

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**STIKLA PAKEŠU KONSTRUKTĪVO PARAMETRU
AUTOMATIZĒTA NOTEIKŠANA**

MAĢISTRA DARBS

Autors: **Mārcis Lukševics**

Stud. apl. Nr.: ml19033

Darba vadītājs: Dr. Dat. Kārlis Freivalds

RĪGA 2021

ANOTĀCIJA

Maģistra darba mērķis ir izstrādāt programmatūru, kas pēc ievades datiem spēj būvinženierim parādīt vairākus iespējamus stikla paketes uzbūves variantus, kas atbilst konstruktīvajām prasībām normatīvos.

Darbā implementēti vadlīnijās aprakstīti vienkāršojumi, lai izveidotu ātri aprēķināmus galīgo elementu modeļus. Uz ~2 miljonu galīgo elementu modeļu aprēķinu bāzes ir trenēti neironu tīklu “surogāta modeļi”. Iegūta >98% precizitāte, izmantojot Google piedāvātu automatizētu mašīnmācīšanās risinājumu.

Izveidotie neironu tīklu modeļi sasniedz ~160-kārtīgu ātruma palielinājumu salīdzinot ar galīgo elementu aprēķiniem, vienu pilnu stikla paketes izvērtējumu paātrinot līdz ~300 milisekundēm. Šāds ātrums atļauj lietot optimizācijas algoritmus – ir pielāgota esoša ģenētiskās optimizācijas bibliotēka “GeneticSharp”, kas izmanto mašīnmācīšanās modeļus, lai atrastu vairākus ekonomiskus, normatīviem atbilstošus stikla paketes uzbūves variantus.

Atslēgvārdi: stikla paketes, būvinženierija, mašīnmācīšanās, automatizācija

ABSTRACT

Title of master's thesis: Automated selection of structural parameters for insulated glass units.

This master's thesis aims to develop software that can find several code-compliant options of insulated glass unit build-ups using the input data given by civil engineer.

The simplifications described in the guidelines are implemented to create quickly computable finite element models. Neural network "surrogate models" have been trained using the results data from ~2 million finite element model calculations. Over 98% accuracy was achieved using an automated machine learning solution provided by Google.

The trained neural network models achieve a ~160-times increase in speed compared to the finite element calculations, accelerating one full evaluation of the glass build-up to ~300 milliseconds. This speed enables the use of optimization algorithms. An existing genetic optimization library "GeneticSharp" has been adapted to use machine learning models to find several economical and code-compliant options of insulated glass unit build-up.

Keywords: insulated glass units, civil engineering, machine learning, automation

AUTOREFERĀTS

Darba novitāte. Apkopojot zināmus būvinženierijas un datorzinātņu paņēmienus, ir radīts jauns risinājums ar skaidru praktisko pienesumu inženieru darbā.

Šis ir pirmais pētījums, kurā mašīnmācīšanās metodes lietotas stikla pakešu konstruktīvo prasību pārbaudēm un tā rezultātā vienas stikla paketes aprēķina laiku var palielināt vismaz par ~160 reizēm, salīdzinot ar ekvivalentu galīgo elementu aprēķinu. Šis ir tieši šī pētījuma pienesums.

Ja apskatam kopējo paātrinājumu gan no vienkāršotu aprēķina modeļu ieviešanas, gan mašīnmācīšanās lietojuma – tad kopumā var runāt pat par ~5000 reižu paātrinājumu – no ~25 minūtēm līdz ~300 milisekundēm. Modeļu vienkāršojumu gadījumā gan nevar teikt, ka šis darbs būtu atklājis ko jaunu. Bet praktiskais rezultāts, implementējot normatīvos aprakstītos vienkāršojumus ar “lielu-deformāciju” galīgo elementu aprēķinu, sasniedz labāku precizitātes un ātrdarbības kombināciju, kā esošās komerciālās aprēķina programmas.

Iegūtais ātrdarbības paātrinājums ļauj izmantot optimizācijas algoritmus – līdz ar to, šis ir arī pirmais pētījums, kurā ir izveidots risinājums stikla pakešu uzbūves optimizācijai.

Literatūras izpēte. Kopā izmantotās literatūras sarakstā ir 70 avoti, no tiem 26 ir zinātniskas publikācijas. Vēl 12 ir būvnormatīvi vai programmatūras lietošanas vadlīnijas – uzticami avoti. Literatūras izpētei tika veikta divos līmeņos – sākumā tika apskatīts zinātnisko publikāciju datu bāzēs pieejamie materiāli par tēmu, lai iegūtu vispārīgu priekšstatu. Pēc tam – darba pētījuma laikā, tika meklēta visa veida padziļināta informācija par to, kā strādā aprēķinu vienkāršojumi, mašīnmācīšanās risinājumi vai optimizācijas metodes. Es, kā darba autors uzskatu, ka tika veikta pietiekama izpēte, lai aptveru iepriekš paveikto apskatītajās jomās.

Izklāsta pamatīgums. Šis darbs pēc savas būtības virzās uz mērķi, kas vispārīgi būtu apzīmējams, kā “izveidot prototipu”. Līdz ar to, gandrīz katru no šī darba nodaļām varētu izvērst par atsevišķu pētījumu. Izklāsta (un arī izpētes) “dziļums” tika vērtēts kritiski, uzdodot jautājumu “Vai šis palīdz sasniegt darba mērķi ierobežotajā pētījumam dotajā laikā?”. Īpaši var atzīmēt mašīnmācīšanās daļu – tā kā automatizētie risinājumi izmanto gandrīz pilnu spektru ar algoritmiem un metodēm, tad katras šīs metodes darbību padziļināti izprast un pamatot ir praktiski neiespējami.

Attiecībā uz programmatūru – tā kā programmatūra šobrīd ir pieejama tikai iekšējai lietošanai A/S UPB, tad viss darbs veidots tā, lai tam būtu skaidrs pienesums arī bez pilna programmas koda publicēšanas. Publicētas ir aktuālākās - mašīnmācīšanās un optimizācijas implementācijas koda daļas. Kā darba autors, uzskatu, ka šis pētījums sniedz pietiekamu

daudzumu informācijas, lai turpmākie pētījumi varētu paveikt ko līdzīgu vai labāku, balstoties uz šajā darbā aprakstīto un secināto. Nekas no darbā aprakstītā nav uzskatāms par komercnoslēpumu.

Paveiktā praktiskā darba apjoms. Darbība pie šī pētījuma aizsākās 2020. gada rudenī, līdz 2021. gada janvāra beigām tika veikta literatūras izpēte un izveidots pirmais aprēķina programmas prototips, kurā tika tikai implementēti būvinženierijas aprēķinu modeļu vienkāršojumi – tas tika aprakstīts kursa darbā. No februāra līdz aprīļa sākumam tika “krāta” datu kopa un paralēli testēti dažādi automatizētās mašīnmācīšanās risinājumi. Turpinājumā tika veikts pētījuma apkopojums un optimizācijas implementācija. Kopā, ~300 stundas ir ieguldītas tikai darba pētījuma rezultātā radušās programmatūras izstrādei. Precīza laika uzskate tieši šī pētījuma veikšanai un aprakstīšanai netika veikta.

Rezultātu aprobācija. Kā minēts, pētījums ir pirmais savā jomā. Līdz ar to, nevaram konkrēti runāt par ātrdarbības salīdzinājumu ar citiem, līdzīgiem pētījumiem. Saistībā ar precizitātes vērtējumu – implementētie aprēķinu modeļu vienkāršojumi kopš februāra ir tikuši pārbaudīti, paralēli veicot aprēķinus ar šo programmatūru un komerciālo programmatūru “SJ Mepla”. Saistībā ar mašīnmācīšanās un optimizācijas lietojumu – šīs daļas darba nodošanas brīdī ir pārbaudītas šī darba ietvaros veiktajos salīdzinājumos (skat. nodaļas 6.7 un 7.4) un praktisks pielietojums inženieru vidū tām sekos.

Darba noformējuma kvalitāte. Darbs ir rakstīts programmā Microsoft Word. Darba veidošanā priekšrocība ir dota dinamiskai sasaistei starp dokumenta daļām, lietojot programmā iebūvētās atsauču iespējas. Tādēļ, piemēram – tabulu virsrakstos nav atsevišķās rindās atdalīta numerācija un virsraksts. Līdzīgi – vārds “Tabula” atsaucēs atsevišķās vietās nav pareizā locījumā, lai saglabātu lietot atsauci, nevis manuāli ierakstītu numuru. Pareizu terminu ievērošanai tika izmantota interneta vietne termini.gov.lv, pareizrakstība tika pārbaudīta, darbu pārlasot un izmantojot Microsoft Word pareizrakstības pārbaudes rīku.

Plagiāta iespējamība. Es, kā šī darba autors, vērtēju plagiātisma uzrādīšanu automātiskās pārbaudēs, kā nelielu. Darbā plaši izmantotas atsauces un nav lietoti gari teksta aizguvumi. Lielākais risks iespējams 7 nodaļā, kurā lietots viens no nedaudzajiem materiāliem latviešu valodā par ģenētiskajiem algoritmiem – ne vienmēr ir skaidrs, kuras no definīcijām var uzskatīt par vispārīgām un kuras ir materiāla autora radītas.

SATURS

APZĪMĒJUMU SARAKSTS	9
IEVADS	10
1. VISPĀRĒJS APRAKSTS NO BŪVINŽENIERIJAS SKATPUNKTA.....	13
2. LITERATŪRAS APSKATS	15
2.1 Moderno tehnoloģiju lietojums stikla aprēķinos.....	15
2.2 Pētījumi par konstruktīvu elementu parametru izvēli lietojot mašīnmācīšanās metodes	17
2.2.1 No konstruktīvo elementu skatpunkta	17
2.2.2 No Mašīnmācīšanās pielietojuma skatpunkta.....	18
2.2.3 4-posmu process	20
2.3 Mašīnmācīšanās modeļu uzbūves noteikšana	21
2.4 Parametru optimizatori.....	22
2.4.1 Parametru optimizatoru tipi un īpašības	22
2.4.2 Optimizācijas metodes būvinženierijas problēmu risināšanā	23
2.5 GEM modeļu vienkāršošana	24
2.5.1 Pieejas aprakstītas būvnormatīvos/standartos	25
2.5.2 Pieejas populārākajās specializētajās stiklu aprēķinu programmās.....	26
2.6 Literatūras apskata apkopojums.....	28
3. DARBĀ IZSTRĀDĀTĀ PROCESA APRAKSTS.....	29
3.1 Procesa plūsma.....	29
3.2 Risinājuma sfēra.....	31
4. GEM MODEĻA APRĒĶINU VIENKĀRŠOŠANA	33
4.1 Stikla pakešu specifiskās īpašības.....	33
4.2 GEM Modeļu vienkāršošana.....	34
4.2.1 Izvēlētie vienkāršojumi šajā pētījumā	34
4.2.2 Izvēlēto vienkāršojumu efekts uz rezultātiem	35
4.3 GEM modeļa parametru izvēle	37
4.3.1 Aprēķinu tips	37

4.3.2	Galīgo elementu tips.....	37
4.3.3	Galīgo elementu izmērs.....	38
5.	GEM MODEĻU ĢENERĒŠANA UN REZULTĀTU DATU KOPAS IZVEIDE.....	39
5.1	Aprēķinu veikšana un datu krāšana.....	39
5.1.1	Vispārīgs procesa apraksts.....	39
5.1.2	Ieejas datu ģenerēšana un rezultātu saglabāšana.....	39
5.1.3	Palīg-programma GEM aprēķinu veikšanai.....	41
5.2	Iegūto datu kopas analīze.....	41
5.2.1	Datu skaits.....	41
5.2.2	Parametru korelācija.....	42
5.2.3	Datu sadalījums.....	44
6.	MAŠĪNMĀCĪŠANĀS MODEĻU TRENĒŠANA UN LIETOŠANA.....	46
6.1	Kritēriji modeļu izvērtēšanai.....	46
6.2	Modeļu precizitātes izvērtēšana no būvinženierijas skatpunkta.....	46
6.3	Mašīnmācīšanās ietvara izvēle.....	48
6.3.1	Teorētisks iespēju salīdzinājums.....	48
6.3.2	Precizitātes un ātrdarbības salīdzinājums, izvēlētais satvars.....	52
6.4	Modeļa parametru izvēle.....	56
6.4.1	Zaudējuma funkcijas izvēle.....	56
6.4.2	Biežāk lietoto parametru apgabalu ievērtēšana.....	57
6.4.3	Trenēšanas ilguma ievērtēšana.....	60
6.5	Modelis maksimālo spriegumu noteikšanai.....	61
6.5.1	Modeļa uzbūve.....	61
6.5.2	Modeļa precizitāte.....	63
6.6	Modelis maksimālo deformāciju noteikšanai.....	65
6.6.1	Modeļa uzbūve.....	65
6.6.2	Modeļa precizitāte.....	67
6.7	Ātrdarbības salīdzinājums ar GEM aprēķiniem.....	68
6.7.1	Nepieciešamais laiks vienam aprēķinam un prognozei.....	69
6.7.2	Nepieciešamais laiks vienas stikla paketes pilnam novērtējumam.....	70
7.	STIKLA PAKETES PARAMETRU OPTIMIZĀCIJA.....	73

7.1 Vispārīgi.....	73
7.2 Ģenētiskā optimizācija	74
7.2.1 Vispārīgi	74
7.2.2 Izvēlētās opcijas dažādu optimizācijas algoritma daļu implementācijā	76
7.3 Optimizācijas bibliotēkas izvēle un implementācija.....	81
7.4 Optimizācijas darbības izvērtējums	82
8. IZVEIDOTĀS LIETOJUMPROGRAMMAS APRAKSTS	86
9. REZULTĀTI	89
SECINĀJUMI	91
IZMANTOTĀS LITERATŪRAS SARAKSTS	93
PIELIKUMI.....	100
1. Pielikums. Programmas pirmkods GEM aprēķinu ievades datu ģenerēšanai.....	101
2. Pielikums. Programmas pirmkods palīg-programmai GEM aprēķinu veikšanai	102
3. Pielikums. Pirmkods mašīnmācīšanās un optimizācijas algoritmu implementācijai	103

APZĪMĒJUMU SARAKSTS

- API – lietojumprogrammu saskarne (*Application Programming Interface*)
- CEN – Eiropas standartizācijas komiteja
- E – elastības modulis
- GEM – Galīgo elementu metode
- GPU – grafiskais procesors (*Graphical Processing Unit*)
- HTTP – hiperteksta transporta protokols
- MAE – vidējā absolūtā kļūda (*Mean Absolute Error*)
- MAPE – vidējā relatīvā kļūda, izteikta procentos (*Mean Average Percentage Error*)
- PCA – principiālo komponentu analīze (*Principal Component Analysis*)
- PIP – programmēšanas valodas Python tipiskais bibliotēku pārvaldītājs
- RAM – operatīvā atmiņa (*Random-Access Memory*)
- RMSE – vidējā kvadrātiskā kļūda (*Root Mean Squared Error*)
- RMSLE – vidējā kvadrātiskā logaritmiskā kļūda (*Root Mean Squared Logarithmic Error*)
- SSD – cietvielu disks (*Solid-State Drive*)
- SVM – atbalsta vektoru mašīna (*Support Vector Machine*), mašīnmācīšanās metode
- ν – puasona koeficients
- vCPU – virtuālais centrālais procesors
- WPF – satvars vizuālās saskarnes definēšanai Microsoft Windows vidē (*Windows Presentation Foundation*)

IEVADS

Kā automatizēti atrast optimālu stikla paketes biezumu? Stikla paketes (*insulated glass units*) tiek lietotas logos, durvīs un stiklotajās fasādēs. Tas sastāv no 2-3 slāņiem stikla un 1-2 gāzes šķirkārtām. Komplicētākā daļa konkrēta stikla paketes uzbūves izvēlē ir konstruktīvo prasību izvērtēšana. Stikla paketes slāņu spriegumiem un izliecēm ir jāiekļaujas robežās, kas aprakstītas kādā no industrijā atzītiem standartiem (piem., DIN 18008 vai EN 16612).

