

Rīgas Tehniskās universitātes

Inženierzinātņu vidusskola

**HEBO algoritma pielietošana optimālas hiperparametru konfigurācijas meklēšanai**

Zinātniskās pētniecības darbs informātikas nozarē

**Darba autors:**

Georgiy Furanets

11. klases skolnieks

**Darba vadītājs:**

Artūrs Ķempelis

M.Sc.Ing., RTU Inženierzinātņu vidusskolas informātikas skolotājs

Rīga 2025

**Anotācija**

Darba tēma ir “HEBO algoritma pielietošana optimālas hiperparametru konfigurācijas meklēšanai”, darba autors - RTU Inženierzinātņu vidusskolas 11. klases skolnieks Georgiy Furanets, darba vadītājs - RTU Inženierzinātņu vidusskolas informātikas skolotājs Artūrs Ķempelis.

Darba mērķis bija analizēt Heteroskedastiskās evolucionārās Bajesa optimizācijas (HEBO) algoritma veiktspēju neironu tīklu optimizācijā. Saistībā ar vispasaulīgo atmežošanu, neironu tīkls tika apmācīts ar datu kopu “Covertype”, kas satur informāciju par Amerikas Savienoto Valstu mežiem. HEBO algoritma veiktspēja tika salīdzināta ar Nejaušas meklēšanas algoritmu.

Pēc eksperimentālas daļas tika konstatēts, ka HEBO algoritma veiktspēja pie neliela iterāciju daudzuma ir lidzīga RS algoritmam, tāču, pieaugot iterāciju skaitām no 10 līdz 100, neironu tīkla informācijas apstrādes precizitāte paaugstinās par 8%.

**Atslēgvārdi:** Datorredze, hiperparametri, atmežošana, konvoluciju neironu tīkli, Heteroskedastiskā evolucionārā Bajesa optimizācija.

**Abstract**

The topic of the work is “Application of the HEBO Algorithm for Finding Optimal Hyperparameter Configurations,” authored by Georgiy Furanets, an 11th-grade student at the RTU Engineering High School, and supervised by Artūrs Ķempelis, an informatics teacher at the RTU Engineering High School.  
 The goal of the work was to analyze the performance of the Heteroscedastic Evolutionary Bayesian Optimization (HEBO) algorithm in optimizing neural networks. In relation to global deforestation, the neural network was trained using the “Covertype” dataset, which contains information about forests in the United States. The performance of the HEBO algorithm was compared to that of the Random Search (RS) algorithm.  
 Following the experimental part, it was observed that the performance of the HEBO algorithm with a small number of iterations is similar to the RS algorithm. However, as the number of iterations increases from 10 to 100, the accuracy of neural network data processing improves by 8%.

**Keywords:** Computer vision, hyperparameters, deforestation, convolutional neural networks, Heteroscedastic Evolutionary Bayesian Optimization.

**Saturs**

[Ievads 5](#_3znysh7)

[I.](#_2et92p0) Literatūras apskats 6

[1.1 Heteroskedastiskā evolucionārā Bajesa optimizācija (HEBO) 6](#_tyjcwt)

[1.2. Nejauša meklēšana (RS) 7](#_3dy6vkm)

[1.3. Literatūras apraksta secinājumi 7](#_1t3h5sf)

[2. Praktiskā daļa. 8](#_4d34og8)

[2.1. “Covertype” datu kopa 8](#_2s8eyo1)

[2.2. Rezultātu apstradāšana 8](#_17dp8vu)

[2.3. Rezultāti 9](#_3rdcrjn)

[Secinājumi 13](#_26in1rg)

[Turpmākie pētījumi 14](#_lnxbz9)

[Izmantotie literatūras avoti un atsauces 15](#_35nkun2)

[Pielikumi 16](#_1ksv4uv)

# Ievads

Informātikas nozarei attīstoties pēdējo dažu gadu desmitu laikā, tās panākumi tiek bieži pielietoti ar datoriem nesaistītās zinātnēs. Komplicēto vienādojumu atrisināšanai un paradību simulēšanai ir nepieciešamas datorzinātņu prasmes, un, lai programmas varētu efektīvi izmantot, tās ir jāoptimizē. Līdz ar to, daudzi mūsdienu pētījumi informātikas nozarē ir saistīti tieši ar algoritmu optimizācijas problēmu.

