ja füüsikalise terviklikkuse. Tulemuste (mitte sisendandmete) agregeerimine tagab statistiliselt kõige stabiilsema ja usaldusväärsema klassifikatsiooni, mis on vähem tundlik üksikute päevade ilmastikuolude suhtes.

- Satelliitpildi esmane andmetöötlus
 - Eraldatakse 10 m ribad (B2, B3, B4, B8, SCL - Sentinel Cloud Mask) ja 20 m ribad (B5, B6, B7, B8A, B11, B12).
 - Andmete ühendamise ja filtreerimise:
 - Ühendatakse kõigi selle päeva piltide väärtused.
 - Andmed filtreeritakse vastavalt SCL (Scene Classification Layer) väärtustele, et eemaldada pilved, vari ja lumi (kasutatakse SCL klasside 4 (Vegetation), 5 (Non-Vegetated) ja 6 (Water) piksleid).
 - Eemaldatakse vead ($B4 > 0$) ja read, kus on puuduvad väärtused.
- Random Foresti treenimine:
 - Iga päeva kohta treenitakse ranger paketiga (Random Forest) mudel, mis ennustab metsa tõenäosust (probability = TRUE).
 - Puude arv on seatud 30-le (optimaalsuse huvides).
- Mudeli salvestamine: Treenitud mudel salvestatakse iga päeva kausta (randomForest.rds).

6.6 Klassifitseerimine ja prognoos

Päevapõhiste mudelite abil arvutatakse metsa tõenäosus igale 5x5 km EPK-ruudule.

- Ruutude valik: Kasutatakse geograafilist kihti (eesti.gpkg), mis sisaldab EPK-ruutude geomeetria ja orbiidiinfot.

- Töötlemise järjekord: Klassifitseeritakse ainult need ruudud ja pildid, millel on olemas mudel ja mille väljundfaili (TIF) veel ei eksisteeri.
- Klassifitseerimise töövoog:
 - Laetakse vastav päeva Random Forest mudel.
 - Loetakse Sentinel-2 ribad (10m ja 20m) ning projitseeritakse/lõigatakse need välja 5x5 km rastri (resolutsiooniga 5 või 10 meetrit - res = 5 või res = 10) šablooni järgi.
 - Muudetakse piksliväärtused andmetabeliks.
 - Filtreeritakse SCL-i järgi ja tehakse ennustus.
 - Tulemus on Metsa tõenäosus väärtusega 0–100.
 - Tulemused salvestatakse INT1U (0-255) TIF-failina spetsiifilisse kausta.

6.7 Lõpliku tõenäosuskaardi loomine

Lõpuks kombineeritakse kõigi kvalifitseeritud päevade klassifitseeritud tulemused üheks koondmaskiks.

- Rastri kombineerimine: Kõik valitud TIF-failid ühe EPK-ruudu kohta (tavaliselt 5x5km) laetakse sisse ja laotakse kokku.
- Pikslite mediaani arvutamine: Kombineeritud rastrist arvutatakse piksli mediaanväärtus. See vähendab pilvedest ja varjudest tingitud müra ning annab stabiilsema hinnangu.

Tulemuskaart teisendatakse 5 m piksliga rastroks, mis võimaldab täpsemalt määrata metsas olevaid joonobjekte.

6.8 Aastate 2017-2025 tõenäosuste võrdlus

Metoodika testimiseks arvutati aastate 2017-2025 satelliitpiltide alusel metsasuse tõenäosused. Nende tulemuste puhul võrreldi nii pikslite tasemel aastate vahelist korrelatsiooni kui ka SMI andmetega võrdluses tulemuste täpsust.

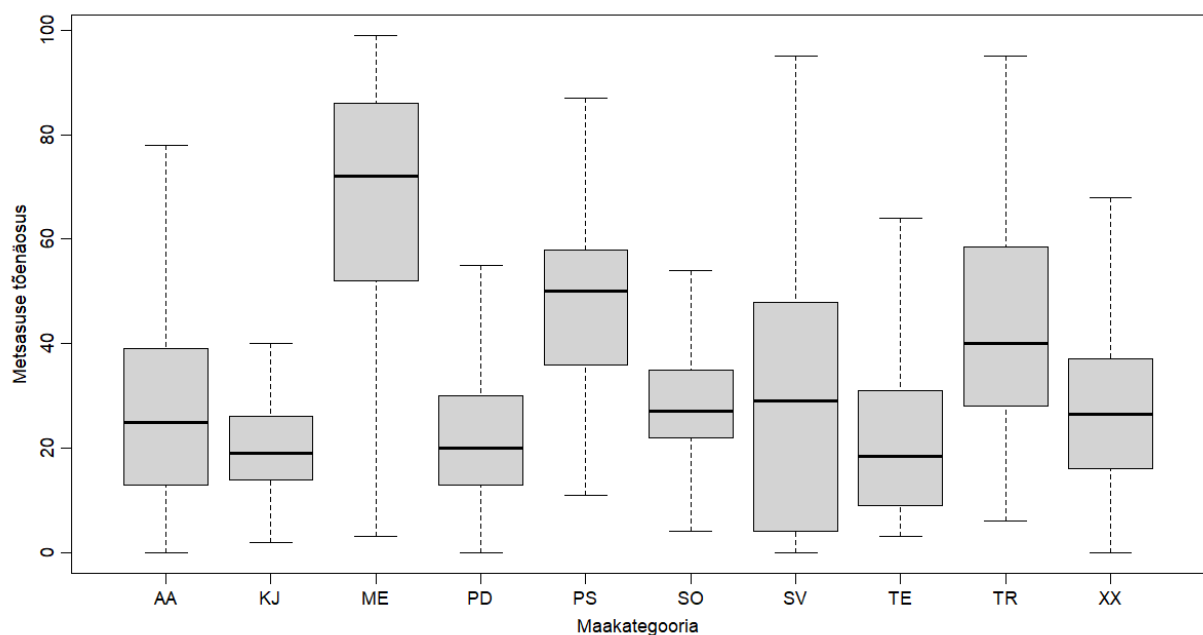
Tabelis on näha, et kahe järjestikuse aasta vahel on korrelatsioon suurem ($\geq 0,95$) ning mida suurem on ajaline vahe, seda väiksemaks muutub ka korrelatsioon, mis on tingitud juba maastikul toimuvatest muutustest.

Tabel 2. Aastate vaheline korrelatsioon

	A2017	A2018	A2019	A2020	A2021	A2022	A2023	A2024	A2025
A2017	1	0,95	0,94	0,93	0,92	0,91	0,92	0,91	0,9
A2018	0,95	1	0,97	0,96	0,94	0,93	0,94	0,94	0,92
A2019	0,94	0,97	1	0,97	0,95	0,94	0,95	0,94	0,93

A2020	0,93	0,96	0,97	1	0,96	0,95	0,95	0,95	0,94
A2021	0,92	0,94	0,95	0,96	1	0,95	0,96	0,95	0,94
A2022	0,91	0,93	0,94	0,95	0,95	1	0,96	0,95	0,94
A2023	0,92	0,94	0,95	0,95	0,96	0,96	1	0,97	0,95
A2024	0,91	0,94	0,94	0,95	0,95	0,95	0,97	1	0,96
A2025	0,90	0,92	0,93	0,94	0,94	0,94	0,95	0,96	1

Joonisel 6 on näha, et kõige suurem on ennustatud metsasuse tõenäosus metsamaal (ME) ning põõsastikel (PS).