Atkarībā no stikla paketes uzbūves, konstruktīvajā aprēķinā ir jāievērtē līdz 35 dažādi parametri, kas saistās ar paketes ģeometriju, slāņu biezumiem un īpašībām, kā arī pieliktajām slodzēm. Tā kā stikla paketes slāņiem ir salīdzinoši mazs biezums, tad precīzu spriegumu noteikšanai ir jālieto nelineārs lielu-deformāciju aprēķins. Gāzes šķirkārtas tilpuma maiņām labāk padosies plānāki slikti, bet ārējām slodzēm labāk pretosies biežāki stikla slāņi. Šie aspekti nosaka procesa sarežģītību un to, ka pārbaudes šobrīd laikietilpīgs process, kas balstās uz inženiera pieredzi un prasmēm.

Literatūras apskats norāda, ka stikla pakešu aprēķinu automatizācija ir maz pētīts temats. Pētījumā [1] 2020. gada rudenī tika norādīts, ka nestspējas prasību izvērtēšana, lietojot mākslīgo intelektu, ir temats, kurā nav veiksmīgu pētījumu. Savukārt pētījums [2], kurā 2017. gadā tika pētīts mašīnmācīšanās lietojums vien-slāņu stikla biezuma izvēlei, norādīja stikla pakešu uzbūves izvēles automatizēšanu kā tālāku pētījumu virzienu.

Šajā darbā pētīti un īstenoti gan datorzinātņu, gan būvinženierijas sfēras risinājumi, lai stikla paketes uzbūves izvēli veiktu kā sistematizētu procesu, pēc iespējas mazākā laikā. Stikla paketei pieliktās slodzes un paketes izmēri tiek uzskatīti par “konstantēm” parametru optimizācijā, jo tās ir atkarīgas no konkrētā projekta prasībām un ir zināmas pirms stikla pakešu aprēķina.

Darba mērķis

Maģistra darba mērķis ir izstrādāt programmatūru, kas pēc ievades datiem spēj būvinženierim parādīt vairākus iespējamus stikla paketes uzbūves variantus, kas atbilst konstruktīvajām prasībām normatīvos.

Balstoties uz šiem variantiem, inženieris var pieņemt lēmumu par kāda no tiem lietošanu projektētajā konstrukcijā.

Piedāvātie risinājumi projektēšanas procesa uzlabošanai

Šajā darbā piedāvāts projektēšanas procesu paātrināt, kombinējot vienkāršojumus, kas izpētīti būvinženierijas industrijā ar aktuālajām iespējām datorzinātņu laukā.

1. (būvinženierija) Galīgo elementu modeļa aprēķinu sarežģītības vienkāršošana, izmantojot būvniecības normatīvos atļautos vienkāršojumus.
2. (būvinženierija/datorzinātnes) Aprēķina modeļu automatizēta ģenerēšana un rezultātu iegūšana.
3. (datorzinātnes) Iepriekš izskaitļoto aprēķinu datu apstrāde, lai prognozētu rezultātu pie iepriekš neizskaitļotiem ievades datiem. Šeit lietotas mašīnmācīšanās metodes, galvenais mērķis ir aizstāt relatīvi ilgu galīgo elementu metodes aprēķinu ar ātru atbildi no iepriekš trenēta mašīnmācīšanās modeļa.
4. (datorzinātnes) Stikla paketes parametru optimizācija, kas spēj sistemātiski meklēt vairākas potenciāli optimālas stikla paketes uzbūves, optimizēšanas laikā saņemot datus no 3. punktā izveidotā modeļa.

Darba struktūra un uzdevumi:

Šis darbs sastāv no ievada, 7 nodaļām, rezultātiem, secinājumiem un izmantotās literatūras saraksta.

Pirmajā nodaļā dots problēmas apraksts no būvinženierijas skatpunkta un vispārīgi aprakstīti piedāvātie risinājumi, lai sasniegtu darba mērķi. Pēc tam apskatīsim literatūru - veidus, kādos fizisko paketi var aprakstīt ar galīgo elementu modeļiem, apskatīt citu pētījumus saistībā ar konstrukciju, īpaši stikla, optimālu parametru atrašanu, apskatīt aktuālās pieejas mašīnmācīšanās un parametru optimizācijas nozarēs;

Trešajā nodaļā ir vispārīgi aprakstīts šajā darbā izstrādātais process un tā sfēra.

Ceturtajā nodaļā izvēlēsimies tādu fiziskās konstrukcijas aprakstu ar galīgo elementu modeli, kas ļauj vienu modeļa aprēķinu veikt pēc iespējas īsākā laikā. Piektā nodaļa apraksta, kā automatizēti tiek ģenerēti GEM modeļi, un to rezultātu dati apkopoti, lai tiktu izmantoti kā ievades dati mašīnmācīšanās modeļu trenēšanai.

Sestā nodaļa apraksta mašīnmācīšanās modeļu trenēšanu un darbināšanu. Šo mašīnmācīšanās modeļu uzdevums ir aizstāt galīgo elementu aprēķinus, un tādējādi būtiski paātrināt procesu.

Septītā nodaļa nonāk līdz šī darba mērķa īstenošanai – izvēlamies un implementējam optimizācijas algoritmu, kas spēj atrast optimālu stikla paketes uzbūvi, balstoties uz inženiera dotajiem ierobežojumiem un ievades datiem.

Darba rezultāti

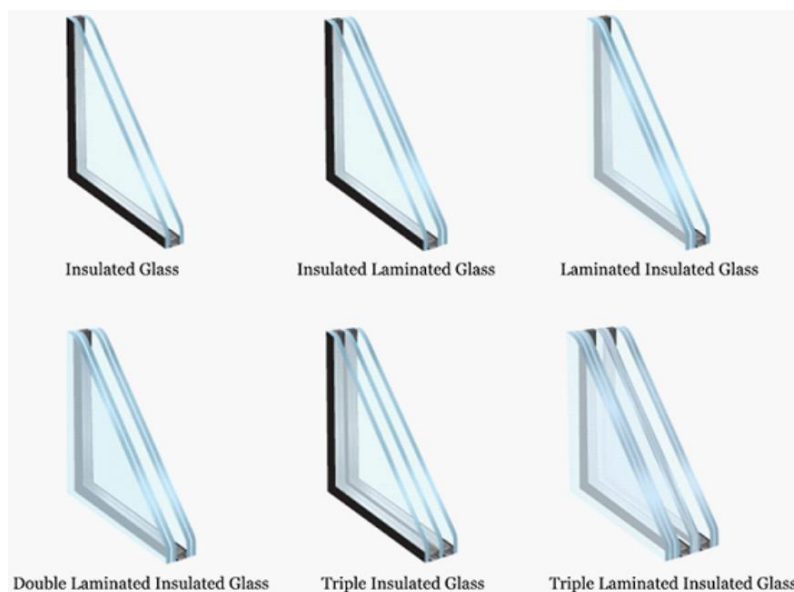
Šī darba ietvaros ir izstrādāta lietojumprogramma, kas, izmantojot mašīnmācīšanās modeļus, spēj aprēķināt vienas paketes uzbūves maksimālās deformācijas un spriegumus ar attiecīgi 99.7% un 98.3% precizitāti. Prognozes vienai parametru kopai aizņem mazāk par 0.4 sekundēm, kas ir vairāk nekā 120 reizes ātrāk (atkarībā no parametriem), nekā līdzvērtīgs galīgo elementu aprēķins.

Lietojumprogrammā ir implementēta ģenētiskās optimizācijas bibliotēka "GeneticSharp", kas atļauj minimizēt paketes uzbūvē lietoto kopējo stikla biezumu.

Šo lietojumprogrammu šobrīd izmēģina A/S UPB meitas-kompānijas SIA Aile Grupa inženieri un pirmās atsauksmes ir pozitīvas.

1. VISPĀRĒJS APRAKSTS NO BŪVINŽENIERIJAS SKATPUNKTA

Stikla paketes tipiski ir taisnstūrveida elementi no vairākiem slāņiem, kas nostiprināti rāmjos visās četrās malās. Stikla paketēm ir jānodrošina droša ēkas ekspluatācija – tām jākalpo vēja aizturēšanai no ārpuses un kā norobežojošai konstrukcijai no iekšpusēs.



Att. 1.1 Stikla pakešu dažādas uzbūves – mainīgs šķirkārtu skaits un laminēta stikla lietojums [3]

Konkrēta stiklu paketes risinājuma izvēli projektā ietekmē:

- Nestspējas prasības;
- Siltumizolācijas prasības;
- Akustiskās prasības;
- Gaismas caurlaidības prasības.

Komplīcētākā daļa konkrēta stikla paketes “uzbūves” izvēlē ir konstruktīvās nestspējas prasību izvērtēšana.

Parametri, kuri jāievērtē projektēšanā, ir:

- Stikla paketes izmēri;
- Slāņu skaits – 1 vai 2 gāzes šķirkārtas.
- Katra no 2-3 stikla slāņu īpašībām.
 - Iespējams lietot parastu/rūdītu stiklu;
 - Iespējams lietot laminētu stiklu no 2 (vai vairāk) kārtām. Laminētā stiklā var kopā laminēt parastu un rūdītu stiklu, dažādus stiklu biezumus.
 - Laminēta stikla uzbūve. Plēves īpašības, kas izmantota, lai laminētu kopā stiklus.

- Pieliktās vēja, apkopes un iemītņieku slodzes.

Aprēķinu matemātiskās sarežģītības dēļ, šobrīd aprēķins vienam konkrētam gadījumam aizņem aptuveni 25¹ minūtes (tikai datora aprēķinu laiks), plus laiks, kas pavadīts ievades modeļa sagatavošanā.

Praktiski tas nozīmē, ka stiklu paketes uzbūves izvēle notiek balsoties uz iepriekšējo pieredzi, attiecīgi “izmēģinot” versijas, kas izvērtētas kā ticamākās. Tas nozīmē, ka pēc vairāku darba dienu darba, ir atrastas dažas stiklu uzbūves versijas, kas izpilda normatīvu prasības. Bet nav objektīva veida, kā noskaidrot, ka nav ticis nepamanīts kāds vēl optimālāks risinājums.

Šī darba rezultāts ir programmatūra, kas dod iespēju sistematizēt stikla paketes uzbūves izvēli – paātrinot viena aprēķina ātrumu un pielietojot optimizācijas algoritmu automātiskai stikla paketes uzbūves izvēlei.

¹ Laiki variē no 10-50 minūtēm atkarībā no darbstacijas, stikla paketes, apskatītajām slodžu kombinācijām. Aprēķins stikla paketei ar izmēru 3500x1150mm, 2-slāņu pakete, abi slāņi laminēti, 132 slodžu kombinācijas, darbstacija ar Intel Core i9-9880H, 32GB atmiņu, SSD disku, aprēķins programmā SJ Mepla 5.0 lietojot 64-bitu risinātāju un visus 16 procesora kodolus aizņem 24 minūtes.

2. LITERATŪRAS APSKATS

2.1 Moderno tehnoloģiju lietojums stikla aprēķinos

Līdzīgi kā pārskatā [1], arī šī darba sākumā ir jānorāda, ka būvniecības industrija ir viena no vismazāk digitalizētajām [4].

2020. gada oktobrī tika publicēts pētījums - pārskats par mākslīgā intelekta lietojumu stiklotajās konstrukcijās [1]. Attēlā Att. 2.1 izcelta joma, kurai veltīts šis maģistra darbs. Stikla konstrukciju stiprības pārbaudes. Kā redzams, līdz šim šajā jomā nav bijuši panākumi, tā ir aktuāls pētījumu subjekts.

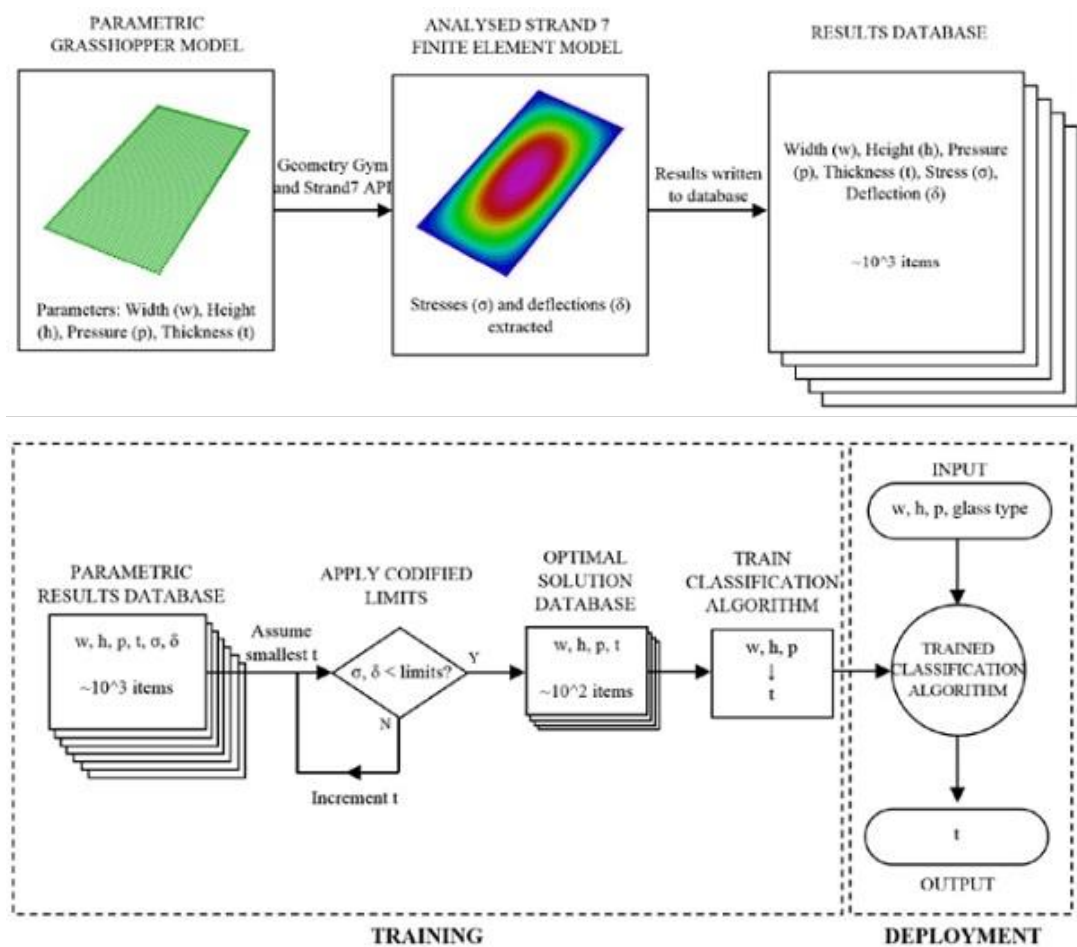
Overview and Summary Table of Examples of this Paper						
Presented Example / Future Potential	Content	Data Format	AI Algorithms	Data Intensity ¹	AI application success ²	Conclusion
Engineering User-Centered Adaptive Façades	Design and of a user-centered yet economic and sustainable façade with adaptive components governed by AI	structured; unstructured	ML, DL	+++	0	conceptual paradigm presented; lots of open research questions;
Data-driven Glass Product Development	Development of new glass products by AI algorithms	structured	ML, DL	++	✓	possible and conducted
Inspection and Control of Glass Products	DL based inspection of glass products demonstrated using CNN for Pummel-Classification of Laminated Glass	structured; unstructured	DL	+++	✓	possible and conducted; model capabilities to be increased
Semantic Segmentation of Glass Edge based on Images	Supervised DL for cut edge detection based on input images and masking images	unstructured	DL	++	✓	possible and conducted
Prediction of Glass Edge Strength based on Process Parameters	Supervised ML for prediction of edge strength of machine-cut glass based on process parameters of the cutting machine	structured	DL	++	✓	possible and conducted
Hyperelasticity Model Calibration for TSSA	Supervised ML for stochastic calibration of a Nelder-Helmholtz Potential to model hyperelasticity of TSSA silicon	structured	ML	+	✓	possible and conducted
Prediction of Glass Breakage Patterns (BREAK)	ML Fracture Pattern Prediction of thermally pre-stressed glasses based on pre-stress magnitude	structured; unstructured	ML	+	✓	possible and conducted; model capabilities to be increased
Designed by AI	AI proposes several designs taking into account user-specific preferences and technical boundaries	structured; unstructured	ML, DL	+++	0	conceptual paradigm presented; lots of open research questions;
Structural Verification	AI proposes verifies technical feasibility of a proposed design in early planning stages	structured; unstructured	ML, DL	+++	0	conceptual paradigm presented; lots of open research questions;
Data-driven Modeling of Materials within Glass-Structures	Use AI together with experimental data as surrogate for constitutive model of complex material	structured	ML, DL	+++	✓	possible and partially conducted