Mūsdienās neironu tīkli tiek bieži izmantoti ekoloģijas jomā. Tie ļauj analizēt lielu datu daudzumu ātri un efektīvi salidzīnājumā ar cilvēkiem. Viena no aktuālākajām ekoloģijas problēmām ir kopējā pasaules atmežošana. Daži eksperti uzskata, ka šīs process ir sācies ar brīdi, kad cilvēce atklāja uguni, un turpinās vēljoprojām [1]. 1990. gadā katru gadu tika zaudēti 12 miljoni hektāri meža [1]. Saistībā ar klimata izmaiņām mēdz notikt arī anomāli atmežošanas gadījumi, kā, piemēram, Brazīlijas mežu platības samazināšanas 2000. gadu sākumā [1], un, spriežot pēc jaunākajiem datiem, šī tendence manāma visās pasaules daļās.

Neironu tīklu precizitāti nosaka tā apmācīšanas rezultāti, kurus savukārt nosaka pirms apmācīšanas izvēlētā hiperparametru konfigurācija. Šiem nolūkiem tiek izstradāti dažādi algoritmi, kas spēj pēc iespējas mazākajā laikā un ar mazāko apmācīšanas iterāciju skaitu noteikt vispiemerotāku hiperparametru konfigurāciju. Daži no bieži izmantotiem algoritmiem ir Režģa Meklēšana (GS), Kokveida Parzena izvērtēšana (TPE), Nejauša meklēšanna (RS) un citi [2]. Savukārt šājā darbā tika pētīts Heteskedastiskās evolucionāras Bajesa optimizācijas (HEBO) algoritms, kas ir ieņēmis 1. vietu 2020. gada *“NeurIPS Black-Box Optimisation Challenge”* konkursā [3] un strauji uzņem popularitāti starp citiem optimizācijas algoritmiem. Tā efektivitāte tiek salīdzināta ar Nejaušas meklēšanas (RS) algoritmu, kas ir vienkaršāks implementācijā un rada labus rezultātus pie neliela datu daudzuma, tāpēc ir bieži izmantots praktisku uzdevumu risināšanā.

**Pētijuma mērķis:**

Salīdzināt HEBO algoritma veiktspēju neironu tīklu precizitātes maksimizācijā, salīdzinot ar RS algoritmu.

**Hipotēze:**

Neironu tīkls, kura apmacīšanas procesā tika izmantots HEBO algoritms, radīs salīdzinoši augstāku precizitāti attēlu atpazīšanā, nekā RS algoritms.

**Darba uzdevumi**:

1. Iepazīties ar raytune bibliotēku, kas ir nepieciešama neironu tīklu apmacīšanai
2. Izmantot iegūto datu kopu, kas satur informāciju par mežiem, katrā no algoritmiem neironu tīkla apmacīšanai
3. Salīdzināt algoritmu veiktspēju un attēlot tos grafiski

# Literatūras apskats

Literatūras apskatā tiek analizētā eksistejoša literatūra par diviem algoritmiem, kas tiek izmantoti optimālas hiperparametru kombinācijas meklēšanai neironu tīklos - Heteroskedastiskā evolucionārā Bajesa optimizācija (HEBO) un Nejaušā meklēšana (Random Search).