Joonis 6. Ennustatud metsasuse tõenäosuse jagunemine maakategoriate kaupa.

6.9 Testitud meetodid

Lisaks kasutusele võetud Random Forest mudelile katsetati käesoleva töö käigus ka teisi mudeleid, millega ennustada metsasuse tõenäosust satelliitpiltide alusel, aga nende ennustuse täpsus jäi valdavalt alla 80%.

Testitud meetodid:

- PyTorch paketiga U-Net tehismärgivõrkudel põhinev ruumist paiknemist arvestava mudelid erinevate parameetrite arvuga
- PyTorch paketiga pikslipõhine tehismärgivõrkudel põhinev ruumilist paiknemist mitte arvestavad mudelid erinevate parameetrite arvuga

- XGBoost mudel pikslipõhiseks klassifitseerimiseks
- GAM tõenäosuse mudel pikslipõhiseks klassifitseerimiseks
- Piiride tuvastamiseks segmenteeriti pilte, aga kuna metsa vahel on joonobjektid ning metsa osas on spekter väga varieeruv, siis ei suutnud algoritm piisava täpsusega eristada piire.

7 LiDAR-i andmete töötlus

LiDAR-i andmed on alla laaditavad EPK2T ruudustiku 1 km² ruutude kaupa. Andmetöötluks on vajalik esmalt alla laadida kõik vajalikud ruudud, kus arvestatakse kogu Eesti katvust soovitud kriteeriumite alusel.

Andmete alla laadimiseks filtreeritakse välja EPK2T andmestikust need ruudud, mis kattuvad Eesti maismaa kaardikihiga. Kuna osad ruudud on ka meres, siis neid ei ole otstarbekas alla laadida.

Projekti käigus on loodud skript, mis võimaldab automaatselt alla laadida kõik vajalikud LiDAR-i andmed, mida antud töötluks puhul kasutatakse. Kuna ühekordne Eestit kattev andmestik võib olla kuni 1 TB suur (kogu andmestik 2008-2024, milles on koos tava ja metsanduslikud andmed, on ca 4,4 TB), siis on väga oluline selle andmete haldamine, kuna nende eraldi hoidmine ei pruugi olla otstarbekas ning mõistlikum on töötlemise järgselt failid ära kustutada ning jätta alles ainult tuletid, mille maht on oluliselt väiksem.

LiDAR-i andmete töötlemisel on esimeseks ülesandeks punktide kõrguse normaliseerimine maapinna kõrgusmudeli alusel. Selleks kasutatakse antud töös Maa- ja Ruumiameti poolt koostatud 10 m piksliga loodud DTM (geoportaalist DTM_10m_eesti.tif).

Andmetöötluksprotsessis loetakse esmalt sisse rlas paketi LiDAR-i andmed, millest töötlemise andmemahu kokkuhoiu jaoks loetakse ainult xyz koordinaadid ning filtreeritakse välja punktid klassidega 7 ja 18 (müra all ja ülal).

Järgmiseks sammuks on sisse lugeda DTM andmestik ainult antud ruudu ulatuses, et hoida kasutatav andmestik minimaalsena. DTM on loodud kasutades kevadisi lende, et vältida maapinna nihet taimkattest tulenevalt. Metsa all olev taimestik võib olla kuni 1 m kõrgune suvisel ajal ning seetõttu tekib sellest maapinna kõrguse nihe ning sellega hinnatakse metsa kõrgus alla.

Igale LiDAR-i punktile leitakse, milline on selle all olev maapinna kõrgus ning selleks tuleb esmalt arvutada, milline on piksel antud punkti all ning seejärel võtta DTM andmestikust vastav punkt. Selleks loetakse kodu DTM andmestik vektorina mällu ning igale LiDAR-i punktile leitakse, millise vektori elemendiga see kattub ning seejärel arvutatakse LiDAR-i punkti kõrguse ja maapinna punkti vahena kõrgus maapinnast.

LiDAR-i andmetest arvutatakse 5 m piksli kohta välja võrastiku katvus (%). Käesolevas töös on kasutatud lävendina 2 m, kuigi metsa puhul on lävendiks 1,3 m ning põhjuseks on konservatiivne lähenemine, kuna maapinna kõrguse mudel võib olla mõningase määramatusega. 5 m piksel võimaldab täpsemalt saada metsavaheliste joonobjektide piire ning samuti metsa piiri.

Kuna 5 m piksli sisse võib jääda vähe LiDARi punkte, siis järgmise sammuna arvutatakse sellest rasterfailist naaberpiksleid arvestades mediaanväärtus ehk 3x3 piksli ala mediaan, mis aitab siluda metsa sees olevaid mikrohäile, aga jätab alles metsa serva. Kui mediaan asemel kasutada keskmist, siis hägustab see metsa serva.

8 Tulemuste kokku panemine

Protsess on realiseeritud R-keeles, kasutades hübriidmeetodit, mis ühendab Eesti Topograafia Andmekogu (ETAK) vektorkihid ja kaugseire (LiDAR, Sentinel-2) rasterandmed. Eesmärk on luua täpsustatud metsamaa ruumikuju, mis arvestab nii ametlikku maakasutust kui ka tegelikku taimkatte olukorda.

Protsess jaguneb viieks põhietapiks:

- **1. Keskkonna ja alusrastri initsialiseerimine**

Esmalt luuakse tühi põhiraster, mis määrab kogu edasise töötamise ruumilise ulatuse ja täpsuse.

- **Ruumiline ulatus:** Eesti põhikaardi ruudustik (koordinaadid: 365000–740000, 6375000–6635000).
- **Resolutsioon:** 5 x 5 meetrit piksel (EPSG:3301).
- **Andmetüüp:** INT1U (täisarvud 0-255), optimeeritud LZW pakkimisega.
- **2. ETAK pinnaobjektide rasterdamine**

Aluskaart moodustatakse ETAK pinnaobjektide kandmisega rastrisse.

- **Pinnakihid:** Skript loeb sisse kõik kaustas olevad ETAK „*_a.shp“ (alad) failid.
- **Reklassifitseerimine:** ETAK koodid teisendatakse üldistatud koodideks (funktsioon arvuta_uus_kood), rühmitades maakasutustüübid.
- **Prioriteetsus:** Kihid kantakse rastrisse järjestikku, kusjuures hilisemad kihid kirjutavad varasemad üle.
- **Erandid:**
 - **Hooned (E_401):** Lisatakse eraldi kihina, et tagada hoonestuse välistamine metsamaskist.
 - **Välised alad:** Eesti maismaapiirist välja jäävad alad nullitakse.