Structural Verification	AI proposes verifies technical feasibility of a proposed design in early planning stages	structured; unstructured
-------------------------	--	--------------------------

ML, DL	+++	0	conceptual paradigm presented; lots of open research questions;
--------	-----	---	---

Att. 2.1 Mākslīgā intelekta lietojums dažādos konstruktīvā stikla inženierijas aspektos [1].

Pētījums [1] norāda, ka “surogātu modeļi” varētu tikt izmantoti, lai ātri parādītu konstrukcijas piemērotību bez tieša aprēķina. 2017. gadā ir publicēts pētījums [2], kurš pēc savas būtības ir ļoti līdzīgs šim darbam – taču krietni mazākā apjomā un tādēļ ar ierobežotu praktisko pielietojamību. Pētījumā veiktie soļi redzami attēlā Att. 2.2. Šī pētījuma autori veica monolīta stikla aprēķinus ar dažādiem stikla izmēriem un vienmērīgi izklaidētu slodzi.



Att. 2.2 Pētījumā [2] veiktie soļi. Inženierijas aprēķins = datu vākšana redzama augšējā daļā. Neironu tīkla trenēšana – apakšējā daļā.

Kā potenciālos tālākos pētījumu virzienus [2] autori norāda:

- Dažādu slodžu ievērtēšanu;
- Algoritmus biezuma optimizēšanai divslāņu un trīs-slāņu stikla paketēm;
- “nekonkrētu” dizaina parametru integrēšanu ieteikumu sistēmā – tādu kā vizuālās kvalitātes vai pēc-sabrukuma uzvedību.

Pirmie no šiem diviem tālākajiem pētījumu virzieniem apskatīti šajā maģistra darbā.

Apskatot stikla inženieriju plašākā kontekstā, jāpiemin pētījums [5], kurā tiek izmantoti gan fizisku eksperimentu, gan simulāciju dati, lai trenētu neironu tīklu, kurš “iesaka” stikla ķīmisko sastāvu. Kā veiksmīgs izmantojuma piemērs tiek minēts Corning® Gorilla® stikls, ko esam raduši redzēt izmantotu mobilajās ierīcēs.

2.2 Pētījumi par konstruktīvu elementu parametru izvēli lietojot mašīnmācīšanās metodes

Šajā darbā relatīvi ilgais galīgu elementu aprēķinus tiks aizstāts ar ātrāku mašīnmācīšanās modeli, kas apmācīts, lietojot iepriekš izrēķinātu galīgo elementu aprēķinu modeļu rezultātus. Tādēļ ir būtiski apskatīt, kā līdzīgas problēmas jau ir atrisinājuši citi pētnieki.

2.2.1 No konstruktīvo elementu skatpunkta

Šajā nodaļā veikts literatūras apskats par to, kādu konstruktīvo elementu aprēķinos jau tikušas lietotas mašīnmācīšanās metodes.

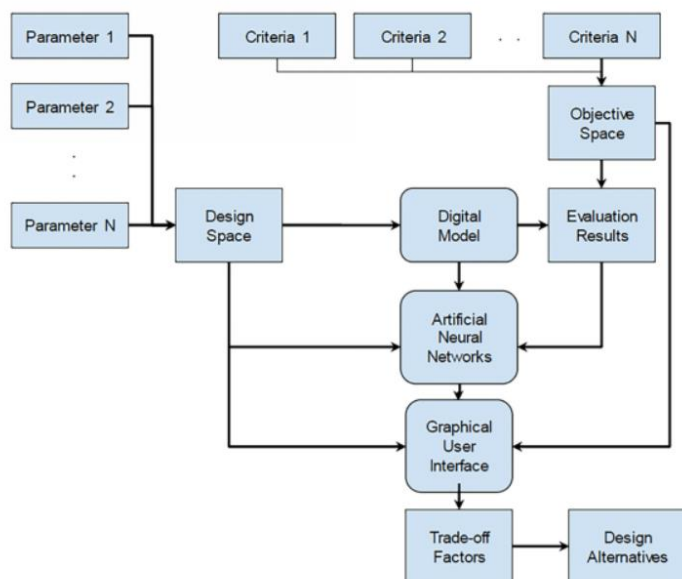
Pētījums [1] norāda uz mākslīga intelekta lietošanu, lai aprēķinātu tērauda cauruļveida profilu ļodzes pretestību un lieces momenta aprēķināšanu plātnēs. Pētījums [6] norāda uz “atbalsta vektora mašīnu” (*support vector machines* jeb SVM) un neironu tīklu lietojumu, lai noteiktu šķērsspēka kapacitāti sijām ar lielu laidumu, kā arī dzelzsbetona siju aprēķinu bez šķērsspēka stiegrojuma, lietojot evolucionāro polinomiālo regresiju (*evolutionary polynomial regression*).

Plašs literatūras apskats atrodams pētījumā [7]. Viena no norādēm ir uz [8], kurā autori veica svāra optimizāciju kompozītam panelim ar stinguma ribām, aizstājot GEM modeli ar neironu tīklu, kurš trenēts ar iepriekšējiem aprēķinu datiem. Līdzīga pieeja ir veikta [9], kurā neironu tīkls lietots kopā ar ģenētisko algoritmu (*genetic algorithm*) – tādējādi optimizācijas algoritms var iegūt atbildes minējumiem būtiski ātrākā laikā, nekā katru reizi veicot GEM modeļa aprēķinu. GEM modeļa “aizstāšana” ar neironu tīklu veikta arī [10], kur noteiktas vienkāršas kopnes izlieces katrā no tās mezgliem.

Vienkāršas kopnes kā piemēru izmanto arī pētījums [11]. Šis pētījums interesants ar to, ka tas piedāvā veselu satvaru, kā integrēt neironu tīklus konstruktīvo elementu projektēšanā. Skatīt attēlu Att. 2.3. Plašāk par vairākos pētījumos izmantoto darbu plūsmu skatīt nodaļā 2.2.3.

Pētījums [6] sava procesa un apjoma ziņā, šķiet, vistuvāk salīdzināms ar šī maģistra darba ietvaros veikto. Aprēķins apskata vien-virziena dzelzsbetona riboto pārseguma plātņu optimizāciju, ņemot vērā arī to provizorisko cenu un CO₂ izmešu daudzumu. Atšķirībā no citiem (un šī) pētījumiem, dati šajā gadījumā ir ģenerēti ar analītisku formulu palīdzību, nevis GEM aprēķiniem. Līdzīgi šajā pētījumā plānotajam, tiek lietoti 8 ievades parametri, kas apraksta gan elementa dimensijas, gan slodzi. Kopumā veikti 1 029 105 “eksperimenti”, ir salīdzināms ar šī darba ietvaros praktiski iespējamo. Pēc tam tiek trenēti vairāki neironu tīkli

un tiek vairāk-objektīvu optimizācija, lai iegūtu Pareto-optimālu risinājumu atbilstoši dotajam dažādu objektīvu nozīmīgumam.



Att. 2.3 Vispārīga ietvara ieteikums konstruktīvo variantu vērtēšanai lietojot neironu tīklus [11]

Parameter	Minimum	Maximum	Increases
Inter-axis (i)	50 cm	100 cm	5 cm
Rib cast-in-place (n)	10 cm	26 cm	2 cm
Compression-Layer (c)	4 cm	8 cm	1 cm
High hollow block (h)	10 cm	40 cm	3 cm
Span one-way slab	350 cm	650 cm	50 cm
Live Loads	0 kN/m ²	4 kN/m ²	2 kN/m ²
Load of Pavement	0 kN/m ²	2 kN/m ²	1 kN/m ²
Load of Partitions	0 kN/m ²	2 kN/m ²	1 kN/m ²

Att. 2.4 Ieejas parametri un to robežas pētījumā par vien-virziena riboto dzelzsbetona plātņu optimizāciju [6]

2.2.2 No Mašīnmācīšanās pielietojuma skatpunkta

Šajā nodaļā iepriekš veiktie pētījumi skatīti no mašīnmācīšanās skatpunkta – kāda metode pielietota, cik daudz ieejas datu lietoti, kādi bijuši metodes parametri, kāda precizitāte sasniegta.

Bisagni un Lanzi pētījumā par kompozītajiem paneļiem [8] pētnieki lietoja atsevišķus neironu tīklus (kopā 10 gab.) katram no parametriem – ar vienu slēpto slāni (*hidden layer*), ar 3-24 neironiem katrā, *tanh()* aktivizācijas funkcijām. Ievades dati tika normalizēti. Ieejas datu apjoms pētījumā aprakstīts neskaidri – zināms, ka tika kopā veikti 70+55 = 125 galīgo elementu

aprēķini, bet nav skaidrs, kādi tieši dati no šiem aprēķiniem tika lietoti ieejas datiem neironu tīklā. Trenētais neironu tīkls tika lietots, lai veiktu optimizāciju, lietojot ģenētisko optimizācijas algoritmu. Salīdzinot neironu tīklu dotās “atbildes” ar galīgo elementu aprēķiniem, tika iegūta 93% precizitāte sliktākajā gadījumā, bet 75% no treniņu un testa seta ieejas datiem precizitāte bija labāka par 96%.

Ruijter pētījumā par paneli ar atvērumiem [9], arī tika lietoti vairāki neironu tīkli (katram izvades parametram viens). Katram neironu tīklam bija viens slēptais slānis, ar 3 ieejas parametriem un vienu izejas parametru. Ievades dati saturēja kopā 1300+ datu kopas. Iegūtā precizitāti bija 80% sliktākajam no izvades parametriem. Arī šajā pētījumā tika lietots ģenētiskais optimizācijas algoritms. Jāņem arī vērā, ka šis pētījums veikts 2003. gadā.

Pētījumā par monolītu stiklu optimālās struktūras izvēli [2] tika kopā tika veikti ~1000 aprēķini. 50% no datiem lietoti tīkla trenēšanai, 25% validācijai un 25% testēšanai. Tika lietots neironu tīkls ar vienu slēpto slāni, apskatīta dažāda neironu skaita lietošana, un gala rezultātiem izmantoti 128 neironi. Šajā gadījumā tika lietoti 4 ieejas parametri (3 skaitliski, viens kategorisks) un tika iegūti līdzīgi rezultāti gan tīklu veidojot kā “regresijas”, gan “klasifikācijas”. Neironu tīkls attiecīgi prognozēja nepieciešamo stikla biezumu ar 80% precizitāti.

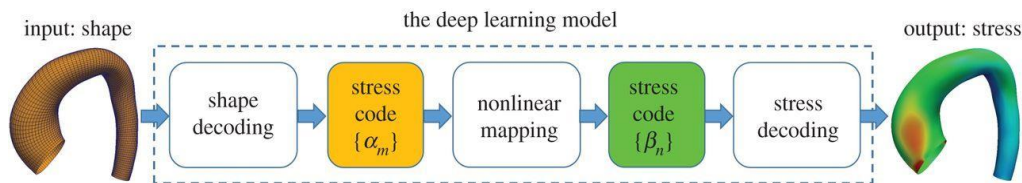
Pētījumā par vien-virziena dzelzsbetona plātnēm [6], tika lietoti 8 ieejas parametri, katram no 5 izejas parametriem tika trenēts savs neironu tīkls ar 3 slēptajiem slāņiem. 70% no datiem tika lietoti trenēšanai, 30% testiem. Tika lietots 0.5 atmešanas (*dropout*) koeficients visiem slāņiem, ReLU aktivācijas funkcija. Skatīt attēlu Att. 2.5. Iegūtā precizitāte bija lielāka par 99.5%.

Model DNN	n_{h1}	n_{h2}	n_{h3}	Dropout1	Dropout2	Dropout3	Learning Rate	Momentum
Cost (€/m ²)	256	1024	16	0.5	0.5	0.5	0.005	0.726322
Energy (MJ/m ²)	1024	1024	16	0.5	0.5	0.5	0.005	0.698062
CO ₂ (kgCO ₂ /m ²)	512	1024	16	0.5	0.5	0.5	0.005	0.567881
Rigidity	512	1024	16	0.5	0.5	0.5	0.005	0.567881
Deflection (cm)	512	512	8	0.5	0.5	0.5	0.005	0.768566

Att. 2.5 Neironu tīklu parametri lietoti pētījumā par vienvirziena dzelzsbetona plātnēm

Šajā nodaļā vērts pieminēt arī pētījumu [12], kurā neironu tīkls lietots, lai aizstātu galīgo elementu metodi spriegumu aprēķinā cilvēka aortai. Lai gan šis ir medicīnas jomas pētījums –

GEM spriegumu aprēķins pēc savas būtības neatšķiras no tā, kas tiek lietots būvkonstrukciju aprēķinos. Pētījuma rezultātā 30 minūšu ilgu galīgo elementu aprēķinu varēja aizstāt ar 1 sekundi ilgu mašīnmācīšanās rezultātu un iegūtā precizitāte bija lielāka par 1%. Kā ievades dati pētījumā, tika lietotas 729 aortu ģeometrijas, katras modelis saturēja 5000 punktus.