## 1.1 Heteroskedastiskā evolucionārā Bajesa optimizācija (HEBO)

Heteroskedastiskā evolucionārā Bajesa optimizācija jeb HEBO ir mašīnmacīšanās algoritms, kas tiek izmantots neironu tīklu optimizācijas uzdevumos. Šīs algoritms ir atvasināts no Bajesa optimizācijas, kas tiek izmantota melnas kastes problēmu risināšanai [3]. Melnas kastes problēmas iekļauj sevī tādu sistēmu pētīšanu, kuru iekšējā struktūra un matemātiskais aparāts nav zināmi. Visparīgā gadījumā melnās kastes uzdevums var tikt formulēts sekojoši:

argmax x ∈ X f(x) [3]

kur X - visu iespējamu neironu tīklu raksturojošo hiperparametru konfigurāciju kopa jeb meklēšanas telpa (search space), x - specifiskā hiperparametru konfigurācija (x∈X), un f(x) - funkcija, kas ir atkarīga no hiperparametru konfigurācijas un nosaka algoritma efektivitāti, pie tām f : X → R [3].

Bajesa optimizācijas pamatā ir Gausa process (GP), kas ir no statistikas nozares aizgūts paņēmiens. Tā ideja ir katrā algoritma iterācijā maksimizēt kādas sistēmu raksturojošas funkcijas f(x) vērtību, kas katrai hiperparametru konfigurācijai aprēķina varbutību, ka tieši izvelētā konfigurācija ir melnās kastes problēmas risinājums [3]. Ja tradicionālo Bajesa optimizāciju pielieto sistēmām, kuru troksnis ir nemainīgs, tad Heteroskedastiskā Bajesa optimizācija (HBO) tiek izmantota pretējos gadījumos, aprakstot sistēmas troksni kā funkciju no ievada datiem. Pēdējā atšķirība starp HEBO un BO ir evolucionāru stratēģiju izmantošana [3]. Tas ļauj algoritmam meklēt konfigurācijas meklēšanas telpā globālāk, kas ir vitāli sakotnējajās algoritma iterācijās.

HEBO algoritma darbošanas instrukcija izskātas sekojoši:

1. Pirmajā algoritma iterācijā tiek izvēlēta patvaļīga hiperparametru kombinācija no meklēšanas telpas, kurai tiek aprēķināta modeļa precizitāte
2. Izvēlētai hiperparametru konfigurācijai tiek palaists Gausa process
3. Sistēmu raksturojošā funkcija izvēlas nakamo hiperparametru konfigurāciju, balstoties uz Gausa procesa izveidoto paredzējumu
4. Tiek atkartoti 2. līdz 4. solis

HEBO ir parādījis augstu efektivitāti neironu tīklu precizitātes maksimizēšanā, līdz ar to tas ir plaši izmantots daudzos praktiskajos uzdevumos, kuros ir nepieciešams precīzi un efektīvi apstradāt lielu datu daudzumu. Piemēram, HEBO algoritms tika izmantots, lai analizētu enerģijas plūsmu celtnēs [4].

## 1.2. Nejaušā meklēšana (RS)

Nejaušās meklēšanas (*Random Search*) algoritms ir bieži pielietots optimālas hiperparametru konfigurācijas meklēšanai. Tā ir plaši izmantota, pateicoties savai vienkaršībai pielietojumā un efektivitātei, stradājot ar salīdzinoši lielu datu daudzumu [5]. Lidzīgi kā Bajesa optimizācija, Nejaušās meklēšanas algoritms arī ir izmantots melnās kastes uzdevumu risināšanai [5]. Tā problēma var tikt formulēta sekojošājā veidā:

argmin 𝜃 ∈ 𝛩 f(𝜃) [5]

Kur 𝜃 ir hiperparametru konfigurācija, 𝛩 ir visu iespējamo konfigurāciju kopa, un f(𝜃) ir funkcija, kas apraksta algoritma rezultātus, pie tam 𝑓 : 𝛩 → 𝑅 [5].