- **3. Joonobjektide ja taristu lisamine**

Joonobjektid (teed, liinid, kraavid) on metsamaskis olulised katkestajad. Need puhverdatakse ja "põletatakse" alusrastrisse, lõigates läbi metsamassiivide (eelduseks on, et lõige toimub vaid väärtuse 35 ehk metsa peal).

- **Teed (E_501):**

- Valim: Teed laiusega 3–8 meetrit.
- Töötlus: 5 m puhver (kogulaius 10 m).
- **Vooluveekogud (E_203):**
 - Valim: Laius 20–40 meetrit.
 - Töötlus: 5 m puhver.
- **Elektriliinid (E_601) – Pingeipõhine puhverdamine:**
 - Liinid < 35 kV: 5 m puhver.
 - Liinid ≥ 35 kV ja < 220 kV: 12.5 m puhver.
 - Liinid ≥ 220 kV: 20 m puhver.
 - *Märkus:* Elektriliinide puhul kasutatakse update=TRUE meetodit, mis kirjutab üle kõik allasuvad klassid.
- **4. Kaugseire andmetega täpsustamine (järeltöötlus)**

Selles etapis korrigeeritakse ETAK andmeid tegeliku taimkatte infoga, kasutades LiDAR (taimkatte kõrgusmudel) ja Sentinel-2 (optiline satelliitpilt) andmeid.

Sisendandmed:

- **LiDAR mask:** Taimkate loetakse kõrgeks, kui lidar_cc_med.tif väärtus ≥ 30 (võra liitus/kõrgusindeks). Siin on võimalik määrata võrastiku katvust erinevate tihedustega, mis vastaks oodatud metsa definitsioonile.
- **Sentinel-2 mask:** Taimkate loetakse metsaks, kui S2_2025_med.tif tõenäosus ≥ 50%.

Korrigeerimisloogika:

1. **Põõsastike ja niitude ümberhindamine:** Kui ETAK klass on märgala või turbakarjäär (koodid 36, 37), kuid LiDAR näitab kõrget taimestikku, määratakse klassiks **Mets (35)**.
2. **Lagedate alade ümberhindamine:** Kui ETAK klass on lagedad alad või haritavad maad (koodid 33, 34), kuid nii LiDAR kui ka Sentinel-2 kinnitavad metsa olemasolu, määratakse klassiks **Mets (35)**.
3. **Kaitsealused pargid:** EELIS andmebaasi parkide kiht lisatakse eraldi (kood 38), kuid seda töödeldakse metsamaskis sarnaselt metsaga.
4. **Linna metsad:** Täiendavalt lõigatakse välja saadud metsamaast linnad (asustusalad, tüüp: 4 - linn, 5 - omavalitsuse sisene linn, 6 - linnaosa)

Lõplik rasterfail on etak_kood_mod_s4.tif, milles on maakasutuse kategooriad vastavalt ETAK koodidele, mis on teisendatud kahekohaliseks jagades sajaliseid kümnega: 301 => 31, 302 => 32 jne. See võimaldab hoida tulemusfaili väiksemana (raster on 8-bitine) ning teiste tulemus võimaldab näha, mis on need muud kategooriad, millega metsamaa piirneb.

Viimase sammuna eraldatakse sellest metsamaa, mis arvutatakse kiht == 35 ning tulemuseks on binaarne metsamaa kaardikiht metsamaa.tif.

Metsamaski testiti SMI andmete alusel, kust välja jäetud proovitükid, mis asuvad saadud metsamaski puhul servast rohkem kui 20 m kaugusel, et vältida asukoha täpsusest tingitud valesti klassifitseerimist. Tuleb arvestada, et 14% proovitükkidest asub metsa servale lähemal kui 20 m, mistõttu statistika põhineb 22 575 SMI proovitükil. Ennustuse täpsus on 97,4%, kappa koeffitsient 0,947.

Lõplik tulemus on rasterkaart, mida tellija saab vajadusel oma äranägemise järgi ka vektoreerida.

Käesoleva töö käigus saadud tulemusel on metsamaa pindala 2,441 miljonit hektarit.

9 Üleantavad tulemused

Käesoleva tööga antakse üle lisaks kirjeldatud metoodikale ka:

- arvutuste koodid (LISAD 1-4).
- 2025. aasta Sentinel-2 ja SMI andmete alusel on arvutatud ühe aasta metsasuse tõenäosuse kaart *S2_2025_med.tif*.
- 2020-2023. aasta suviste ALS lendude alusel on koostatud LIDARi katvuse kaart *lidar_cc_med.tif*.
- Saadud tulemuste põhjal kogu Eestit kattev täiendatud ETAK rasterkaart *etak_kood_mod_s4.tif*.
- Viimasest on tuletatud metsamask.tif, kus on binaarne väärtus, kas piksliväärtus on 35 või mitte.

LISA 1. Sentinel-2 arvutuse kood

```
# =====
# SEADISTUS JA PAKETID
# =====

# Puhastame keskkonna, kui jooksutame interaktiivselt
if (interactive()) {
  rm(list = ls())
  gc()

  sapply(names(sessionInfo())$otherPkgs, function(x) {
    detach(paste0("package:", x), character.only = T)
  })
}

# Laeme vajalikud paketid (kasutame pacman'i mugavaks laadimiseks)
if (!require("pacman"))
  install.packages("pacman")
pacman::p_load(sf,
               terra,
               ranger,
               mirai,
               glue,
               jsonlite,
               httr2,
               dplyr,
               purrr,
               progress)

# Eeldame, et see fail tekitab muutuja 'cli_opt' (nt cli_opt$aasta)
# ja defineerib 'workDir'
source("set_cli_params.R")

# Hoiaime alles algsed muutujad, et hiljem puhastamisel mitte kõike kustutada
algne_keskkond <- c(ls(), "algne_keskkond")

# =====
# 1. SENTINEL-2 NIMEKIRJA KOOSTAMINE
# =====
cat(glue::glue("Kontrollib Sentinel-2 aasta {cli_opt$aasta} nimekirja...\n"))

nimekiri_fail <- glue::glue("sentinel_nimekiri_{cli_opt$aasta}.csv")

if (!file.exists(nimekiri_fail)) {
  # Parameetrid
  paev_1 <- glue::glue("{cli_opt$aasta}-05-01T00:00:00Z")
  paev_2 <- glue::glue("{cli_opt$aasta}-09-30T23:59:59Z")
  bbox   <- "22.5,57,28,60"

  # Soovitud ruutude nimekiri
  wgrd_nimed <- c(
    'T34VEK',
    'T34VEL',
    'T34VFK',
    'T34VFL',
    'T35VLE',
    'T35VLF',
    'T35VLG',
    'T35VMD',
  )
}
```

```

      'T35VME',
      'T35VMF',
      'T35VMG',
      'T35VND',
      'T35VNE',
      'T35VNF'
    )

    # API päringu koostamine
    base_url <- "https://catalogue.dataspace.coperni-
cus.eu/resto/api/collections/Sentinel2/search.json"

    params <- list(
      startDate      = paev_1,
      completionDate = paev_2,
      cloudCover     = "[0,20]",
      box            = bbox,
      productType    = "S2MSI2A",
      sortParam      = "startDate",
      maxRecords     = 2000,
      # Suurendasin veidi limiiti
      page           = 1
    )

    url_query <- paste0(names(params), "=", params, collapse = "&")
    full_url  <- paste0(base_url, "?", url_query)

    # Andmete pärimine
    resp <- jsonlite::fromJSON(full_url)

    if (length(resp$features) == 0)
      stop("Andmeid ei leitud!")

    # Andmete töötlemine (base R asemel dplyr ahel)
    nimekiri <- resp$features$properties |>
      as_tibble() |>
      filter(relativeOrbitNumber %in% c(36, 136, 79)) |>
      mutate(
        guid = basename(services$download$url),
        ruut = substr(title, 39, 44),
        nimi = gsub("SAFE", "zip", title),
        kuupaev = substr(nimi, 12, 19)
      ) |>
      filter(ruut %in% wgrd_nimed)

    # Filtreerimine: ainult kuupäevad, kus on rohkem kui 5 pilti (kvaliteedikont-
roll)
    s2_tabel <- nimekiri |>
      group_by(kuupaev) |>
      filter(n() > 5) |>
      ungroup() |>
      select(nimi, guid, kuupaev, ruut)

    write.csv(s2_tabel, nimekiri_fail, row.names = FALSE)
    cat(glue::glue("Nimekiri salvestatud: {nrow(s2_tabel)} kirjet.\n"))

  } else {
    cat("Nimekirja fail on juba olemas.\n")
  }

  # Puhastame mälu (v.a algsed muutujad ja nimekiri_fail)