Att. 2.6 Mašīnmācīšanās process stresa noteikšanai cilvēka aortas modeļos [12]

Salīdzinoši ar citiem pētījumiem šajā nodaļā, šis izcēlās ar sarežģītāku mašīnmācīšanās pielietojumu. Sākumā rezultāti 5000 punktos tikai samazināti uz mazāku skaitu ievades parametru, lietojot PCA. Pēc tam tika lietots tīkls ar 2 slēptajiem slāņiem un 128 neironiem katrā tīklā. Tika lietota “Softplus” aktivācijas funkcija. Pēc tam tika lietota “zema ranga aproksimācija (*Low rank approximation*), lai no izejas neironiem iegūtu spriegumus katrā punktā. Modeļa trenēšana notika lietojot “Adamax” algoritmu. Monte-Carlo kross-validācija tika veikta ar 90%/10% treniņa/testa datu sadalījumu. Pētījuma rezultātā 30 minūšu ilgu galīgo elementu aprēķinu varēja aizstāt ar 1 sekundi ilgu mašīnmācīšanās rezultātu, iegūtā precizitāte bija lielāka par 99%.

2.2.3 4-posmu process

Faktiski visos apskatītajos pētījumos skaidri parādās četru posmu process, kas ir par pamatu arī šim maģistra darbam.

1. Izvēlēties/identificēt ievades datus, kas ir kā parametri galīgo elementu modelim;
2. Lietojot galīgo elementu metodi, veikt pietiekamu skaitu mēģinājumu. Izvēlēties rezultātu datus, kurus lietot kā mašīnmācīšanās procesa izejas datus.
3. Trenēt mašīnmācīšanās modeli. Vairākos no apskatītajiem avotiem [6] [7] [8] [9] tika izvēlēts veidot vairākus neironu tīklus – ar vienādiem ieejas datiem, bet katram savu izejas parametru.
4. Pēc-apstrādāt mašīnmācīšanās modeļa dotos rezultātus. Avotos [8] [9] [11] tika lietots ģenētiskais optimizācijas algoritms.

2.3 Mašīnmācīšanās modeļu uzbūves noteikšana

Šī darba ietvaros ir uzdevums iegūt precīzu “surogāta” mašīnmācīšanās modeli, kas var aizstāt galīgo elementu aprēķinus. Mašīnmācīšanās modeļu algoritmu un uzbūves noteikšana ir būtiska daļa, lai varētu sasniegt augstu precizitāti.

Ar būvinženieriju saistītajos pētījumos ir lietotas dažādas pieejas mašīnmācīšanās modeļu izvēlē un uzbūves noteikšanā. Pētījumā par vien-virziena dzelzsbetona plātnēm [6] neironu tīklu uzbūve ir atrasta, lietojot nejaušu meklēšanu (*random search*). Bisagni un Lanzi [8] pētījumā tika lietota sarežģītāka pieeja – trenēti atsevišķi vairāki tīkli un pēc tam kombinēti ansambļi. Pētījumā par monolītu stiklu optimālās struktūras izvēli [2] tika izvēlēti un salīdzināti 25 dažādi “platuma” neironu tīkli. Lai gan pētījumā [8] ir lietota sistemātiska pieeja ansambļa būvēšanā, neviens no pētījumiem neveic optimālās modeļa arhitektūras meklēšanu.

Pēdējā laikā, popularitāti sāk gūt automatizēti mašīnmācīšanās satvari. Atkarībā no katra satvara funkcionalitātes, tie apskata šādus aspektus [13]: datu sagatavošana, datu pēcapstrāde (*feature engineering*), modeļa uzdevuma klasifikācija/regresija izvēle, modeļu/algoritmu izvēle, hiperparametru izvēle, modeļa vērtēšanas metriku izvēle, rezultātu izvērtēšana, problēmu detektēšana.

Pētījumā par modernākajām automatizētajām mašīnmācīšanās metodēm praktiskam pielietojumam [14] norādīts piemērs, kurā automatizēts risinājums 30 minūtēs atrada labāku risinājumu, nekā 3 speciālistu komanda no atpazīstamās kompānijas “PriceWaterhouse Coopers” vairākās nedēļās. Ja iepriekš minētais ir viens specifisks piemērs, tad Google [15] un Amazon [16] risinājumi ir uzrādījuši labus rezultātus vairākās “Kaggle” sacensībās [17].

ASV banku sektora milža CapitalOne pētījumā [18] 2019. gada rudenī tika salīdzinātas dažādu satvaru iespējas – tajā kopā tika salīdzināti 14 satvari. Šī darba tapšanas laikā 2021. gada pavasarī, interneta vietne “AI multiple” [19] parādīja kopā 24 dažādus automātiskās mašīnmācīšanās satvarus. Pētījumā [16], kurā aprakstīts “AutoGluon”, tas salīdzināts ar 5 citiem populārākajiem automatizētajiem risinājumiem:

Popular AutoML frameworks for classification and regression with tabular data. We indicate whether a table of RAW data with missing and non-numerical values can be handled automatically without manual preprocessing and specification of feature types.

AUTOML FRAMEWORK	OPEN	RAW	NEURAL NETWORK	CASH STRATEGY	MODEL ENSEMBLING
AUTO-WEKA	✓	×	SIGMOID MLP	BAYESIAN OPTIMIZATION	BAG, BOOST, STACK, VOTE
AUTO-SKLEARN	✓	×	NONE	BAYESOPT + META-LEARN	ENSEMBLE SELECTION
TPOT	✓	×	NONE	GENETIC PROGRAMMING	STACKING
H2O	✓	✓	MLP + ADADELTA	RANDOM SEARCH	STACKING + BAGGING
GCP-TABLES	×	✓	ADANET (??)	ADANET (??)	BOOSTING (??)
AUTOGLUON	✓	✓	EMBED CATEGORICAL + SKIP-CONNECTION	FIXED DEFAULTS (SET ADAPTIVELY)	MULTI-LAYER STACKING + REPEATED BAGGING

Att. 2.7 Vairāku Automatizēto mašīnmācīšanās satvaru salīdzinājums [16]

No Att. 2.7 redzamajiem populārajiem satvariem, vecākais ir TPOT [20], kas parādījies 2015. gada rudenī. 2016. gadā Auto-Weka sāka atbalstīt šajā maģistra darbā aktuālās regresijas problēmas [21]. 2016. gada rudenī parādījās arī Auto-Sklearn [22] un H2O [23]. Google AutoML Tables “Beta” izlaide notika 2019. gada aprīlī [24]. Kā pēdējais ar savu satvaru ir pievienojies “Amazon” ar risinājumu “AutoGluon” 2020. gada martā [16]. Visi šie satvari ir relatīvi jauni.

Interesanti, ka no Att. 2.7 redzamajiem satvariem, katram ir kāda nozīmīga atšķirība. Nemainīgs ir tas, ka visos satvaros kā gala modelis tiek no vairākiem iepriekš satrenētiem atsevišķiem modeļiem. Būtiski atšķiras divi no šiem algoritmiem – “GCP-Tables” jeb “Google Cloud AutoML Tables” [25] un “AutoGluon”. Google risinājums izceļas ar to, ka tas integrēts mākoņpakalpojumā ar grafisko saskarni un skaidru ieteikto trenēšanas jaudu un attiecīgi skaidrām izmaksām. Tas ir arī vienīgais no risinājumiem, kura pirmkods nav atvērts. “AutoGluon” risinājums atšķiras ar to, ka tas necenšas optimizēt hiperparametrus (kā to dara visi citi satvari), bet fokusējās uz fiksētu parametru modeļu kombināciju vairākos līmeņos.

2.4 Parametru optimizatori

Maģistra darba ietvaros, optimālas stikla paketes uzbūves (t.i., stikla biezumu un šķirkārtu) meklēšanai paredzēts lietot parametru optimizatoru. Tādēļ literatūras apskatā ir apskatīti vispārīgo optimizācijas metožu tipi, kā arī algoritmi, kas iepriekš veiksmīgi lietoti būvinženierijas problēmu optimizācijā.

2.4.1 Parametru optimizatoru tipi un īpašības

Parametru optimizācija jeb argumentu vērtību atrašana, pie kuras funkcijas vērtība sasniedz maksimumu, minimumu vai konkrētu vērtību, ir ļoti plaša matemātikas nozare. Vispārīgi, optimizācijas metodes var sadalīt 3 tipos [26] :

- Algoritmi, kas atrod optimālos parametrus ar galīgu skaitu soļu. Tāds, piemēram, ir “Simplex” algoritms, kas popularitāti ieguvis, jo ir iebūvēts Microsoft Excel optimizācijas rīkā “Solver”. Šādi algoritmi ir piemēroti lineāru problēmu risināšanā, t.i., kad starp parametriem un mērķa funkciju ir lineāra sakarība.
- Iteratīvie algoritmi, kas pamazām konverģē uz optimālo risinājumu – bet nekad neapstājas. Tādēļ tiem ir nepieciešams konverģences kritērijs. Pie šīs grupas pieder tādas populāras metodes kā Ņūtona metode un Gradientu metode.

- Heiristiskie algoritmi. Šie algoritmi, teorētiski, var arī risinājumu neatrast (vai arī atrast neoptimālu risinājumu), tomēr praksē tie var būt noderīgi. Pie šīs sadaļas pieder evolucionējošie algoritmi.

2.4.2 Optimizācijas metodes būvinženierijas problēmu risināšanā

Literatūras izpētē tika secināts, ka daudzos pētījumos ir ticis izmantots kāds no ģenētiskajiem algoritmiem, kas pieder pie evolucionējošo algoritmu kopas.

Avotā [27] optimizētas augstceltņu konstrukcijas lietojot viena-mērķa ģenētisko algoritmu. Pētījums [28] apraksta tērauda telpiska rāmja konstrukciju optimizāciju, lietojot ģenētisko algoritmu vairāku-mērķu optimizāciju. Pētījums [29] apraksta kopņu optimizāciju, lietojot vairāku-mērķu “The Third Evolution Step Differential Evolution (GDE3)” algoritmu, kas pieder pie ģenētisko algoritmu grupas. Pētījums [30] lieto optimizēšanu ar “daļiņu spieta” metodi (*Particle Swarm Optimization*) evolucionējošo algoritmu, lai paredzētu dzelzsbetona konstrukciju progresīvo sabrukumu.

Ģenētiskos algoritmus lieto arī iepriekšējās nodaļās aprakstītie pētījumi [8] un [9].

Praktiskam pielietojumam konstrukciju projektēšanā, ir izstrādāta arī programmatūra [31], kas ļauj pielietot 8 dažādus heiristiskos (tai skaitā ģenētiskos) optimizācijas algoritmus kopā ar komerciālo konstrukciju aprēķinu programmu “SCIA”.

Avots [32] argumentē, kādēļ konstrukciju optimizācijā heiristiskie (t.sk. evolucionējošie) algoritmi ir daudz piemērotāki par iteratīvajiem, atsaucoties uz [33] galvenajiem secinājumiem:

- Tie ir piemēroti, lai efektīvi izpētītu visu parametru telpu un meklētu globālu optimumu, dažkārt atklājot negaidītus risinājumus;
- Tiem nav nepieciešama gradientu aprēķināšana, un tie var kā ievades datus lietot kategoriskus (diskrētus) mainīgos;
- Mērķa funkcijai nav jābūt konveksai;
- Ātra optimizācijas metodes konfigurēšana.
- Avots [32] papildina sarakstu ar piezīmi, ka konstrukciju optimizācijas optimālajam risinājumam, ir grūti izvēlēties labu “sākuma punktu”. Iteratīvo algoritmu lietošana var radīt konverģences problēmas.
- Avots [34] papildina sarakstu ar piezīmi, ka evolucionējošie algoritmi ļauj lietotājam viegli pārskatīt optimizācijas procesu – t.i. redzēt starprezultātus. Kā arī ir iespējas dialogam starp algoritmu un lietotāju (šis punkts gan nav sīkāk paskaidrots).

Mans, kā maģistra darba autora, viedoklis ir, ka atsevišķs iemesls ģenētisko algoritmu plašajai lietošanai ir šīs optimizācijas metodes lietojums vienā no populārākajiem parametriskās modelēšanas rīkiem “Grasshopper” [35]. Šis rīks no “ģeometrijas modelēšanas” ir attīstījies par plaša spektra vizuālās programmēšanas rīku būvniecības nozarē (t.sk. tam ir GEM aprēķinu spraudnis “Karamba” [36]). “Grasshopper” ir iebūvēts parametru optimizators “Galapagos”, kas lieto ģenētisko algoritmu. Šī pieeja ir tālāk attīstīta spraudnī “Octopus” [37], kas spēj veikt vairāku-mērķu optimizāciju, lietojot evolucionējošo algoritmu un Pareto principu.

Populārā rīka “Galapagos” autors savā rakstā [34] skaidri norāda uz tā diviem galvenajiem trūkumiem:

- Evolucionējošie algoritmi ir ļoti lēni.
- Evolucionējošie algoritmi negarantē, ka optimums tiks atrasts.

Ja otrā no šīm īpašībām ir vispārīga heuristisko algoritmu īpašība, tad pirmajai ir vērts pievērst uzmanību. Rīks “Galapagos” sākotnēji ir ticis izstrādāts tieši ģeometrisku formu optimizācijai, kur katra iterācija aizņem ļoti maz laika, jo ir matemātiski vienkārša.

Ja vienas iterācijas/aprēķina veikšana ir darbietilpīga (piem. GEM aprēķina veikšana), tad būtiski ir lietot algoritmu, kas var tuvināties optimumam ar relatīvi mazāku iterāciju skaitu. Šim piemērotā [38] ir Beijesa optimizācijas (“Bayesian optimization”) metode, kas aktīvi tiek lietota mašīnmācīšanās modeļu hiperparametru optimizācijā (piemēram, satvars HyperOpt [39]).

Autors atrada vienu pētījumu, kurā salīdzināta Beijesa optimizācijas un Evolucionējošās optimizācijas metodes priekš lietojuma konstrukciju optimizācijā. 2020. gada septembrī publicētajā pētījumā [40] zinātnieki veica Dzelzsbetona sijas vairāku-mērķu optimizāciju un salīdzināja Beijesa optimizācijas algoritmu ar “Non-dominated Sorting Genetic Algorithm II (NSGA-II)” un secināja, ka Beijesa algoritms strādā būtiski labāk, ja vērtē, kādu uzlabojumu var iegūt pēc katras iterācijas izpildes.

Apskatot Beijesa optimizācijas darbības principus rakstā [41], šī darba autoram šķita būtiska ideja, ka Beijesa optimizators darbības laikā būvē “surogāta modeļi”. Tas pēc būtības ir līdzīgs process, kā nodaļā 2.2.2 aprakstītajos pētījumos, kur GEM modeļa rezultātu aizstāšanai tika būvēti/trenēti neironu tīklu “surogāta modeļi”.