Nejaušās meklēšanas algoritma darbošanās principu var aprakstīt sekojošos soļos, kas tiek atkārtoti katrā algoritma darbošanās iterācijā [5]:

1. Algoritms izvēlas pieejamu hiperparametru konfigurāciju 𝜃 no iespējamo konfigurāciju kopas 𝛩
2. Tiek aprēķināta algoritma efektivitāte izvēlētai konfigurācijai
3. Strateģija tiek atjaunināta, balstoties uz iepriekšējo iterāciju rezultātiem

Sākotnēji Nejaušas meklēšanas algoritms bija domāts izmantošanai sistēmām, kurās nav trokšņu [5]. Taču tika konstatēts, ka šī metode var tikt izmantota visām sistēmām, pat ar radītiem trokšņiem. Ņemot vērā šīs meklēšanas metodes efektivitāti optimizācijas uzdevumu risināšanā, tā tiek bieži pielietota praktisko uzdevumu risināšanai. Par piemēru var kalpot pētījums, ko zinātnieki ir veikuši Spānijā, Baskovas provincē [6]. Šī pētījuma laikā tika pētīta ūdens kvalitāte ūdens tilpēs, izmantojot neironu tīklu, kas tika apmācīts ar RS [6]. Pēc šī pētījuma rezultātiem ir skaidrs, ka Nejaušās meklēšanas algoritms rada labus rezultātus liela datu komplekta apstradāšanā [6].

## 1.3. Literatūras apraksta secinājumi

Mūsdienās neironu tīkli tiek bieži pielietoti dažāda veida datu analīzei, kas palīdz praktisku problēmu risināšanā. Neironu tīklu datu apstrādes precizitāti nosaka modeļa hiperaprametru konfigurācija. Tā ir noteikta pirms apmācīšanas procesa, un šiem nolūkiem tiek izmantoti dažādi optimizācijas algoritmi.

Šajā darbā tiek analizēti divi algoritmi, primāri Heteroksedastiskā evolucionārā Bajesa optimizācija (HEBO), kura efektivitāte tiek salidzināta ar Nejaušas meklēšanas (RS) algoritma efektivitāti. Abi algoritmi specializējas melnās kastes problēmu risināšana, kas raksturo sistēmas bez acimredzama matemātiskā apraksta. Salīdzinājumam izmanto skalāro F1 vērtību, kas ir nosākama no apjukuma matricas.

Dažādi optimizācijas algoritmi tiek bieži izmantoti praktisku uzdevumu risināšanai. Tie ir vitāli svarīgi jebkādām problēmām, kurās nepieciešama liela datu daudzuma efektīva un ātra analīze, piemēram, mežu situāciju konstanta novērošana.

# 2. Praktiskā daļa.

Eksperimentālajā daļā tika praktiski salīdzināti iepriekš aprakstīti algoritmi. Apmācīto modeļu precizitātes tika salīdzinātas, izmantojot skalāro F1 vērtību.

Kods ir aizgūts no eksistējoša projekta (skat. 1. pielikumu). Konkrēti neironu tīklu apmācīšanai ar “Covertype” datu kopu, pielietojot HEBO un RS algoritmus, tika izmantots autora modificēts oriģinālais kods (skat. 2. pielikumu).

Programmā abi algoritmi tiek pielietoti vienam neironu tīklam, lai notiktu godīga veiktspējas salīdzināšana. Modelis, kas tiek apmacīts smadzeņu audzēja attēlu noteikšanai, ir RFCE neironu tīkls, kas tiek ieviests no Python “Ray” bibliotēkas. Programmā tiek ieviests mežu attēlu datubāzes kontrolieris, kas sagatvo attēlus apmacīšanas procesām. Izmantojot “Ray” bibliotēku, tiek ieviesti arī HEBO un RS algoritmi, kas tiek pielietoti modelim katrā programmas solī.

Programmas izpildes laikā tiek palaista salīdzināšanas funkcija, kas izpilda sekojošu instrukciju komplektu:

1. Programma iziet cauri datubāzei un sagatavo meža attēlus apmacīšanas procesam
2. Neironu tīkls tiek apmācīts mežu veida noteikšanai, izmantojot attiecīgu hiperparametru optimizēšanas algoritmu
3. Programma izved neironu tīkla F1 skalāru vērtību pēc apmācīšanas

Pēc programas izpildes rezultāti tiek apkopoti to turpmākai analīzei.