```

```

rm(list = setdiff(ls(), c(algne_keskkond, "nimekiri_fail")))
gc()

#
=====
# 2. ALLALAADIMINE
#
=====
cat("Alustan puudevate failide allalaadimist...\n")

sentinel_nimekiri <- read.csv(nimekiri_fail)

# Moodustame failiteed
sentinel_nimekiri$file_path <- file.path(
  workDir,
  "sentinel",
  cli_opt$aasta,
  sentinel_nimekiri$kuupaev,
  substr(sentinel_nimekiri$nimi, 1, 3),
  # Sentinel nime algus (nt S2A)
  sentinel_nimekiri$nimi
)

puudu_nimekiri <- sentinel_nimekiri |> filter(!file.exists(file_path))

if (nrow(puudu_nimekiri) > 0) {
  source("copsi_klass.R") # Eeldame, et copsi klass on siin
  s2class <- copsi$new()

  # Hankime esmased tokenid
  # NB! Tokenite kirjutamine kettale (c:/temp) on ebakindel.
  # Parem on hoida neid mälus s2class objektis.
  s2class$update_token(Sys.getenv("COPERNICUS_USERNAME"),
    Sys.getenv("COPERNICUS_PASSWORD"))

  pb <- progress_bar$new(format = " Laadimine [:bar] :current/:total (:per-
cent) eta: :eta", total = nrow(puudu_nimekiri))

  for (i in seq_len(nrow(puudu_nimekiri))) {
    pb$tick()
    rida <- puudu_nimekiri[i, ]

    # Looime kausta
    if (!dir.exists(dirname(rida$file_path))) {
      dir.create(dirname(rida$file_path), recursive = TRUE)
    }

    # Värskendame tokenit iga faili eel (või kontrolli aegumist)
    # Lihtsustuse mõttes teeme update iga kord, kuigi see pole kõige efektiiv-
sem
    s2class$update_token(Sys.getenv("COPERNICUS_USERNAME"),
      Sys.getenv("COPERNICUS_PASSWORD"))

    # Päring
    req_url <- glue::glue(
      "https://download.dataspace.copernicus.eu/odata/v1/Pro-
ducts({rida$guid})/$value"
    )

    tryCatch({
      httr2::request(req_url) |>

```

```

      httr2::req_auth_bearer_token(s2class$access_token) |>
      # Verbosity 0 on vaikne, progressi näitab meie väline loop
      httr2::req_perform(path = rida$file_path, verbosity = 0)
    }, error = function(e) {
      cat(glue::glue("\nViga faili {rida$nimi} laadimisel: {e$message}\n"))
    })
  }
} else {
  cat("Kõik failid on juba olemas.\n")
}

#
=====
# 3. MUDELI TREENIMINE (Päeva kaupa)
#
=====

# Laeme SMI andmed (veendu, et failid on olemas)
load("data/prtosa.RData")
load("data/prt.RData")

cat("Valmistan ette treeningpunktid...\n")

# Valmistame ette ruumiandmed (Ground Truth)
prt_valik <- prt |>
  filter(aasta >= cli_opt$aasta - 4, aasta <= cli_opt$aasta) |>
  select(aproovitykk_id, koord_n, koord_e, aasta) |>
  inner_join(
    prtosa |> select(aproovitykk_id, osa_nr, maakatgrp, maakategooria) |> fil-
ter(osa_nr == 0 &
maakategooria != "MM"),
    by = "aproovitykk_id"
  ) |>
  st_as_sf(
    coords = c("koord_e", "koord_n"),
    crs = 3301,
    remove = FALSE
  )

test_ala <- rast(ext(365000, 740000, 6375000, 6635000),
  res = 10,
  crs = "epsg:3301")
mets_sf <- read_sf(r"(\mm_workdir\etak_data\ETAK_EESTI_SHP\E_305_puittaimes-
tik_a.shp)")
rst_mets <- rasterize(mets_sf, test_ala, background = 0)
rst_mets_f <- focal(rst_mets, 5, sum) %% 25 == 0

prt_valik$mitte_piir <- extract(rst_mets_f, prt_valik)[, 2]
write_sf(prt_valik, "smi_valik.sqlite")
cat(glue::glue("Punkte kokku: {nrow(prt_valik)}\n"))

# Leiame kuupäeva-kaustad
sentinel_root <- file.path(workDir, "sentinel", cli_opt$aasta)
zip_files <- list.files(
  sentinel_root,
  pattern = "zip$",
  recursive = TRUE,
  full.names = TRUE
)
paev_dirs <- unique(dirname(zip_files))