2.5 GEM modeļu vienkāršošana

Pirmais šī pētījuma priekšlikums esošā inženierijas procesa uzlabošanā (skat. darba ievadu) ir galīgo elementu modeļa aprēķinu sarežģītības vienkāršošana, izmantojot būvniecības normatīvos un vadlīnijās aprakstītos vienkāršojumus.

Šis ir aspekts, kas būtiski ietekmē visu tālāko pētījuma gaitu datorzinātņu laukā – iespējas iegūt optimālu rezultātu ir atkarīgas no viena potenciālā risinājuma aprēķina ilguma. Pie esošās situācijas, kad viena aprēķina ilgums ir aptuveni 25 minūtes (skat. 14. lpp), ir nepraktiski lietot optimizāciju, kā arī varētu savākt tikai nelielu datu kopu mašīnmācīšanās modeļu apmācībai. Lai aizstātu GEM aprēķinus ar apmācītu modeli, ir nepieciešami iepriekš ģenerēti dati, kādi šobrīd nav pieejami. Jo ātrāk šādu datu kopu iespējams ģenerēt, jo lielākas iespējas uz šī darba mērķa sasniegšanu.

Turpmāk šajā nodaļā apskatīti esošie risinājumi, kā ar analītisku formulu lietošanu, ir iespējams būtiski paātrināt GEM modeļa rezultātu iegūšanu.

2.5.1 Pieejas aprakstītas būvnormatīvos/standartos

Šī darba ietvaros tika iepazīts dažādu Eiropā un ASV lietotu stikla aprēķinu standartu salīdzinājums [42] kā arī jaunākais Eiropā CEN aprobētais standarts EN 16612 [43]. Man, kā šī darba autoram, ir iepriekšēja pieredze stiklu aprēķinos pēc normatīva DIN 18808 [44].

Jāpiebilst, ka šī darba ietvaros ir būtiski veikt aprēķinus, kas ir atbilstoši DIN 18808 vai EN 16612, jo tie ir normatīvi, kas aktuāli galvenajos eksporta tirgos Latvijas kompānijām.

- Lietotie standarti pieļauj **gaisa šķirkārtas vienkāršošanu**, lietojot pieeju aprakstītu 2006. gadā izdotajā Feldmeier aprakstā “Klimabelastung und Lastverteilung bei Mehrscheiben-Isolierglas” [45]. Šis praktiski nozīmē, ka gaisa šķirkārtas netiek modelētas, bet tiek atsevišķi aprēķinātas stikla plaknes, tām pieliekot papildu slodzes, kas simulē gaisa šķirkārtu darbību.
- Standartos ir dažādas pieejas **laminēta stikla slāņu kopdarbības vienkāršošanai** GEM modeļos. Faktiski visi standarti atļauj laminētu stiklu aizstāt ar monolītu stiklu, kuram lietots ekvivalentais biezums.
 - DIN 18008 [44] prasa ignorēt kopdarbību statistiskajos aprēķinos, bet lietot koeficientu 1.1 nestspējas uzlabošanai. Līdz ar to ekvivalento stikla biezumu var vienkārši aprēķināt pēc būvmehānikas formulām;
 - EN 16612 [43] atļauj inženierim izvēlēties piemērotu metodi. Kā arī kā informatīvu piedāvā savu metodi. Saistībā ar lamināta kopdarbības ievērtēšanu, ir veikts pētījums, ka EN 16612 implementētā metodes rezultāti nav uzticami [46];
 - Jauna un daudzsološa metode lamināta kopdarbības ievērtēšanai ir “Enhanced Effective Thickness” metode, aprakstīta [47]. Šī metode piedāvā

aprēķināt ekvivalento biezumu, ievērtējot stikla paketes malu attiecību un slodzes veidu;

2.5.2 Pieejas populārākajās specializētajās stiklu aprēķinu programmās

Galīgo elementu metode (GEM) ir visplašāk lietotā metode inženierijas problēmu risināšanai [48]. Tā tiek arī plaši izmantota konstrukcija, tai skaitā stikla pakešu aprēķinos. Tomēr GEM modeļa pieņēmumi (līdz ar to – tā sarežģītība) dažādās esošajās specializētajās komerciālajās programmās stiklu pakešu aprēķinam, atšķiras.

Apskatītas programmas MEPLA SJ, Glastik un Dlubal RFEM modulis RF-Glass. Šīs ir komerciālās programmas, kuras licencētas darba autora darba vietai A/S UPB. Tomēr jāmin, ka šīs arī ir vienīgās šī pētījuma veikšanas laikā zināmās specializētās aprēķina programmas un risinājumu dažādības trūkums daļēji ir šī darba tēmas aktualitātes iemesls.

Nevienai no apskatītajām trim programmām nav API piekļuves, kas ļautu tās tiešā veidā izmantot šī darba ietvaros.

Programma MEPLA SJ 4.0

Šī programma ir šobrīd iespējām bagātākā stikla pakešu aprēķinu programma. Tā ir programma, kas ieguvusi industrijas uzticību, jo jau vismaz 14 gadus darbojās specifiski ar stikla aprēķiniem. Programmatūra piedāvā veikt lineāru un nelineāru “lielu-deformāciju” (*large deformations*) aprēķinu [49].

Programma neveic normatīvos piedāvātos vienkāršojumus.

- Šķirkārtu gāzes darbība tiek modelēta ar 3D galīgajiem elementiem.
- Laminēti stikli tiek modelēti kā vairāk-slāņu 2D galīgie elementi (t.i., kā kompozītas plātnes) [49]. Šāda pieeja nav jauna no galīgo elementu modelēšanas viedokļa, bet nav implementēta citās stiklu aprēķinu programmās un ļauj būtiski samazināt aprēķina laiku salīdzinot ar vairāk-slāņu 3D telpisko elementu lietošanu.
- Viens “tipisks” aprēķins ar 132 kombinācijām (aptuvenais skaits modelim ar tipiskām slodzēm) aizņem 10-50 minūtes, atkarībā no stikla izmēra un galīgo elementu tīkla (*mesh*) izmēra.

Programma Glastik 3.0

Programmas Glastik 3.0 niša ir pretēja MEPLA SJ. Tā ir vienkārša vedņa (*wizard*) tipa programma.

- Tā lieto lineāru statikas aprēķinu [50], līdz ar to tās rezultāti ir konservatīvi (t.i., neprecīzi);
- Gaisa šķirkārta netiek modelēta, tiek lietota Feldmeier [45] līdzīga pieeja vienkāršojumam, tomēr Glastik dokumentācija [50] tiešā veidā neatsaucas uz Feldmeier;
- Laminēta stikla kopdarbības ievērtēšanai, lietotājam tiek lūgts ievadīt koeficientu no 0 līdz 1, kur 1 nozīmē pilnu kopdarbību, bet 0 nozīmē atsevišķas stikla loksnes. Šī pieeja faktiski atļauj tikai pielietot DIN 18008 pieeju.

Programmas lielākais trūkums ir tās darbība kā “melnajai kastei”. Programma spēj dot rezultātu <15 sekundēs. Precizitāte ir “aizstāta” ar ātrdarbību. Tomēr jāņem vērā, ka lineārās metodes atšķirība no “lielu pārvietojumu” var būt 20% un vairāk (skat. nodaļu 4.3.1)

Programmas Dlubal RFEM modulis RF-Glass

Programmas Dlubal RFEM modulis RF-Glass faktiski ir atsevišķa programma-programmā. Dlubal RFEM ir vispārīga galīgo elementu aprēķinu programma būvinženierijas problēmām un spēj modelēt brīvas formas ģeometriju, lietojot 1D, 2D un 3D galīgos elementus. Tai ir arī pieejams API.

RF-Glass lieto kopējā programmā definētās slodzes, bet ģenerē jaunu-neatkarīgu galīgo elementu modeli. Šim modulim API nav pieejams. Šis modulis ir jaunākais no trim apskatītajām programmām.

- Gaisa šķirkārta tiek modelēta kā 3D telpiskie galīgie elementi. Netiek lietoti normatīvos piedāvātie vienkāršojumi.
- Stikla slāņi un lamināta plēves arī modelētas kā 3D telpiskie galīgie elementi. Netiek lietoti normatīvos piedāvātie vienkāršojumi.

Programmai teorētiski vajadzētu nodrošināt visprecīzākos rezultātus. Tomēr pilns kombināciju kopas (132^2 kombinācijas) aprēķins aizņem ap 11³ stundām, līdz ar to programmas racionālai lietošanai būtu būtiski jāsamazina slodžu kombināciju skaits, kas citā veidā “atņem” precizitāti.

² Apskatītas 44 kombinācijas ar vēja, līnijas, temperatūras un punkta slodzi. Apskatītas vasarā un ziemā iespējamās gaisa spiediena maiņas šķirkārtās. 44 kombinācijas ULS apskatītas gan pie pilna lamināta stinguma, gan ignorējot to (DIN18088 prasība), tādējādi rezultējoties 88 ULS kombinācijās. SLS apskatītas tikai kombinācijas bez lamināta plēves stinguma – 44 SLS kombinācijas. Kopā 88+ 44 = 132 kombinācijas.

³ 1150x3500mm divslāņu stikla pakete ar laminētiem stikliem. Tikla izmērs = 50mm.

2.6 Literatūras apskata apkopojums

Stikla konstrukciju stiprības pārbaužu automatizācija ir aktuāls, bet maz pētīts temats [1], kurā ir veikti maza apjoma pētījumi (piemēram, [2]), bet pētījumi nav izvērsti uz pielietojamību stikla paketēm. Lai gan, galīgo elementu aprēķinu darbietilpības dēļ, ir problemātiski ģenerēt datus no pilniem stikla pakešu modeļiem (skat. 2.5.2), tomēr ir vairāki inženierijas vienkāršojumi, kurus var piemērot, lai būtiski vienkāršotu GEM modeļus [45] [47].

Ir bijuši vairāki veiksmīgi mašīnmācīšanās lietojumi galīgo elementu aprēķinu aizstāšanai ar daudz ātrākiem neironu tīkliem [6] [8] [9] [12], no kuriem pētījums [6] šķiet arī provizoriski salīdzināms parametru skaita un kopējas sarežģītības ziņā ar maģistra darba ietvaros paredzēto.

Pēdējo piecu gadu laikā parādījušies un strauji popularitāti guvuši “AutoML” risinājumi, kas automatizēti izvēlas mašīnmācīšanās modeļa arhitektūru, sagatavo datus un veic citas darbības ar mērķi uzlabot modeļa precizitāti. Šādi risinājumi atsevišķos gadījumos var iegūt labāku rezultātu kā cilvēki-profesionāļi [14] un tie iegūst augstas vietas mašīnmācīšanās sacensībās [17]. Kā divi jaunākie no populāriem satvariem jāpiemin 2019. gada “Google AutoML Tables” [15] un 2020. gadā publicētais “Amazon AutoGluon” [16].

Evolucionējošie algoritmi ir līdz šim populārākā parametru optimizācijas metode ar būvinženieriju saistītos pētījumos. Pie nosacījuma, ka GEM modelis tiek aizstāts ar ātri strādājošu apmācītu mašīnmācīšanās modeli, evolucionējošie algoritmi ir piemēroti arī šim pētījumam. Ja GEM modelis netiktu aizstāts, tad Beijesa optimizācijas algoritmi būtu provizoriski piemērotākie.

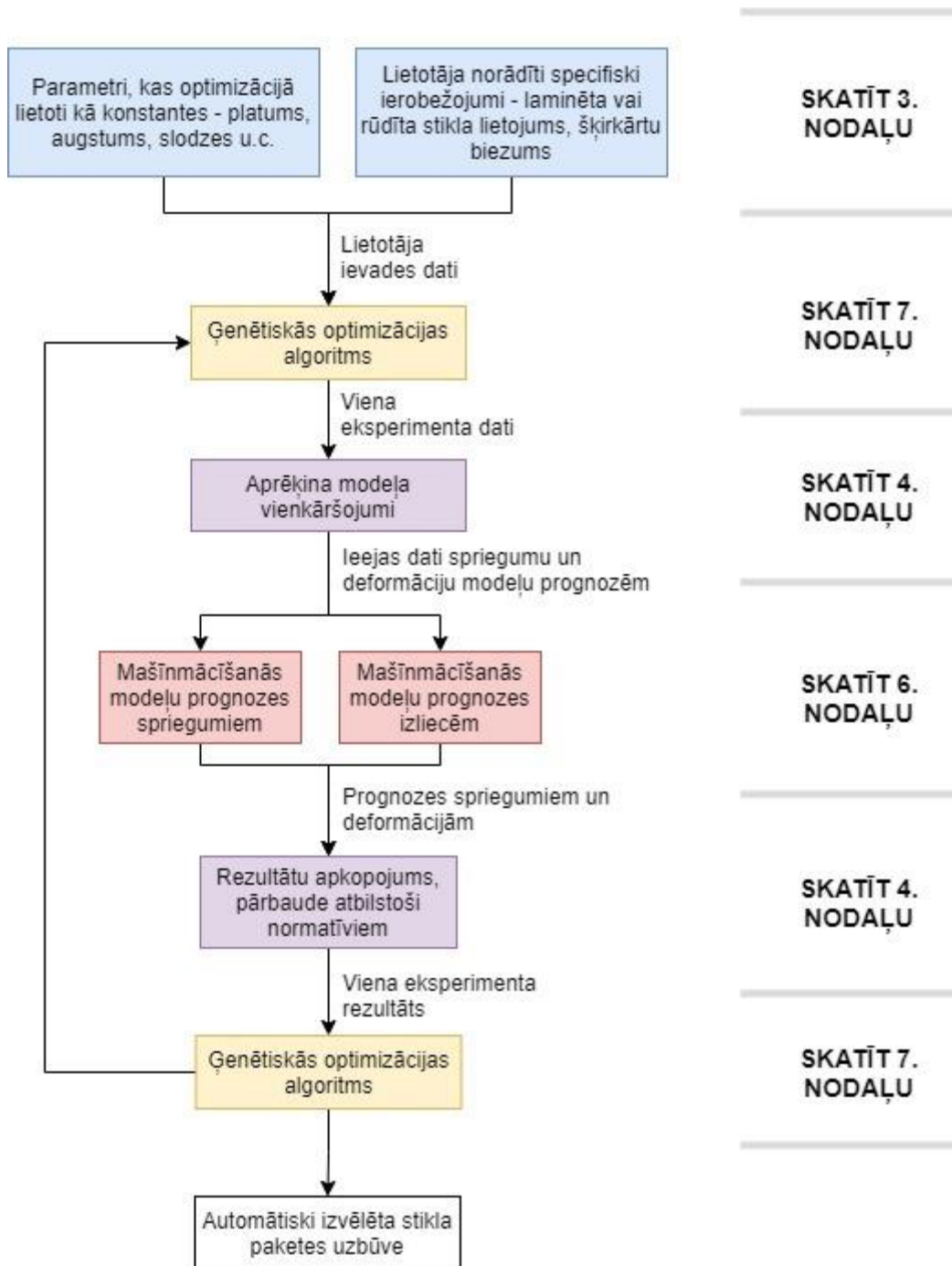
Attiecībā uz GEM modeļu vienkāršošanu, divi Eiropā plašāk izmantotie standarti EN 16612 [43] un DIN 18808 [44] atsaucās uz 2006. gadā izstrādāto metodoloģiju [45]. Laminēta stikla vienkāršošanā daudzsološa ir “Enhanced Effective Thickness” metode [47]. Šobrīd populārākajās specializētajās stiklu aprēķinu programmās nav implementēti šie vienkāršojumu apvienojumā ar lielu-pārvietojumu galīgajiem aprēķiniem.