## 2.1. “Covertype” datu kopa

HEBO algoritma veiktspēja tika pārbaudīta uz “Covertype” datu kopas. Tā sastāv no 581012 apmācīšanas un testēšanas datiem. Šī datu kopa tika izveidota 1998. gadā un satur informāciju par Amerikas Savienoto Valstu mežiem [7].

“Covertype” ir klasisks vairāku klašu datu kopas piemērs. Katram datu komplektam atbilst vērtības, kas apraksta kāda meža apgabala augstumu virs jūras līmeņa, virsmas slīpumu, augsnes veidu un citus parametrus, kas ir nepieciešami pašreizējās mežu ekoloģiskas situācijas izvērtēšanai.

## 2.2. Rezultātu apstradāšana

Pēc abu optimizācijas algoritmu pielietošanas neironu tīklam ir nepieciešams veids kā skaidri un viegli salīdzināt katra algoritma rezultātus. Šiem nolūkiem tiek izmantotas neskaidrības matricas, uz kuriem tiek attēloti apmācīšanas procesa rezultāti [8]. Uz vienas ass atrodas n vērtības, kas parāda, cik katra veida rezultātu atbilst kādai no n iespējamām grupām (šajā gadījumā - mežu raksturojoši parametri), taču uz otras ass - faktiskie rezultāti. Perfektajā gadījuma visas ne nulles vērtības atrodas uz matricas galvenas diagonāles, savukārt parējās vērtības ir vienādas ar nulli [8].

No neskaidrības matricas rezultātiem ir iespējams aprēķināt skalāro F1 vērtību [8]. Tā tiek izmantota optimizācijas algoritmu veiktspējas salīdzināšanā un ir tieši proporcionāla algoritma sekmēm. Binārai vērtībai neskaidrības matricai ir izmēri 2 x 2 un tā sastāv no 4 vērtībām [8]:

*TP - Patiess pozitīvs*

*FP - Aplams pozitīvs*

*TN - Patiess negatīvs*

*FN - Aplams negatīvs*

Tad F1 vērtība ir aprēķināma sekojoši:

*Precizitāte = TP / (TP + FP) [8]*

*Atsaukums = TP / (TP + FN) [8]*

*F1 = (2 \* Precizitāte \* Atsaukums) / (precizitāte + Atsaukums)*

Lielākam pazīmju skaitām F1 vērtību var aprēķināt pēc lidzīga principa, taču aprēķini ir sarežģītāki. Zemāk ir redzams piemērs neskaidrības matricai, kas satur informāciju par 3 klašu datu kopas rezultātiem.

A screenshot of a chart

Description automatically generated

1. *attēls. Vairāku (triju) klašu neskaidrības matrica (Autora tulkojums) [8].*

## 2.3. Rezultāti

Rezultāti, kas tika izmantoti HEBO un RS algoritmu precizitātes salīdzināšanai, ir iegūti no darba autora veiktā eksperimenta.

  HEBO algoritma darbošanās princips paredz, ka apmācīšanas procesa laikā tas mācās no viena procesa ietvaros iepriekš iegūtiem rezultātiem. Līdz ar to, tā veiktspēja ir atkarīga no apmācīšanas iterāciju skaita. Viens solis sastāv no 6 apmācīšanas procesiem - gan ar HEBO, gan ar RS tiek apmācīti 3 neironu tīkli, ar 10, 50 un 100 apmācīšanas iterācijām. Šī izvele ir pamatota ar to, ka šos skaitļus ir viegli atcerēties, tie ir apaļi un vēsturiski šī kombinācija tika izmantota neironu tīklu apmācīšanā. Pēc katra procesa vienam no sešiem apmācīšanas procesiem tiek aprēķināta vidējā F1 vērtība un piefiksēta tabulā. Lai iegūtu objektīvus datus, visi 6 apmācīšanas procesi tika īstenoti 10 reizes, kopā - 60 procesi.