```

```

# Funktsioon, mis jooksutatakse paralleelselt ühel zip failil
process_zip_training <- function(zip_path, prt_valik) {
  tryCatch({
    # 10m ribad
    rst_1 <- terra::rast(zip_path, subds = 1)
    meta_1 <- jsonlite::fromJSON(
      sf::gdal_utils(
        "info",
        sf::gdal_subdatasets(zip_path)$SUBDATASET_1_NAME,
        options = "-json",
        quiet = TRUE
      )
    )
    names(rst_1) <- meta_1$bands$metadata[[1]]$BANDNAME

    # 20m ribad
    rst_2 <- terra::rast(zip_path, subds = 2)
    meta_2 <- jsonlite::fromJSON(
      sf::gdal_utils(
        "info",
        sf::gdal_subdatasets(zip_path)$SUBDATASET_2_NAME,
        options = "-json",
        quiet = TRUE
      )
    )
    names(rst_2) <- meta_2$bands$metadata[[1]]$BANDNAME

    # Extract väärtused
    vrt_1 <- terra::extract(rst_1, prt_valik, ID = FALSE)
    vrt_2 <- terra::extract(rst_2, prt_valik, ID = FALSE)

    # Kombineeri
    combined <- cbind(vrt_1, vrt_2)
    combined$maakat <- prt_valik$maakatgrp

    # Tagasta ainult täielikud read
    return(na.omit(combined))
  }, error = function(e)
    return(NULL))
}

# Käime läbi päevade kataloogid
for (paev_dir in paev_dirs) {
  mudel_fail <- file.path(paev_dir, "randomForest.rds")

  if (!file.exists(mudel_fail)) {
    cat(glue::glue("Treenin mudelit kaustas: {basename(paev_dir)}\n"))

    zip_nimekiri <- list.files(paev_dir, "zip$", full.names = TRUE)

    # Paralleelne andmete lugemine
    daemons(min(length(zip_nimekiri), 4)) # Piirame tuumade arvu mõistlikult

    # Kogume andmed kõigist päeva piltidest
    data_list <- mirai_map(zip_nimekiri,
      process_zip_training,
      .args = list(prt_valik = subset(prt_valik,
mitte_piir)))$.progress]

    daemons(0) # Sulge workerid
  }
}

```

```

# Ühendame tulemused
test_data <- do.call(rbind, data_list)

if (is.null(test_data) || nrow(test_data) < 100) {
  cat("  Liiga vähe andmeid mudeli jaoks, jätan vahele.\n")
  next
}

# Filtreerimine (Pilved jms välja)
# SCL klassid: 4 (Veg), 5 (Non-Veg), 6 (Water)
valid_scl <- c(4, 5, 6)

# Veergude valik (Band 2-8, 11, 12)
cols_needed <- c("maakat", paste0("B", c(2:8, 11, 12)))

train_df <- test_data |>
  filter((as.integer(SCL) - 1) %in% valid_scl) |>
  filter(B4 > 0) |> # Lihtne vigaste pikslite eemaldamine
  select(all_of(cols_needed)) |>
  mutate(maakat = factor(maakat))

if (nrow(train_df) > 50) {
  # Treenime Random Forest mudeli
  # Ennustame: Kas on Mets (ME) või mitte
  rf_mud <- ranger::ranger(
    formula      = factor(maakat == "ME") ~ .,
    data         = train_df,
    num.trees     = 50,
    # Optimeerimiseks vähendatud puude arvu
    probability   = TRUE,
    num.threads  = 4      # Rangeri sisemine paralleelsus
  )

  save(rf_mud, file = mudel_fail)
  cat("  Mudel salvestatud.\n")
}
}

#
=====
# 4. KLASSIFITSEERIMINE (Ennustamine)
#
=====
cat("Alustan klassifitseerimist...\n")

eesti <- read_sf("data/eesti.gpkg")

# Dünaamiline orbiitide töötlemine (R136, R036, R079)
# Eeldame, et 'eesti' failis on veerud nimega R136, R036 jne, mis on 1/0
orbiidi_veerud <- grep("^R\\d+", names(eesti), value = TRUE)

epk_ruudud <- eesti |>
  st_drop_geometry() |>
  select(nr, wgrd_ruut, xmin, ymin, all_of(orbiidi_veerud)) |>
  tidyr::pivot_longer(
    cols = all_of(orbiidi_veerud),
    names_to = "orbiit",
    values_to = "on_orbiidis"
  ) |>

```

```

filter(on_orbiidis == 1) |>
select(-on_orbiidis)

# Leiame kõik zip failid ja seostame need ruutudega
zip_failid <- list.files(sentinel_root,
                        "zip$",
                        recursive = TRUE,
                        full.names = TRUE)

tootlus_nimekiri <- data.frame(zip_path = zip_failid) |>
  mutate(
    failinimi = basename(zip_path),
    wgrd_ruut = substr(failinimi, 39, 44),
    orbiit = substr(failinimi, 34, 37)
  ) |>
  inner_join(epk_ruudud,
            by = c("wgrd_ruut", "orbiit"),
            relationship = "many-to-many") |>
  mutate(
    tif_path = file.path(dirname(zip_path), "tiles", paste0(nr, ".tif")),
    olemas = file.exists(tif_path),
    # Mudeli fail peab olema zip failiga samas kaustas (vanemkaustas)
    mudel_path = file.path(dirname(zip_path), "randomForest.rds")
  )

# Filtreerime välja need, mis vajavad tegemist
tootlus_queue <- tootlus_nimekiri |>
  filter(!olemas) |>
  filter(file.exists(mudel_path)) # Ainult kui mudel on olemas

if (nrow(tootlus_queue) == 0) {
  cat("Kõik ruudud on juba klassifitseeritud või puuduvad mudelid.\n")
} else {
  # Jagame listiks mirai jaoks
  tiles_list <- split(tootlus_queue, seq(nrow(tootlus_queue)))

  tile_data <- tiles_list[[1]]

  # Töölisfunktsioon klassifitseerimiseks
  classify_worker <- function(tile_data) {
    library(terra)
    library(ranger)
    library(sf)
    library(jsonlite)

    # 1. Laeme mudeli (OLULINE: see peab toimuma tööli sees)
    load(tile_data$mudel_path)

    # 2. Rasterite lugemine (sarnane treenimisele)
    # 10m
    rst_1 <- terra::rast(tile_data$zip_path, subds = 1)
    meta_1 <- fromJSON(
      gdal_utils(
        "info",
        gdal_subdatasets(tile_data$zip_path)$SUBDATASET_1_NAME,
        options = "-json",
        quiet = T
      )
    )
    names(rst_1) <- meta_1$bands$metadata[[1]]$BANDNAME
  }
}

```

```

# 20m
rst_2 <- terra::rast(tile_data$zip_path, subds = 2)
meta_2 <- fromJSON(
  gdal_utils(
    "info",
    gdal_subdatasets(tile_data$zip_path)$SUBDATASET_2_NAME,
    options = "-json",
    quiet = T
  )
)
names(rst_2) <- meta_2$bands$metadata[[1]]$BANDNAME

# 3. Lõikame välja õige 5x5km tüki (Template)
ext_ruut <- ext(tile_data$xmin,
               tile_data$xmin + 5000,
               tile_data$ymin,
               tile_data$ymin + 5000)
rst_template <- rast(ext_ruut, res = 5, crs = "epsg:3301") # NB! Panin
res=10 (Sentinel native), mitte 5

# Project/Crop andmetest
rst_inp_1 <- project(rst_1, rst_template)
rst_inp_2 <- project(rst_2, rst_template)

# Teeme andmed tabeliks
# optimeerimine: töötle ainult valid piksleid
stack <- c(rst_inp_1, rst_inp_2)
df <- as.data.frame(stack, xy = TRUE, cells = TRUE)

# Filter SCL järgi (4,5,6)
valid_rows <- which(df$SCL %in% 4:6)

if (length(valid_rows) > 10) {
  data_inp <- df[valid_rows, ]

  # Ennustamine
  pred <- predict(rf_mud, data_inp)

  # Tulemused (tõenäosus * 100)
  vals <- as.integer(pred$predictions[, 2] * 100)

  # Kirjutame tulemused tühja rasterisse
  out_rst <- rast(rst_template)
  out_rst[] <- NA # Init with NA
  out_rst[data_inp$cell] <- vals

  # Salvestamine
  if (!dir.exists(dirname(tile_data$tif_path)))
    dir.create(dirname(tile_data$tif_path), recursive = T)

  writeRaster(
    out_rst,
    tile_data$tif_path,
    overwrite = TRUE,
    datatype = "INT1U",
    # 0-255
    gdal = c("COMPRESS=LZW", "TILED=YES")
  )
  return(TRUE)
}
return(FALSE)