Vērtējot literatūras apskatu, var secināt, ka neviens no atsevišķiem šī darba posmiem nav uzskatāms par novitāti. Tomēr apkopojot zināmus inženierijas paņēmienus ar lietotājam draudzīgiem mašīnmācīšanās risinājumiem, pētījums ir pirmais stikla pakešu inženierijas jomā, turklāt ar skaidru praktisko pienesumu inženieru darbā.

3. DARBĀ IZSTRĀDĀTĀ PROCESA APRAKSTS

3.1 Procesa plūsma

Att. 3.1 parādīts procesa pārskats, kas nodrošina automātisku stikla paketes uzbūves izvēli. Tālākās darba nodaļas apraksta šī procesa posmus.



Att. 3.1 Process automatizētai stikla paketes uzbūves izvēlei

Apraksts zemāk un Att. 3.1 vispārīgi apraksta nodaļā 8 parādītās lietojumprogrammas darbību.

Procesa sākumā tiek ievadīti izejas dati. Viena daļa no šiem ieejas parametriem ir nepieciešami stikla paketes aprēķinos, bet optimizācijas procesā darbojas kā konstantes. Šī datu daļa būtībā apraksta visu, izņemot stikla paketē lietotos stiklus un šķirkārtu biezumus, kuri tiek izvēlēti šajā procesā – stikla paketes izmērs, lietotās laminēto stiklu plēves, pieliktās slodzes.

Otra daļa no šiem datiem ir ierobežojumi optimizācijai. Tos inženieris var ievadīt, balstoties uz projekta specifiskām prasībām. Pie šiem datiem ietilpst ierobežojumi rūdīta stikla lietošanai, laminēta/monolīta stikla lietošanai, ārējā un iekšējā stikla biezumu attiecība, tikai simetrisku laminētu stiklu lietojums un gaisa šķirkārtas maksimālais un minimālais biezums.

Lietojot šos datus, ģenētiskās optimizācija algoritms izveido mēģinājumu kopu pirmajai “paaudzei”. Tiek lietoti iepriekš definētie ieejas dati un ģenerētas vairākas versijas stiklu paketes uzbūvei. Katrs mēģinājums pēc būtības ir kādas konkrēts stikla paketes uzbūves pārbaude. Optimizācijas algoritma aprakstu skatīt nodaļā 7.

Tad kad zināmi visi dati priekš pārbaudāmās stikla paketes konkrētajā mēģinājumā, notiek tās aprēķinu modeļa vienkāršošana, kas rezultējās 40 – 240 un vairāk vienkāršiem aprēķiniem priekš taisnstūrveida stikla loksnes ar pieliktām slodzēm. Skatīt nodaļu 4.2.1.

Katram no šiem vienkāršajiem aprēķiniem rezultāti tiek iegūti ar mašīnmācīšanās modeļu palīdzību. Šī pētījuma ietvaros, sākumā tika izrēķināts liels skaits aprēķinu, lietojot GEM modeļus (skat. nodaļu 5). Pēc tam, balstoties uz GEM aprēķinu datiem, tika trenēti mašīnmācīšanās modeļi (skat. nodaļu 6).

Kad zināmi visi vienkāršoto aprēķinu rezultāti, tie tiek apkopoti un apskatīts, vai stikla pakete atbilst lietotāja izvēlētajā normatīva (DIN 18008 vai EN 16612) prasībām un lietotāja dotajiem ierobežojumiem stikla tipam u.c. Ja pakete atbilst normatīvam un ierobežojumiem, tad tās biezums tiek iekļauts ģenētiskās optimizācijas “paaudzes” datu kopā.

Optimizācijas mērķis ir izvēlēties stikla paketes uzbūvi ar minimālo kopējo stikla biezumu. Kad veikti visi vienas “paaudzes” mēģinājumu aprēķini un iegūts katra mēģinājuma kopējais stikla biezums, tiek izvēlēti labākie mēģinājumi, tie tiek mutēti un veidota nākamā “paaudze”. Labākais mēģinājums tiek parādīts lietotājam.

Tā kā ģenētiskā optimizācija ir heuristisks process, tad optimizācijas algoritms pats nevar izvērtēt, kad atrasts optimāls risinājums. Lietotājam ir iespēja regulēt optimizācijas apstāšanos vai nu norādot maksimālo optimizācijas darbības laiku, vai arī norādīt paaudžu skaitu, kuru laikā rezultātam stagnējot, optimizācija tiek apturēta.

3.2 Risinājuma sfēra

Šī nodaļa apraksta stikla pakešu un tām pielikto slodžu īpašības, kurām ir piemērojami šajā darbā aprakstītie risinājumi. Sfēras parametri ir izvēlēti atbilstoši darba autora darba vietas A/S UPB ražošanas iespējām un tirgus specifikai. Tas praktiski nozīmē, ka izveidotie risinājumi ir piemēroti stikla paketēm, kuras lieto Skandināvijā, Vācijā un Lielbritānijā.

Vispārīgie risinājuma sfēras parametri:

- Taisnstūrveida stikla paketes ar 2 vai 3 stikla slāņiem;
- Pārbaudes atbilstoši normatīviem DIN18008 un EN16612;
- Monolīti un laminēti stikla slāņi. Laminētam stiklam pieņemts, ka kopā salīmētas 2 monolīta stikla loksnes.

Geometrijas parametri:

- Stikla lokšņu biezumi 4mm, 5mm, 6mm, 8mm, 10mm, 12mm;
- Gaisa šķirkārtu biezumi 8mm, 10mm, 12mm, 14mm, 15mm, 16mm, 18mm;
- Stikla paketes īsākais izmērs 400mm līdz 3000mm;
- Stikla paketes garākais izmērs 400mm līdz 5000mm;

Slodžu parametri:

- Punktveida slodze pieņemta kā pielikta stikla paketes centrā;
- Punktveida slodzes izmērs 100mm;
- Punktveida “raksturīgās” (*characteristic*) slodzes spēks līdz 1.5kN
- Līnijveida “raksturīgās” (*characteristic*) slodzes spēks līdz 1.5kN/m
- Gaisa šķirkārtu biezumi 8mm, 10mm, 12mm, 14mm, 16mm, 18mm;

Izejot no augstāk definētajiem sfēras parametriem, tika izvēlēti parametri un to augšējās/apakšējās robežas priekš GEM aprēķiniem. Uz šo GEM aprēķinu rezultātu bāzes tika veidoti mašīnmācīšanās modeļi (skat nodaļas 4 un 5). Tabula 3.1 parāda datu apgabalu ar kādu ir trenēti modeļi:

Tabula 3.1 Ieejas datu apkopojums – viena eksperimenta parametri

Parametrs	Apakšējā robeža	Augšējā robeža	Skaitlisks/ kategorisks	Piezīmes
Paketes platums	0.4 m	3.0 m	Skaitlisks	
Paketes augstums	0.4 m	5.0 m	Skaitlisks	
Stikla biezums	4 mm	24mm	Skaitlisks	4mm ir plānākais viena-slāņa stikls. 24mm ir ekvivalents laminātam no 2x12mm stikliem ar ļoti stingu līmējošo plēvi.
Izkliedēta slodze	0.0 kN/m ²	25.0 kN/m ²	Skaitlisks	Augšējā robeža izvēlēta, lai ievērtētu lielas klimatiskās slodzes, kas var rasties šaurās vai ļoti stingās stikla paketēs.
Līnijas slodze – vērtība	0.0 kN/m	2.25 kN/m	Skaitlisks	Maksimālā raksturīgā slodze kā 1.5kN/m un maksimālais drošības koeficients 1.5 atbilstoši EN1990 [51]
Līnijas slodze - novietojums	0.1 no paketes augstuma	0.5 no paketes augstuma	Skaitlisks	Ja slodze ir augstāk par 0.5 no augstuma, tad tiek apskatīts “apgriezta” pakete – attiecība 0.6 būs vienāda ar 0.4 utt.
Punkta slodze – vērtība	0 kN	2.25 kN	Skaitlisks	Maksimālā raksturīgā slodze kā 1.5kN/m un maksimālais drošības koeficients 1.5 atbilstoši EN1990 [51]
Vai izkliedētās slodzes virziens ir tāds pats kā punkta un līnijas slodzēm?	Nē (-1)	Jā (1)	Kategorisks	Nav apskatīts gadījums, kurā punkta un līnijas slodzēm ir pretēji virzieni, jo tāds nav iespējams – šīs slodzes vienmēr tiks pieliktas virzienā no ārpuses uz stikla paketes gāzes šķirkārtu.

4. GEM MODEĻA APRĒĶINU VIENKĀRŠOŠANA

Darba ievadā aprakstīti piedāvātie risinājumi projektēšanas procesa uzlabošanai. Šajā nodaļā aprakstīts pirmais no šiem soļiem – GEM aprēķinu vienkāršošana. Attiecīgi – šī nodaļa apraksta būvinženierijā svarīgus aspektus.

4.1 Stikla pakešu specifiskās īpašības

Kā fizisks objekts, stikla pakete sastāv no:

- Vairākām stikla kārtām, tipiski vienas kārtas biezums ir daudz mazāks par pārējām divām dimensijām (platums un augstums). Stikls ir amorfs materiāls, tomēr statikas aprēķinu veikšanai elastības modulis tiek pieņemts nemainīgs $E = 70\,000\text{ MPa}$, $\nu = 0.23$ [43] [44]. Stikls ir elastīgs izotropis materiāls;
- Vienas vai vairākām gaisa šķirkārtām. Tajās tiek iepildīta gāze, kas var būt gaiss, bet biežāk tiek izmantots 90% argona un 10% gaisa maisījums. Aprēķinu daļai būtiski tas, ka šī ir gāze ar atbilstošām termodinamiskajām īpašībām – tā izplešas/saraujas temperatūras ietekmē, kā arī gāzes spiediens uz visām ierobežojošajām robežām ir vienāds.
- Ja tiek lietots laminēts stikls, stikli tiek salīmēti kopā ar plēvi. Plēves īpašības būtiski variē atkarībā no pieliktās slodzes ilguma, kā arī temperatūras [52]. Normatīvs [44] nosaka, ka aprēķini ir jāapskata ar diviem galējiem pieņēmumiem – pilnīgi stingu plēvi un bez plēves (kā divas neatkarīgas loksnes).

No augstāk parādītajām sastāvdaļām izriet galvenie izaicinājumi fiziskā objekta aprakstīšanai ar galīgo elementu metodi

- Tipiski stikla deformācijas ir lielas, salīdzinoši ar tā biezumu – līdz ar to lineārs aprēķins dos neprecīzus rezultātus (lielākas izlieces/spriegumus). Skat. nodaļu 4.3.1.
- Gaisa šķirkārtas darbības ievērtēšana;
- Laminēta stikla mainīgo īpašību ievērtēšana;

Saistībā ar gaisa šķirkārtu un laminētu stiklu, ir dažādas pieejas, kā šiem izaicinājumiem pieiet normatīvi un populārākās specializētās stiklu aprēķina programmas. Skatīt nodaļu 2.5.2.

4.2 GEM Modeļu vienkāršošana

Lai izpildītu šī darba mērķi, ir izveidota sistēma, kas atrod vienu vai vairākas optimālās stikla paketes parametru kopas. Lai veiktu optimizāciju, sistēmai noteikti jāveic daudzu parametru kopu pārbaudes – t.i., vienkāršākajā gadījumā tas nozīmētu daudzu galīgo elementu modeļu izveidošanu un aprēķināšanu. Jo vienkāršāks modelis, jo ātrāk tas var tik aprēķināts, un ātrāk lietotājam var parādīt rezultātu.

4.2.1 Izvēlētie vienkāršojumi šajā pētījumā

Pēc normatīvu un esošās programmatūras apskates (skat. nodaļu 2.5), tika izvēlēti šādi galīgo elementu modeļa vienkāršojumi:

- Gaisa šķirkārtas vienkāršotas atbilstoši Feldmeier [45] pieejai, kas lietota arī DIN 18808 [44] un EN16612 [43]. Gaisa šķirkārtas netiek modelētas – bet aizstātas ar to darbību simulējošām izkliedētajām slodzēm;
- Laminēta stikla efekts vienkāršots atbilstoši “Enhanced Effective Thickness” [47] pieejai. Lamināta slāņi netiek atsevišķi modelēti – bet aizstāti ar ekvivalenta biezuma monolītiem stikliem.

Šo abu vienkāršojumu ieviešana priekš darbā apskatītajām paketēm (skat. nodaļu 3.2) nozīmē, ka galīgo elementu modeļi tiek vienkāršoti līdz vien-slāņa taisnstūrveida loksneim, kas balstītas pa perimetru un kurām pieliktas vienmērīgi izkliedētas, līnijveida un punktveida slodzes. Viens vienkāršotā GEM modeļa aprēķins attiecas uz:

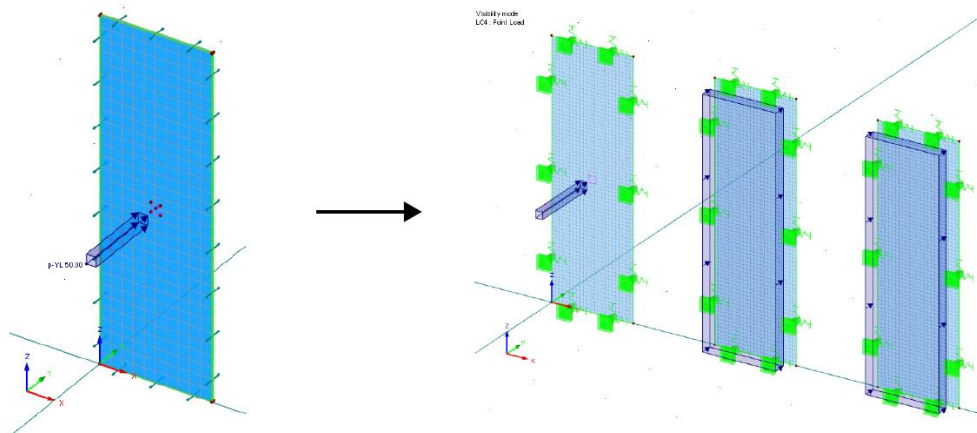
- Vienu stikla paketes stikla slāni (ārējo, iekšējo, vidējo);
- Vienu slodžu kombināciju; Slodžu kombināciju skaitu nosaka inženieris, ievērtējot konkrētā projekta prasības.
- Laminētiem stikliem – vienu laminējošās plēves stinguma pieņēmumu;

Tātad no viena sarežģīta modeļa esam nonākuši līdz vairākiem vienkāršotiem modeļiem. Šo modeļu skaits ir atkarīgs no stikla slāņu skaita un tā, vai kāds no slāņiem ir laminēts. Tāpat kā “sarežģītajam” modelim, arī vienkāršotajiem modeļiem ir jāapskata vairākas slodžu kombinācijas.