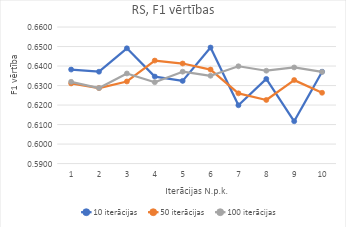
Iegūtie rezultāti ir attēloti 3 tabulās un grafikos - F1 vērtības salīdzināšana HEBO algoritmam ar dažādu iterāciju skaitu, F1 vērtības salīdzināšana RS algoritmam ar dažādu iterāciju skaitu un HEBO salīdzināšana ar RS algoritmu. Grafikos atsevišķi rezultāti ir savienoti ar līnijām, kas parada algoritma veiktspējas tendences visu apmācīšanas procesu laikā.

Sakotnēji tika palaistas 30 simulācijas ar RS algoritmu, un to rezultāti ir redzami 1. tabulā.

1. *Tabula. RS algoritma F1 vērtības pie dažādiem apmācīšanas iterāciju skaitiem*

|  |  |  |  |
| --- | --- | --- | --- |
|  | RS, F1 vērtība | | |
| N.p.k. | 10 iterācijas | 50 iterācijas | 100 iterācijas |
| 1 | 0.6382 | 0.6311 | 0.6319 |
| 2 | 0.6371 | 0.6287 | 0.6287 |
| 3 | 0.6491 | 0.6321 | 0.6362 |
| 4 | 0.6346 | 0.6428 | 0.6317 |
| 5 | 0.6324 | 0.6413 | 0.6371 |
| 6 | 0.6495 | 0.6382 | 0.6350 |
| 7 | 0.6199 | 0.6260 | 0.6399 |
| 8 | 0.6334 | 0.6226 | 0.6376 |
| 9 | 0.6117 | 0.6328 | 0.6393 |
| 10 | 0.6371 | 0.6263 | 0.6370 |

Grafiski tie izskātas sekojoši:



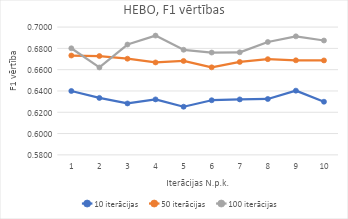
1. *Attēls. RS algoritma F1 vērtības pie dažādiem apmācīšanas iterāciju skaitiem (Autora veidotais attēls)*

Talāk sekoja tāds pašs simulāciju skaits, taču šoreiz ar HEBO algoritmu. To rezultāti ir apkopoti 2. tabulā.

1. *Tabula. HEBO algoritma F1 vērtības pie dažādiem apmācīšanas iterāciju skaitiem (Autora veidotais attēls)*

|  |  |  |  |
| --- | --- | --- | --- |
|  | HEBO, F1 vērtība | | |
| N.p.k. | 10 iterācijas | 50 iterācijas | 100 iterācijas |
| 1 | 0.6400 | 0.6733 | 0.6801 |
| 2 | 0.6335 | 0.6728 | 0.6621 |
| 3 | 0.6283 | 0.6703 | 0.6836 |
| 4 | 0.6321 | 0.6668 | 0.6920 |
| 5 | 0.6252 | 0.6682 | 0.6787 |
| 6 | 0.6313 | 0.6622 | 0.6761 |
| 7 | 0.6321 | 0.6673 | 0.6763 |
| 8 | 0.6325 | 0.6699 | 0.6860 |
| 9 | 0.6403 | 0.6688 | 0.6913 |
| 10 | 0.6299 | 0.6687 | 0.6874 |

Šie dati arī tika attēloti grafiski, ko var redzēt 3. attēlā.



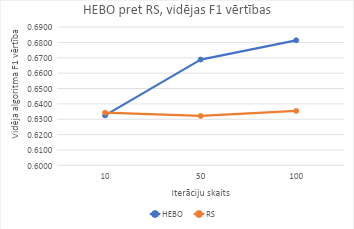
1. *Attēls. HEBO algoritma F1 vērtības pie dažādiem apmācīšanas iterāciju skaitiem (Autora veidotais attēls)*

Visbeidzot pēc visu 60 simulāciju palaišanas tika aprēķinātas 6 vērtības – HEBO un RS algoritmu vidējās F1 vērtības no visiem 10 mēģinājumiem pie 10, 50 un 100 apmācīšanas iterācijām. Tās tika apkopotas 3. tabulā.