```

```

}

# Käivitame mirai
daemons(n_max_workers_2) # Või n_max_workers_2, kui on defineeritud

cat(glue::glue("Klassifitseerin {length(tiles_list)} ruutu...\n"))

mirai_jobs <- mirai_map(tiles_list, classify_worker)

pb <- progress_bar$new(format = " Sentinel klassifitseerimine [:bar] :percent
:current/:total (:eta)", total = length(tiles_list))

print(status())

while (unresolved(mirai_jobs)) {
  stats <- status()$mirai
  if (!is.null(stats)) {
    pb$update(stats[["completed"]] / sum(stats))
  }
  Sys.sleep(0.5)
}

daemons(0)
cat("Töö valmis.\n")
}

```

LISA 2. LiDAR-i arvutamise kood

```
#
=====
# SEADISTUS JA PAKETID
#
=====

# Puhastame keskkonna, kui jooksutame interaktiivselt
if (interactive()) {
  rm(list = ls())
  gc()
  sapply(names(sessionInfo())$otherPkgs, function(x) {
    detach(paste0("package:", x), character.only = T)
  })
}

vorastiku_katvus <- 30

# Laeme vajalikud paketid (kasutame pacman'i mugavaks laadimiseks)
if (!require("pacman"))
  install.packages("pacman")
pacman::p_load(sf,
               terra,
               ranger,
               mirai,
               lidR,
               glue,
               jsonlite,
               httr2,
               dplyr,
               purrr,
               progress)

source("set_cli_params.R")

laz_nimekiri <- read_sf("data/epk2t_nimekiri.sqlite")
head(laz_nimekiri)

laz_nimekiri$laz_path <- with(laz_nimekiri,
                             file.path(
                               "",
                               "mm_workdir",
                               "laz_by_epk",
                               nr10000,
                               nr,
                               paste0(nr, "_", aasta, "_", tyypp, ".laz")
                             ))

laz_nimekiri$puudu <- !file.exists(laz_nimekiri$laz_path)

laadi_nimek <- subset(laz_nimekiri, puudu)
i <- 1
if (nrow(laadi_nimek) > 0) {
  url <- "https://geoportaal.maaamet.ee/index.php?lang_id=1&plugin_act=ot-
sing&kaardiruut={ruut}&andmetyypp=lidar_laz_{tyyp}&dl=1&f={fi-
lename}&no_cache=693bef75b46f8&page_id=614"
}
```

```

for (i in 1:nrow(laadi_nimek)) {
  cat(i, "\r")
  yks_fail <- laadi_nimek[i, ]
  ruut <- yks_fail$nr
  tyypp <- yks_fail$tyyp
  filename <- basename(yks_fail$laz_path)

  download.file(glue::glue(url),
                yks_fail$laz_path,
                mode = "wb",
                quiet = T)

}
}

las_nimek <- list.files(r"(H:\mm_workdir\laz_by_epk)",
                      "laz",
                      recursive = T,
                      full.names = T)

library(mirai)
daemons(0)

daemons(64)

tulemus <- mirai_map(las_nimek, function(las_rada) {
  output <- gsub("laz$",
                "tif",
                gsub(
                  "mm_workdir/laz_by_epk",
                  "crown_cover",
                  normalizePath(las_rada, winslash = "/")
                ))
  if (!file.exists(output)) {
    library(lidR)
    library(terra)
    set_lidr_threads(2)

    las <- readLAS(las_rada, select = "xyz", filter = "-drop_class 7 18")

    dtm <- crop(rast(r"(H:\01_metsamask\DTM_10m_eesti.tif)", ext(las))
    las_n <- filter_poi(las - dtm, Z > -2 & Z < 50)

    if (!dir.exists(dirname(output))) {
      dir.create(dirname(output), recursive = T)
    }

    writeRaster(
      template_metrics(
        las_n,
        list(
          crown_cover = mean(Z > 2) * 100,
          p80 = quantile(Z[Z > 2], 0.8) * 10
        ),
        rast(round(ext(las), -1), res = 5, crs = "epsg:3301")
      ),
      output,
      datatype = "INT2U",

```

```
        overwrite = T,  
        gdal = c("COMPRESS=LZW"),  
        NAflag = 0  
    )  
}  
return(0)  
  
}) [.progress]  
  
daemons(0)
```