Vienkāršākajā gadījumā, kur tiktu apskatītas tikai ~20 slodžu kombinācijas, tikai vēja slodze, 2-slāņu pakete ar monolītiem (nevis laminētiem) stikliem, būtu jāveic 20 x 2=40 aprēķini.

Tomēr praktiski parasti projekta prasības prasa izvērtēt daudz vairāk slodžu kombinācijas, tiek lietotas 3-slāņu paketes ar laminētiem stikliem. Līdz ar to, tipiskā aprēķinā

ar ~40 slodžu kombinācijām, 3 slāņu paketi ar laminēta stikla lietošanu, ir jāveido 3 slāņi x 2 lamināta stingumi = 6 GEM modeļi un katram jāveic 40 aprēķini, kopā 240 aprēķini.



Att. 4.1 Pilns stikla paketes galīgo elementu modelis (pa kreisi) un vienkāršots modelis (pa labi).

4.2.2 Izvēlēto vienkāršojumu efekts uz rezultātiem

Tika veikts salīdzinājums divslāņu stikla paketei starp:

- Mepla SJ programmatūras rezultātiem, kuros starpslānis modelēts ar 3D telpiskajiem galīgajiem elementiem un laminēts stikls modelēts kā “kompozīts” ar sasaistītiem diviem čaulas tipa elementu slāņiem.
- Dlubal RFEM programmatūras rezultātiem (vispārīga modeļa, nevis moduļa RF-Glass), kuros starpslāņa efekti aizstāti ar izkliedētām slodzēm uz iekšējās/ārējās paketes un laminēti stikli aizstāti ar ekvivalenta biezuma monolītiem. Dlubal RFEM modeli ģenerēts ar šī pētījuma laikā izveidoto programmu, kas aprakstīta nodaļā 8.

Aprēķināta 1150mm plata, 3500mm augsta stikla pakete ar 6+6mm laminētu stiklu ārpusē, 5+5mm laminētu stiklu iekšpusē un 14mm biezu starpslāni.

Detalizēts aprēķins un būvinženierijas procesa apraksts ir ārpus šī darba sfēras (*scope*). Tomēr, lai varētu vērtēt iegūtos rezultātus, svarīgi izprast kopējo drošības līmeni, kādu nodrošina drošības faktori būvnormatīvos. Atbilstoši nodaļā 6.2 parādītajam, vērtējot tikai drošības koeficientus, kas tiek pielikti papildus slodzēm un materiālam, tiek iegūta teorētiska stiprības “rezerve” vismaz $1.2 \times 1.35 = 1.62 = 62\%$. Šāda “rezerve” ļauj vērtēt vienkāršojumu ietekmi uz aprēķinu.

Tabulā zemāk parādīts deformāciju salīdzinājums pie 10 slodžu kombinācijām ar lielāko izlieci:

Tabula 4.1 Deformāciju salīdzinājums GEM modeļiem ar un bez inženierijas vienkāršojumiem

Nr.p.k	Kombinācijas apraksts	Mepla SJ modelis, izlieces [mm]	Dlupal RFEM vienkāršots modelis, izlieces [mm]	
1	SLS Winter +wind gov. line no PVB	8.55	8.62	1%
2	SLS Summer +wind gov. line no PVB	8.31	8.24	-1%
3	SLS Winter +wind gov. point no PVB	8.10	8.09	0%
4	SLS Winter +wind line gov. no PVB	7.95	8.05	1%
5	SLS Summer +wind gov. point no PVB	7.92	7.80	-2%
6	SLS Winter -wind gov. no PVB	7.52	7.47	-1%
7	SLS Summer +wind line temp gov. no PVB	7.43	7.28	-2%
8	SLS Winter +wind line temp gov. no PVB	7.25	7.32	1%
9	SLS Summer -wind gov. no PVB	7.19	7.05	-2%
10	SLS Summer +wind gov. no PVB	7.03	6.90	-2%

Redzams, ka noteicošajās kombinācijās deformāciju rezultāti vienkāršotam modelim, nelineāram, lielu deformāciju aprēķinam ir 2% robežās, kas vērtējams kā labs rezultāts.

Turpinājumā Tabula 4.2 parāda salīdzinājumu 10 slodžu kombinācijām ar lielākajiem principiālajiem spriegumiem.

Tabula 4.2 Maksimālo principiālo spriegumu salīdzinājums GEM modeļiem ar un bez inženierijas vienkāršojumiem

Nr.p.k	Kombinācijas apraksts	Mepla SJ modelis, spriegumi [MPa]	Dlupal RFEM vienkāršots modelis, spriegumi [MPa]	
1	ULS Winter +wind point gov. no PVB	26.91	25.88	-4%
2	ULS Winter +wind gov. point no PVB	26.43	25.82	-2%
3	ULS Winter +wind point temp gov. no PVB	23.12	22.45	-3%
4	ULS Summer +wind point gov. no PVB	22.91	21.91	-4%
5	ULS Summer +wind gov. point no PVB	22.23	21.84	-2%
6	ULS Winter +wind gov. line no PVB	21.58	22.21	3%
7	ULS Winter +wind line gov. no PVB	21.03	21.66	3%
8	ULS Winter no wind point gov. no PVB	20.28	18.99	-6%
9	ULS Winter +wind line temp gov. no PVB	18.85	19.36	3%
10	ULS Summer +wind gov. line no PVB	18.07	18.85	4%

Spriegumiem atšķirība ir lielāka nekā deformācijām, līdz 6% nelineāram aprēķinam, kas vērtējams kā viduvējs, bet akceptējams rezultāts. Spriegumiem lielākas izmaiņas rada precīzs galīgo elementu tīkla punktu izvietojums, kas programmās Dlupal RFEM un Mepla SJ ir nedaudz citādāks (abas programmas to ģenerē automātiski). Šo efektu palielina tas, ka konkrētajā salīdzinātajā gadījumā noteicošā ir punkta slodze, kas ir īpaši jūtīga uz galīgo elementu tīkla izmaiņām.

4.3 GEM modeļa parametru izvēle

4.3.1 Aprēķinu tips

Kā apskatīts CNR-DT 210 (Stikla projektēšanas vadlīnijas Itālijā) [53] 8.1 nodaļā redzamajos testos (Att. 4.2), deformācijas lineāram aprēķinam un “lielu-pārvietojumu” (*large deformations*) aprēķinam atšķiras būtiski. Jāpaskaidro, ka visi rezultāti it no GEM aprēķiniem, ar “FEM” šajā gadījumā ir saprasts aprēķins ar 3D telpiskajiem elementiem, bez vienkāršojumiem.

GEM aprēķina modelis tiks aprēķināts, lietojot ģeometriski-nelineāru, “lielu-pārvietojumu” aprēķinu.

Table 8.2. Comparison of the various solutions for rectangular laminated glass.

METHOD		Maximum deflection at the SLS	Maximum stress at the ULS
Wölfel-Ben-nison	linear	9.648 mm	18.47 MPa
	Non-linear	8.44 mm +14%	16.38 MPa +13%
EET	linear	11.79 mm	20.42 MPa
	Non-linear	9.72 mm +21%	17.19 MPa +19%
FEM	linear	11.51 mm	21.40 MPa
	Non-linear	9.679 mm +19%	15.01 MPa +42%

Table 8.3. Comparison of the various solutions for square-shaped laminated glass.

METHOD		Maximum force at the SLS	Maximum stress at the ULS
Wölfel-Ben-nison	linear	8.937 mm	14.86 MPa
	Non-linear	7.825 mm +15%	13.74 MPa +8%
EET	linear	12.12 mm	17.25 MPa
	Non-linear	9.634 mm +26%	14.79 MPa +17%
FEM	linear	12.33 mm	18.02 MPa
	Non-linear	9.469 mm +30%	13.58 MPa +33%

Att. 4.2 - Izvilkumi no CNR-DT 210 veiktajiem salīdzinājumiem starp lineāru un lielu-pārvietojumu aprēķiniem. Procentu vērtības norāda, cik daudz lineāra aprēķina/deformācijas vērtība atšķiras no nelineāra aprēķina

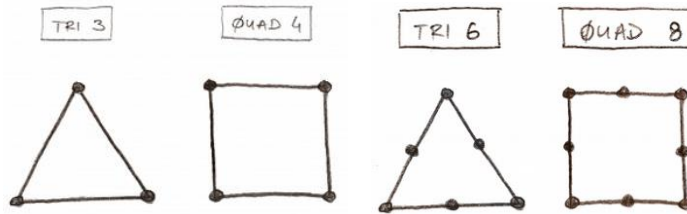
4.3.2 Galīgo elementu tips

Stikla slāņu biezums ir būtiski mazāks nekā pārējās divas dimensijas. Stikla loksne, galvenokārt strādā liecē, bet, tiklīdz tiek ievērtēta deformētā forma, tā papildus stingumu dod arī spēki stikla loksnes plaknē.

Iepriekš minētais raksturojums nozīmē, ka jālieto 2D čaulas (*shell*) tipa elementi.

Kā atzīmēts [54], lietojot pirmās pakāpes elementus (Att. 4.3), loksnes stingums var tikt novērtēts par lielu, tādēļ vēlams lietot otrās pakāpes galīgos elementus. Šādus otrās pakāpes 2D čaulas elementus lieto arī iepriekš minētās programmas “Mepla SJ” un “Glastik”, kā arī “Dlubal RFEM” (vispārīgos plātnes aprēķinos).

Aprēķinos šajā darbā lietoti otrās pakāpes galīgie elementi, kā aprakstīts “Dlubal RFEM” rokasgrāmatā [55].



Att. 4.3 - Pirmās pakāpes (first order) galīgie elementi TRI3 un QUAD4, Otrās pakāpes (second order) galīgie elementi TRI6 un QUAD8 [54]

4.3.3 Galīgo elementu izmērs

Pēc manas, darba autora, pieredzes, tieši galīgo elementu tīklojuma izmērs un precīzs punktu novietojums, parasti rada lielākās atšķirības, ja tiek veikts salīdzinājums starp dažādām galīgo elementu programmām vai vienas programmas dažādiem aprēķiniem.

Galīgo elementu optimāls izmērs ir atkarīgs no apskatītās konstrukcijas tipa, slodžu veida un nepieciešamā rezultātu tipa (deformācijas vai spriegumi). Vispārīga vadlīnija ir lietot elementus ar maksimālo izmēru $1/10$ no laiduma izmēra – vienīgo literatūras atsauci gan šim var atrast dzelzsbetona plātņu aprēķina metodikā [56]. Tomēr šāds izmērs būs nepietiekams gadījumiem, kad pielikta punkta slodze. Punkta slodzes apgabalā lietoti galīgie elementi ar izmēru $1/2$ no punkta slodzes izmēra.

5. GEM MODEĻU ĢENERĒŠANA UN REZULTĀTU DATU KOPAS IZVEIDE

Darba ievadā aprakstīti piedāvātie risinājumi projektēšanas procesa uzlabošanai. Šajā nodaļā aprakstīts otrais no šiem soļiem – aprēķina modeļu automatizēta ģenerēšana un aprēķins, lai iegūtu datu kopu, kas tālāk lietojama mašīnmācīšanās modeļu izveidei. Iepriekšējās divās nodaļās noskaidrojām, kādus vienkāršojumus veiksīm, kāds būs viens aprēķināmais GEM modelis, kādu datu diapazonu izvēlēsimies. Šajā nodaļā veiksīm pašus aprēķinus – krāsim un analizēsim datu kopu ar dažādu galīgo elementu modeļu aprēķinu.

5.1 Aprēķinu veikšana un datu krāšana

5.1.1 Vispārīgs procesa apraksts

Galīgo elementu modeļu ģenerēšana un aprēķins veikts izmantojot Dlubal RFEM programmatūru, versijas 5.19 un 5.21. Programma strādā Windows vidē. Lielā aprēķinu skaita dēļ tie tika veikti uz 6 datoriem paralēli. Kopumā aprēķinu veikšanā tika veikti šādi pieci posmi:

1. Šī darba ietvaros izveidota palīg-programma tika vietota uz visiem 6 datoriem. Pie katra eksperimenta aprēķina uzsākšanas, tā sūtīja HTTP pieprasījumu pēc datiem serverim;
2. Ģenerēti ievades dati. Ģenerēšanu veica uz servera izvietots skripts, kas uz katra datora HTTP pieprasījumu pēc datiem pretī nosūtīja ievades datu kopu vienam eksperimentam.
3. Veidots galīgo elementu modelis. Kad datu kopa tika saņemta, atbilstoši tai tika ģenerēts galīgo elementu modelis un palaists tā aprēķins, nolasīti rezultāti – maksimālais principiālais spriegums un maksimālā izliece.
4. Iegūtie rezultāti tika nosūtīti serverim kā HTTP pieprasījums.
5. Skripts servera pusē saņēma rezultātus un tos pievienoja kā jaunu rindu teksta failā.

Servera iesaiste ļāva organizēti veikt aprēķinus, veicot minimālu iestatīšanas darbu uz katra no aprēķinus veicošajiem datoriem un rezultātus saglabāt vienā vietā uz servera.

5.1.2 Ieejas datu ģenerēšana un rezultātu saglabāšana

Šajā nodaļā aprakstīti skripti, kas ģenerē datus un saglabā rezultātus. Tie ir glabāti uz servera un izsaukti no palīgprogrammas uz katra no vairākiem aprēķinus veicošajiem datoriem.

Ieejas datu ģenerēšanai izveidotais Python skripts ievietots 1. pielikumā. Būtiskais šajā skriptā:

- Nejaušo datu ģenerēšanā tiek lietots “random.uniform” vienmērīgais sadalījums. Tas nozīmē, ka katrs no ievades parametriem lielā datu kopā tieksies būt vienmērīgi sadalīts visā apgabalā.
- Pirms katra no parametru ģenerēšanas tiek iestatīta jauna sēkla (“seed”) priekš nejaušu skaitļu ģenerēšanas. Šī “sēkla” ir atkarīga gan no laika, gan arī unikāla reizinātāja katram parametram. Tas nodrošina, ka tiks lietotas dažādas “sēklas” katrā skripta izsaušanas reizē (laika iespaids). Vienā izsaušanas reizē katra parametra ģenerēšana būs neatkarīga (unikāla reizinātāja iespaids). Šī pieeja tika lietota pēc tam, kad darba procesā tika ģenerēti daļa nederīgu – atkārtotu un savstarpēji korelētu datu.

Pēc izsaušanas šis skripts atgriež tekstu, kura piemērs izskatās šādi:

```
2.7423376683667757 1.085048091860993 0.017217318590873214 21226.811001935293 0
0.1 0.0 376.2741019112764 0.09146340304699005 -1 0.10850480918609931 0.05
2927905242327172399
```

Šis ir ar atstarpēm atdalīts teksts, kurš attiecīgi secīgi satur šādus parametrus:

- Stikla paketes platumu [m]
- Stikla paketes augstumu [m]
- Stikla paketes biezumu [m]
- Izklidētas slodzes lielumu [N/m²]
- Punkta slodzes lielumu [N]
- Punkta slodzes izmēru [m]
- Punkta slodzes spiedienu (atkarīgs no lieluma un izmēra) [N/m²]
- Līnijas slodzes lielumu [N/m]
- Līnijas slodzes attālumu no stikla paketes apakšas [m]
- Vai izklidētā slodze ir tajā pašā virzienā, kurā līnijas un punkta slodze [-1/+1]
- Galīgo elementu tīkla maksimālais izmērs vispārīgi [m]
- Galīgo elementu tīkla izmērs punkta slodzes pielikšanas vietā [m]
- Jaucēj vērtība (*hash*), kas veido unikālu izvadi atkarībā no visiem iepriekš minētajiem mainīgajiem. Būtiska, lai pārbaudītu dublikātus datu kopā.