*3. Tabula. HEBO un RS algoritmu F1 vērtības salīdzinājums pie dažādiem apmācīšanas iterāciju skaitiem*

|  |  |  |
| --- | --- | --- |
| Iterāciju skaits | HEBO | RS |
| 10 | 0.6325 | 0.6343 |
| 50 | 0.6688 | 0.6322 |
| 100 | 0.6814 | 0.6354 |

Grafiski šie radītaji ir skaidri redzami 4. attēlā.



1. *Attēls. HEBO un RS algoritmu F1 vērtības salīdzinājums pie dažādiem apmācīšanas iterāciju skaitiem (Autora veidotais attēls)*

# Secinājumi

No eksperimenta datiem ir redzams, ka Nejaušās meklēšanas algoritma veiktspēja atrodas nemainīgājā diapazonā neatkarībā no iterāciju skaita. Savukārt HEBO algoritms rada tādus pašus rezultātus ar nelielu iterāciju skaitu, bet tā veiktspēja pieaug, pieaugot iterāciju skaitam, kas ir redzams 4. attēlā. Šī likumsakarība izriet no HEBO algoritma darbošanas principa, kas paredz, ka tas apmācās no iepriekš iegūtiem datiem. Līdz ar to izveidota hipotēze apstiprinas jebkādam iterāciju skaitam, sakot ar kādu vērtību intervalā starp 10 un 50 apmācīšanas iterācijām.

Darba izpildes laikā tika izpildīti visi uzstādīti mērķi. Tika izpētīta literatūra par neironu tīklu pielitošanu ekoloģijas problēmu risināšanai, par HEBO un RS algoritmu darbošānas principiem un pielietojumu, kā arī tika apskatīta un pielitota Python raytune bibliotēka. Visbeidzot tika iegūti nepieciešāmi rezultāti algoritmu salīdzināšanai, apmācot neironu tīklu ar abiem algoritmiem.

Darba izpildes laikā tika izpildīti visi uzstādītie uzdevumii:

1. Sanāca iepazīties ar raytune bibliotēku, tās implementāciju neironu tīklu apmācīšanā un optimālu hiperparametru konfigurāciju optimizēšanā
2. Veiksmīgi tika izmantota “Covertype” datu kopa, kas ļāva apmācīt RFCE neironu tīklu mežu informācijai analizēšanai
3. Eksperimentā tika vairākkārt salidzinātas HEBO un RS algoritmu precizitātes informācijas apstrādē, kas nodrošināja augstāku rezultātu objektivitāti, kā arī iegūties rezultāti tika atteloti grafiski vieglākai uztverei

# Turpmākie pētījumi

No iegūtiem rezultātiem ir acimredzams, ka, izmantojot dažādus hiperparametru kombināciju optimizācijas algoritmus, var iegūt atšķirīgus modeļa precizitātes rezultātus. Taču pēc apmacīšanas modeļu precizitāte vēl joprojam nav perfekta, kas var novest pie aplamām diagnozēm, izmantojot augstāk aprakstītus modeļus. Līdz ar to jaunu, efektīvāku algoritmu, kas spēs optimizēt hiperparametrus ātri un ļaus sasniegt augstus rezultātus, izstrāde ir ļoti aktuāla mūsdienās.