LISA 3. Tulemuse kokku panemise kood

```
#
=====
# 0. SEADISTUS
#
=====

# Puhastame keskkonna
if (interactive()) {
  rm(list = ls())
  gc()
  # Eemaldame laetud paketid, et vältida konflikte
  if (!is.null(sessionInfo()$otherPkgs)) {
    sapply(names(sessionInfo()$otherPkgs), function(x) {
      detach(paste0("package:", x), character.only = TRUE)
    })
  }
}

#
=====
# 1. LOGIMISE FUNKTSIOON
#
=====

# Abifunktsioon teadete väljastamiseks koos kellaajaga
logi_teade <- function(teade) {
  cat(sprintf("[%s] %s\n", format(Sys.time(), "%H:%M:%S"), teade))
}

logi_teade("--- SKRIPT ALUSTAB TÖÖD ---")

# Paketid
logi_teade("Laen vajalikud paketid...")
library(sf)
library(terra)
library(glue)

# -----
#
# KONSTANDID JA RAJAD
# -----

logi_teade("Seadistan konstandid ja rajad...")
# Rajad
PATH_TEMP      <- "tmp"
PATH_ETAK      <- r"(H:\mm_workdir\etak_data\ETAK_EESTI_SHP)"
PATH_OUTPUT    <- "kihid/2025"
PATH_PUUDU     <- "andmed/eesti_puudu.gpkg"

# Rastrite seadistus
CRS_EST        <- "epsg:3301"
RASTER_EXTENT  <- ext(365000, 740000, 6375000, 6635000)
RASTER_RES     <- 5
RASTER_TYPE    <- "INT1U"
GDAL_OPTS      <- c("COMPRESS=LZW", "TILED=YES")
```

```

# Puhastame ja seadistame temp kausta
logi_teade("Valmistan ette ajutise (temp) kausta...")
if (!dir.exists(PATH_TEMP)) dir.create(PATH_TEMP, recursive = TRUE)
unlink(list.files(PATH_TEMP, full.names = TRUE))
terraOptions(
  tempdir = PATH_TEMP,
  datatype = RASTER_TYPE,
  todisk = T,
  verbose = F,
  #memfrac = 0.2
  memmax = 5
)

# Looime väljundkausta
if (!dir.exists(PATH_OUTPUT)) {
  logi_teade(glue("Loon väljundkausta: {PATH_OUTPUT}"))
  dir.create(PATH_OUTPUT, recursive = TRUE)
}

# -----
#
# ABIFUNKTSIOONID
# -----
#

arvuta_uus_kood <- function(kood) {
  kood %/% 100 * 10 + kood %% 100
}

#
=====
# 2. PÕHIRASTRI LOOMINE JA PINNAOBJEKTID
#
=====

logi_teade("Initsialiseerin tühja põhirastri...")
rst <- rast(RASTER_EXTENT, res = RASTER_RES, crs = CRS_EST)

# Leiame kõik "_a.shp" lõpuga failid
kihid <- list.files(PATH_ETAK, pattern = "_a\\.shp$", full.names = TRUE)
logi_teade(glue("Leidsin {length(kihid)} pinnakihti töötlemiseks."))

koond_andmed <- NULL

logi_teade("--- Alustan pinnaobjektide rasterdamist ---")

for (kiht in kihid) {
  failinimi <- basename(kiht)
  nimi_ilma_laiendita <- tools::file_path_sans_ext(failinimi)

  logi_teade(glue("Töötlen faili: {failinimi}"))

  # Loeme sisse
  query_sql <- glue("SELECT kood FROM {nimi_ilma_laiendita}")
  kiht_andm <- read_sf(kiht, query = query_sql)

  # Arvutame uue koodi ja rasterdame
  if (nrow(kiht_andm) > 0) {
    kiht_andm$kood_u <- arvuta_uus_kood(kiht_andm$kood)
    logi_teade(glue("-> Rasterdan {nrow(kiht_andm)} objekti..."))
    rst <- rasterize(kiht_andm, rst, field = "kood_u", update = TRUE)
  }
}

```

```

# Statistika
pindala <- as.numeric(sum(st_area(kiht_andm)))
koond_andmed <- rbind(koond_andmed, data.frame(kood = kiht_andm$kood_u[1],
pindala = pindala / 1E4))
} else {
  logi_teade(" -> Hoiatus: Fail on tühi, liigun edasi.")
}
}

#
=====
# 3. ERANDID JA LISAKIHID (HOONED, PUUDUVAD ALAD)
#
=====

# Eesti maismaast välja jääva osa nullimine
if (file.exists(PATH_PUUDU)) {
  logi_teade("Töötlen 'puuduvaid alasid' (välised alad)...")
  liigne <- read_sf(PATH_PUUDU)
  liigne$kood_u <- 0
  rst <- rasterize(liigne, rst, field = "kood_u", update = TRUE)
  logi_teade(" -> Välised alad nullitud.")
} else {
  logi_teade("INFO: Puuduvate alade faili ei leitud, jätan vahele.")
}

# Hoonete lisamine (E_401)
logi_teade("Töötlen hooned (E_401)...")
fail_hoone <- file.path(PATH_ETAK, "E_401_hoone_ka.shp")

if (file.exists(fail_hoone)) {
  nimi_hoone <- tools::file_path_sans_ext(basename(fail_hoone))
  hoone_andm <- read_sf(fail_hoone, query = glue("SELECT kood FROM
{nimi_hoone}"))
  hoone_andm$kood_u <- arvuta_uus_kood(hoone_andm$kood)

  logi_teade(glue(" -> Lisan {nrow(hoone_andm)} hoonet rastrisse..."))
  rst <- rasterize(hoone_andm, rst, field = "kood_u", update = TRUE)
} else {
  logi_teade("VIGA: Hoonete faili ei leitud!")
}

#
=====
# 4. JOONOBJEKTID (TEED JA VOOLUVEEKOGUD)
#
=====

process_linear_feature <- function(failitee, sql_query, buffer_size, ras-
ter_obj, target_val = 35) {
  failinimi <- basename(failitee)
  logi_teade(glue("Töötlen joonobjekti: {failinimi}"))

  if (!file.exists(failitee)) {
    logi_teade(glue(" -> VIGA: Faili {failinimi} ei leitud!"))
    return(raster_obj)
  }

  logi_teade(" -> Loen andmeid...")
  andmed <- read_sf(failitee, query = sql_query)

```

```

andmed$kood_u <- arvuta_uus_kood(andmed$kood)

logi_teade(glue(" -> Puhverdan {nrow(andmed)} objekti ({buffer_size}m)..."))
puhverdatud <- st_buffer(andmed, buffer_size)

logi_teade(" -> Rasterdan ajutisse kihti...")
kiht_kood <- andmed$kood_u[1]
line_rast <- rasterize(puhverdatud, raster_obj, field = "kood_u")

logi_teade(glue(" -> Maskeerin (kirjutan üle vaid väärtuse {target_val}
peal)..."))
mask_logic <- (line_rast == kiht_kood) & (raster_obj == target_val)

raster_obj[mask_logic == 1] <- kiht_kood
logi_teade(" -> Valmis.")
return(raster_obj)
}

# Teed (E_501)
rst <- process_linear_feature(
  failitee = file.path(PATH_ETAK, "e_501_tee_j.shp"),
  sql_query = "SELECT kood FROM e_501_tee_j WHERE laius >= 3 AND laius <= 8",
  buffer_size = 5,
  raster_obj = rst
)

# Vooluveekogud (E_203)
rst <- process_linear_feature(
  failitee = file.path(PATH_ETAK, "E_203_vooluveekogu_j.shp"),
  sql_query = "SELECT kood FROM E_203_vooluveekogu_j WHERE laius >= 20 AND
laius <= 40",
  buffer_size = 5,
  raster_obj = rst
)

# Salvestame vahetulemuse
output_orig <- file.path(PATH_OUTPUT, "etak_kood_orig.tif")
logi_teade(glue("Salvestan vahetulemuse: {output_orig}"))
writeRaster(rst, output_orig, gdal = GDAL_OPTS, datatype = RASTER_TYPE, NAflag
= 0, overwrite = TRUE)

#
=====
# 5. ELEKTRILIINID (ERILINE PUHVRILOOGIKA)
#
=====

logi_teade("Töötlen elektriliine (E_601)...")
fail_elekter <- file.path(PATH_ETAK, "E_601_elektriliin_j.shp")

if (file.exists(fail_elekter)) {
  logi_teade(" -> Loen ja filtreerin elektriliine...")
  elekter_andm <- read_sf(fail_elekter, query = "SELECT kood, nimipinge FROM
E_601_elektriliin_j")
  elekter_andm <- subset(elekter_andm, !is.na(nimipinge))

  elekter_andm$kood_u <- arvuta_uus_kood(elekter_andm$kood)

  # Puhvri määramine
  elekter_andm$puhver_raadius <- 5