Pašlaik tiek veikti daudzi pētījumi, lai izstradātu jaunus hiperparametru optimizācijas algoritmus. Tie tiek izmantoti mūsdienīgu problēmu risināšanā, kurā tiek apstradāts liels datu daudzums. Viens no pēdējā laika piemēriem ir Covid-19 rentgenbilžu analīze ar neironu tīklu palīdzību, kas tika apmacīti, izmantojot jaunus hiperparametru optimizācijas algoritmus [9]. Daži no tiem algoritmiem ir NSGA-III [9], NSGA-II [9], R-NSGA-II [9], SMS-EMOA [9], MOEA/D [9], PBT (Population Based Training) [10], PRA (Probability-based Resource Allocation) [10] un CASOH (Combined-Sampling Algorithm to Search the Optimized Hyperparameters) [11]. Šie algoritmi var radīt labākus rezultātus, nekā HEBO, jo tie izmanto alternatīvas metodes, kas var būt ātrākas un efektīvākas.

Spriežot pēc pētījumu skaita, kas izstrāda jaunus algoritmus hiperparametru optimizācijas uzdevuma risināšanai, var droši apgalvot, ka turpmākiem pētījumiem šajā jomā ir liels potenciāls. Jaudīgāki algoritmi var atrast pielietojumu ne tikai ekoloģijas jomā, bet arī citās zinātnēs, kas pieprāsa liela datu daudzuma efektīvu apstradāšanu.

# Izmantotie literatūras avoti un atsauces

[1] R. A. Houghton, “Deforestation,” *Biological and Environmental Hazards, Risks, and Disasters, Second Edition*, pp. 273–275, Jan. 2023, doi: 10.1016/B978-0-12-820509-9.00020-4.

[2] Aloïs Bissuel, “Hyper-parameter optimization algorithms: a short review,” Apr. 2019, Accessed: Jan. 26, 2025. [Online]. Available: https://techblog.criteo.com/hyper-parameter-optimization-algorithms-2fe447525903

[3] A. I. Cowen-Rivers *et al.*, “HEBO Pushing The Limits of Sample-Efficient Hyperparameter Optimisation,” Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.03826

[4] V. Martinez-Viol, E. M. Urbano, M. Delgado-Prieto, and L. Romeral, “Automatic model calibration for coupled HVAC and building dynamics using Modelica and Bayesian optimization,” *Build Environ*, vol. 226, p. 109693, Dec. 2022, doi: 10.1016/J.BUILDENV.2022.109693.

[5] S. Andradóttir, “Chapter 20 An Overview of Simulation Optimization via Random Search,” *Handbooks in Operations Research and Management Science*, vol. 13, no. C, pp. 617–631, Jan. 2006, doi: 10.1016/S0927-0507(06)13020-0.

[6] F. Hosseini, C. Prieto, and C. Álvarez, “Hyperparameter optimization of regional hydrological LSTMs by random search: A case study from Basque Country, Spain,” *J Hydrol (Amst)*, vol. 643, p. 132003, Nov. 2024, doi: 10.1016/J.JHYDROL.2024.132003.

[7] Jock Blackard, “Covertype,” 1998.

[8] Evidently AI Team, “How to interpret a confusion matrix for a machine learning model,” Jan. 2025, Accessed: Jan. 26, 2025. [Online]. Available: https://www.evidentlyai.com/classification-metrics/confusion-matrix

[9] B. Gülmez, “A new multi-objective hyperparameter optimization algorithm for COVID-19 detection from x-ray images,” *Soft comput*, Oct. 2024, doi: 10.1007/s00500-024-09872-z.

[10] W. Li, X. Yin, M. Ye, P. Zhu, J. Li, and Y. Yang, “Efficient hyperparameter optimization with Probability-based Resource Allocating on deep neural networks,” *Neurocomputing*, vol. 599, p. 127907, Sep. 2024, doi: 10.1016/J.NEUCOM.2024.127907.

[11] N. H. Tiep *et al.*, “A New Hyperparameter Tuning Framework for Regression Tasks in Deep Neural Network: Combined-Sampling Algorithm to Search the Optimized Hyperparameters,” *Mathematics*, vol. 12, no. 24, Dec. 2024, doi: 10.3390/math12243892.

## Pielikumi

1. Oriģinālais kods – <https://github.com/NEX19/hyper-parameter-comparison>
2. Autora kods - <https://github.com/gfuranets/ZPDHyperparameters>