```

```

elekter_andm$puhver_raadius[elekter_andm$nimipinge >= 35] <- 12.5
elekter_andm$puhver_raadius[elekter_andm$nimipinge >= 220] <- 20

logi_teade("  -> Puhverdan ja rasterdan (update=TRUE)...")
etak_mod <- rasterize(st_buffer(elekter_andm, elekter_andm$puhver_raadius),
rst, field = "kood_u", update = TRUE)

output_mod <- file.path(PATH_OUTPUT, "etak_kood_mod.tif")
logi_teade(glue("Salvestan elektriliinidega rastri: {output_mod}"))
writeRaster(etak_mod, output_mod, gdal = GDAL_OPTS, datatype = RASTER_TYPE,
NAflag = 0, overwrite = TRUE)
} else {
  logi_teade("VIGA: Elektriliinide faili ei leitud!")
  etak_mod <- rst # Jätkame ilma elektriliinideta
}

#
=====
# 6. JÄRELTOÖTLUS (LIDAR JA S2)
#
=====

logi_teade("Vabastan mälu (gc)...")
gc()

logi_teade("Alustan järeltöötlust (LiDAR ja Sentinel)...")

fail_lidar <- file.path(PATH_OUTPUT, "lidar_cc_med.tif")
fail_msi    <- file.path(PATH_OUTPUT, "S2_2025_med.tif")

if (file.exists(fail_lidar) && file.exists(fail_msi)) {

  logi_teade("  -> Loen LiDAR ja Sentinel maskid sisse...")
  lidar_mask <- rast(fail_lidar) >= 30
  msi_mask   <- rast(fail_msi) >= 50

  # 1. Korrektsioon
  logi_teade("  -> Korrektsioon 1: Koodid 36,37 + LiDAR kõrged -> 35")
  etak_mod[etak_mod %in% c(36,37) & lidar_mask == 1] <- 35

  out_s1 <- file.path(PATH_OUTPUT, "etak_kood_mod_s1.tif")
  logi_teade(glue("  -> Salvestan etapp 1: {out_s1}"))
  writeRaster(etak_mod, out_s1, gdal = GDAL_OPTS, datatype = RASTER_TYPE,
NAflag = 0, overwrite = TRUE)

  # 2. Korrektsioon
  logi_teade("  -> Korrektsioon 2: Koodid 33,34 + LiDAR kõrged + MSI kõrged ->
35")
  on_lage <- etak_mod %in% c(33, 34)
  etak_mod[on_lage & lidar_mask & msi_mask] <- 35

  out_s2 <- file.path(PATH_OUTPUT, "etak_kood_mod_s2.tif")
  logi_teade(glue("  -> Salvestan lõpptulemuse: {out_s2}"))
  writeRaster(etak_mod, out_s2, gdal = GDAL_OPTS, datatype = RASTER_TYPE,
NAflag = 0, overwrite = TRUE)
  gc()

  #etak_mod <- rast(out_s2)

  logi_teade(glue("  -> Filtreerin metsamaa"))

```

```

metsamaa <- etak_mod == 35
logi_teade(glue(" -> Eemaldan augud < 0.1 ha"))
metsamaa_s <- sieve(metsamaa, 41, 4)
etak_mod[metsamaa & metsamaa_s == 0] <- 38

out_s3 <- file.path(PATH_OUTPUT, "etak_kood_mod_s3.tif")
logi_teade(glue(" -> Salvestan lõpptulemuse: {out_s3}"))
writeRaster(etak_mod, out_s3, gdal = GDAL_OPTS, datatype = RASTER_TYPE,
NAflag = 0, overwrite = TRUE)

if(!"etak_mod" %in% ls()){
  etak_mod <- rast(out_s3)
}

#=====
logi_teade("Töötlen eelise pargid ...")
fail_pargid <- file.path(PATH_ETAK, "eelis_pargid.shp")

hoone_andm <- read_sf(fail_pargid)
hoone_andm$kood_u <- 38

logi_teade(glue(" -> Lisan {nrow(hoone_andm)} hoonet rastrisse..."))
etak_mod <- rasterize(hoone_andm, etak_mod, field = "kood_u", update = TRUE)

out_s4 <- file.path(PATH_OUTPUT, "etak_kood_mod_s4.tif")
logi_teade(glue(" -> Salvestan lõpptulemuse: {out_s4}"))
writeRaster(etak_mod, out_s4, gdal = GDAL_OPTS, datatype = RASTER_TYPE,
NAflag = 0, overwrite = TRUE)

jagunemine <- freq(etak_mod)

jagunemine$spindala <- jagunemine$count * 0.0025

mets_sf <- as.polygons(ifel(etak_mod == 35, 1, NA))

unlink(file.path(PATH_OUTPUT, "metsamaa_s1.sqlite"))

writeVector(mets_sf, file.path(PATH_OUTPUT, "metsamaa_s1.sqlite"), filetype =
"SQLite")

mets_sf <- read_sf(file.path(PATH_OUTPUT, "metsamaa_s1.sqlite"))
mets_sf_m <- st_cast(mets_sf, "POLYGON")

write_sf(mets_sf_m, file.path(PATH_OUTPUT, "metsamaa_s2.sqlite"))

logi_teade("Järeltöötlus valmis!")

} else {
  logi_teade("HOIATUS: LiDAR või Sentinel failid puuduvad, jätan järeltöötluste
vahele.")
}

logi_teade("--- SKRIPT LÕPETAS EDUKALT ---")

```

LISA 4. Käsurea parameetrite skript

```
suppressPackageStartupMessages(library("optparse"))

option_list <- list(
  make_option(
    c("-v", "--verbose"),
    action = "store_true",
    default = TRUE,
    help = "Print extra output [default]"
  ),
  make_option(
    c("-q", "--quietly"),
    action = "store_false",
    dest = "verbose",
    help = "Prindi väljund"
  ),
  make_option(
    c("-o", "--overwrite"),
    action = "store_false",
    dest = "overwrite",
    default = F,
    help = "Kirjutab üle"
  ),
  make_option(
    c("-a", "--aasta"),
    type = "integer",
    default = as.numeric(substr(Sys.Date(), 1, 4)),
    help = "Arvutuse aasta [vaikimisi %default]",
    metavar = "number"
  ),
  make_option(c("-z", "--zip_path"), help = "Ühe Sentinel-2 zip faili rada")
)

cli_opt <- parse_args(OptionParser(option_list = option_list))

n_max_workers_2 = as.integer(parallel::detectCores() / 2)
n_max_workers_3 = as.integer(parallel::detectCores() / 3)
workDir = file.path(substr(getwd(), 1, 2), "mm_workdir")

dtm_inp_path <- file.path(getwd(), "data", "DTM_10m_eesti.tif")
tif_out_path <- file.path(workDir, "tif_data")
mtr_out_path <- file.path(workDir, "lidar_meetrik")
laz_inp_path <- file.path(workDir, "laz_data")
etk_inp_path <- file.path(workDir, "etak_data")
```