Savukārt aprēķinu rezultāti uz katra datora esošajā palīg-programmā tiek sūtīti kā HTTP pieprasījums un pievienoti teksta failam ar ļoti vienkārša PHP skripta palīdzību:

```

1. <?php
2.     $received_data = $_POST['data'];
3.     file_put_contents('data.txt', "$received_data", FILE_APPEND);
4.     echo "ok";
5. ?>

```

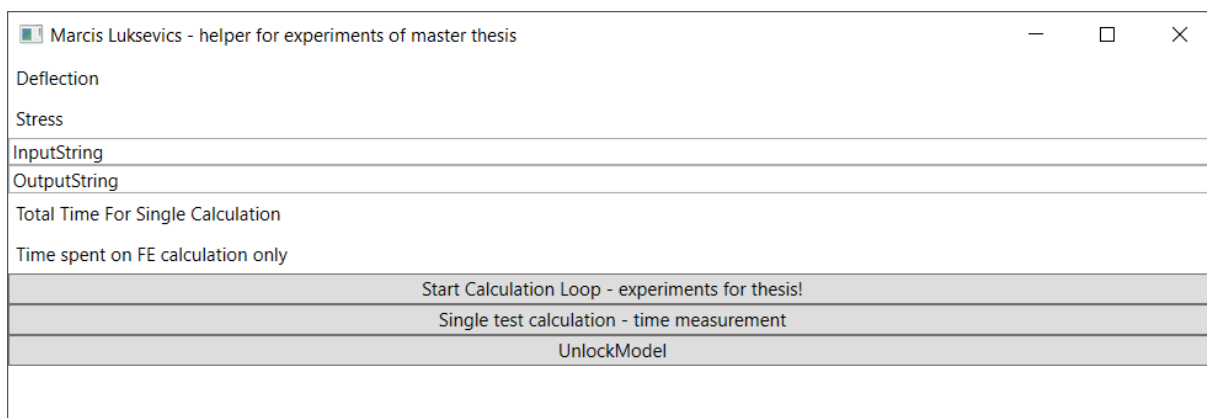
5.1.3 Palīg-programma GEM aprēķinu veikšanai

Lai veiktu GEM aprēķinus uz katra no vairākiem datoriem, tika izveidota C#.NET palīg-programma ar WPF lietotāja saskarni.

Pilns programmas pirmkods ir pievienots 2. pielikumā, un saskarne redzama Att. 5.1.

Šāda programma tika izvietota uz visiem 6 aprēķinus veicošajiem datoriem, tā tika palaista un attiecīgi tā vadīja procesu, kā aprakstīts nodaļā 5.1.1 – pieprasīja ievades datus, ģenerēja modeli, nolasīja rezultātus un nosūtīja rezultātu datus uz serveri.

No šīs programmas kā būtisks aspekts jāpiemin, ka ātrdarbības paātrināšanai visas darbības notiek operatīvajā atmiņā. Tas ir – sākot aprēķināt eksperimentus, tiek atvērta programma, izveidots jauns fails un visas tālākās darbības notiek šajā failā, bez saglabāšanas uz cietā diska.



Att. 5.1 - Izveidotā palīgprogramma GEM aprēķinu datu kopas iegūšanai

5.2 Iegūto datu kopas analīze

5.2.1 Datu skaits

Lietojot nodaļā 5.1 aprakstīto procesu, tika veikti kopā 2.4 miljoni galīgo elementu aprēķinu. Kā aprakstīts nodaļā 6.4.2, eksperimenti tika veikti ar dažādiem slodžu pielikšanas veidiem:

- Vispārīgs gadījums – pielikta gan izkļiedētā, gan punkta, gan līnijas slodze;
- Tikai izkļiedētā slodze;
- Izkļiedētā slodze un līnijas slodze;

- Izklīdētā slodze un punkta slodze;

Tabula 5.1 Apkopojums par aprēķināto eksperimentu skatu

Slodzes tips	Datu pielietojums	Datu skaits
Vispārīgs gadījums	Modeļa trenēšana	981 144 (49% no treniņu datiem)
Tikai izklīdētā slodze	Modeļa trenēšana	487 088 (24% no treniņu datiem)
Izklīdētā slodze un līnijas slodze	Modeļa trenēšana	274 186 (14% no treniņu datiem)
Izklīdētā slodze un punkta slodze	Modeļa trenēšana	262 484 (13% no treniņu datiem)
	Kopā dati modeļu trenēšanai:	2 004 902
Vispārīgs gadījums	Testa dati	100 000
Tikai izklīdētā slodze	Testa dati	100 000
Izklīdētā slodze un līnijas slodze	Testa dati	100 000
Izklīdētā slodze un punkta slodze	Testa dati	100 000
	Kopā testa dati:	400 000

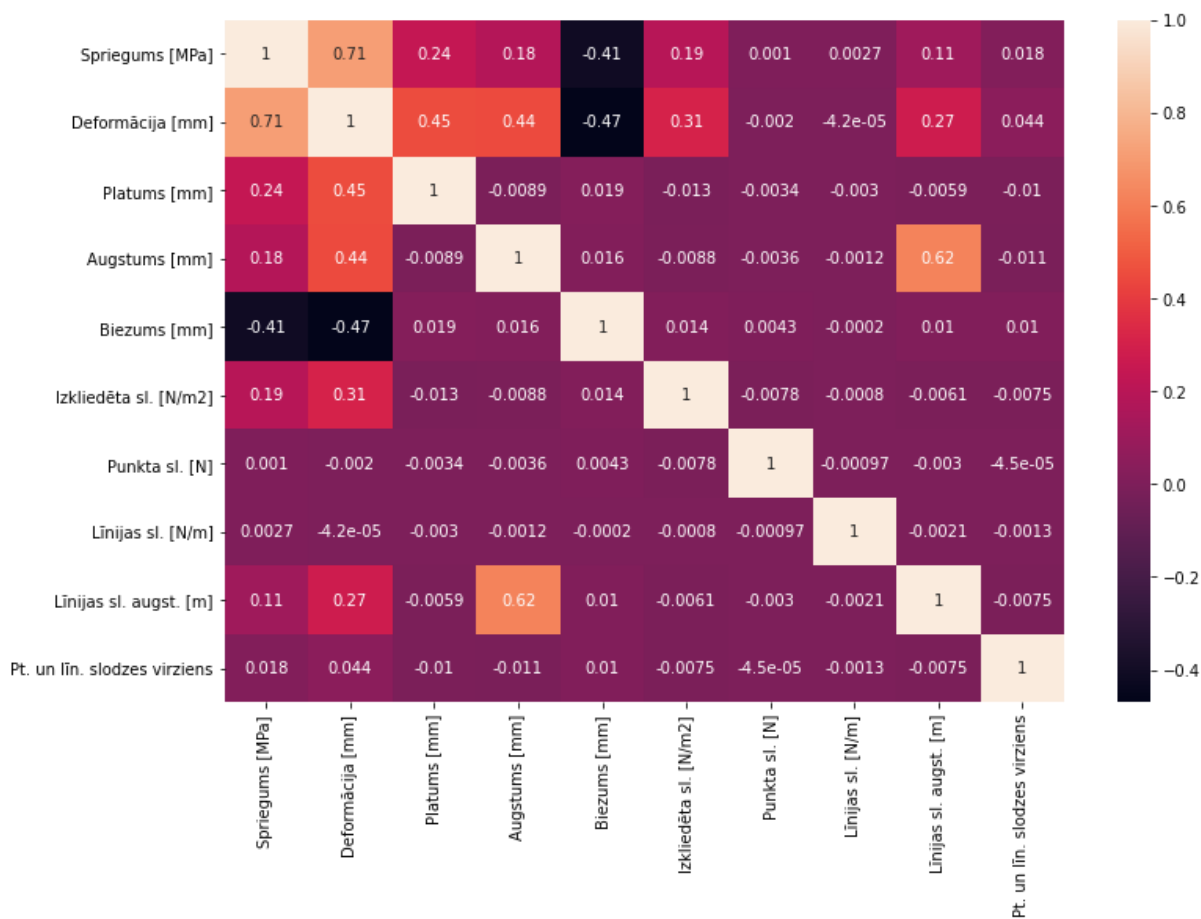
Turpmākajās nodaļās analizēti modeļa trenēšanā lietotie dati. Tā kā testa dati ir ģenerēti tieši tāpat, tad tiek uzskatīts, ka to īpašības būs līdzīgas. Pēc ~2 miljonu datu sasniegšanas datu ģenerēšana tika turpināta, tādēļ testiem tika izvēlētas 100 000 nejaušu eksperimentu lielas datu kopas no turpmāk ģenerētajiem datiem.

Tāpat jāpiebilst, ka lietotais automatizētais mašīnmācīšanās risinājums Google AutoML Tables lieto 10% no datiem testēšanai savas sistēmas ietvaros. Tā kā šis komerciālais risinājums nav atvērtā koda risinājums un tā darbība nav plaši aprakstīta no izstrādātāju puses, tad tika izlemts modeļu precizitātes pārbaudei lietot atsevišķu datu kopu, kas nav tikusi padota Google automatizētajam risinājumam.

Modeļa treniņa un testa datos netika lietoti dati, kuros deformācija vai spriegums ir mazāks par 0.1mm vai 0.1MPa – datu precizitāte faktiski nav svarīga šajā apgabalā (skatīt nodaļu 6.2), un pie tik mazām vērtībām iespējamā relatīvā kļūda ir liela.

5.2.2 Parametru korelācija

Datiem, kas padoti galīgo elementu aprēķinu veikšanai, būtu jābūt neatkarīgiem vienam no otra, lai pilnībā aptvertu visu iespējamo ievades datu kopu. Savukārt ievades parametru korelācija ar rezultātu (spriegumu vai izlieču) datiem norāda uz to, kuri no ievades parametriem visvairāk ietekmē rezultātus. Att. 5.2 parādīti savstarpējie Pīrsona korelācijas koeficienti vispārīga gadījuma datu kopai (ar visu tipu slodzēm) ar 981 144 mēģinājumu.



Att. 5.2 Pīrsona korelācijas koeficienti treniņa datiem ar pieliktām visām slodzēm

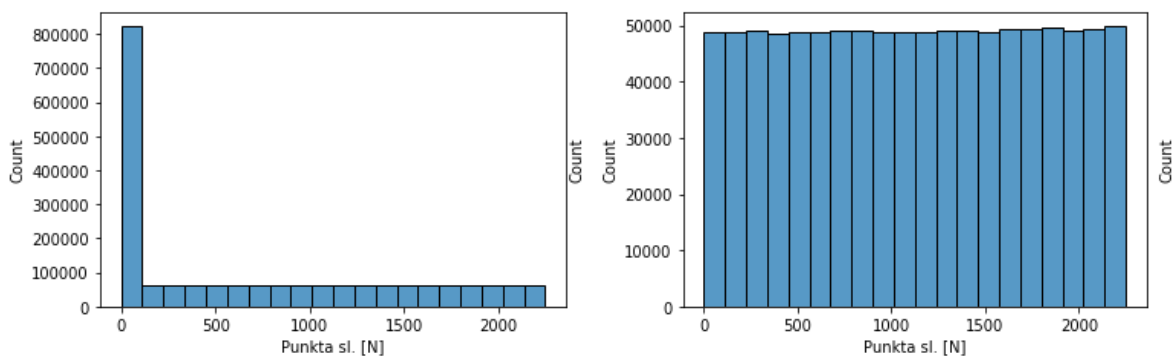
Šajā attēlā redzamas šādas īpašības:

- Koeficientu absolūtās vērtības vairumā gadījumi ir ļoti mazas, tas liecina, ka ievades dati galīgo elementu aprēķinos ir neatkarīgi viens no otra.
- Izņēmums ir līnijas slodzes augstums, kas korelē ar stikla paketes augstumu. Tas bija sagaidāms, jo datu ģenerēšanā slodzes augstums tiek izvēlēts kā daļa no paketes kopējā augstuma. Tas veikts, lai neradītu gadījumus, kad slodzes augstums ir lielāks par pašu paketes augstumu.
- Lielākai deformācijai atbilst lielāki spriegumi.
- No slodzēm būtiski lielāka ietekme uz izlieci ir izkliedētajai slodzei.
- Lielākiem stikla izmēriem atbilst lielākas izlieces un spriegumi. Lielākam biezumam – mazāki.

Korelācijas koeficienti gadījumiem, kuros kāda no slodzēm nav pielikta, ir novērojamas tādas pašas sakarības, kā Att. 5.2 un tie nav atsevišķi iekļauti šajā darbā.

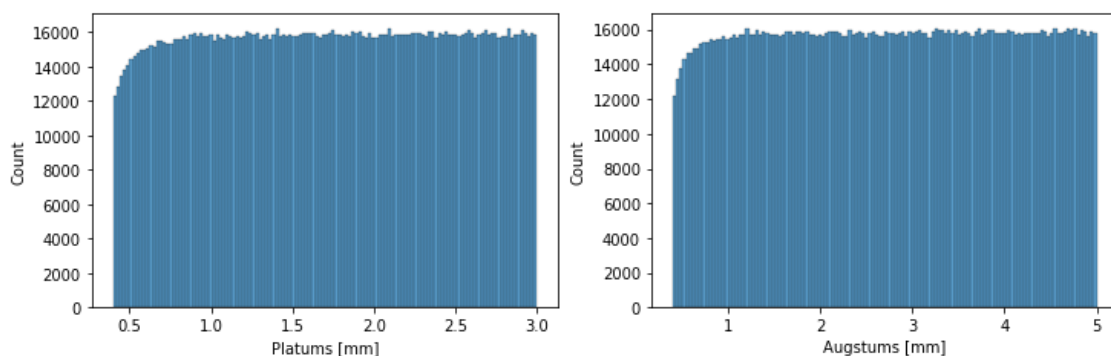
5.2.3 Datu sadalījums

Parametriem, kas lietot kā ievades dati aprēķinos, būtu jāatbilst vienmērīgam sadalījumam, aptverot no minimālās līdz maksimālajai vērtībai. Tika aplūkotas datu histogrammas, kas apliecina vienmērīgo sadalījumu. Att. 5.3 redzamas Punkta slodzes spēka histogrammas – visiem ~2 miljoniem datu un tikai gadījumam ar visām pieliktām slodzēm. Redzams kā pie visiem datiem izceļas datu daļa, kuru vērtība ir 0 (attiecīgi – slodze nav pielikta).



Att. 5.3 Histogramma ar punkta slodzes sadalījumu pie visiem datiem (pa kreisi) un datiem ar visām pieliktām slodzēm (pa labi)

Aplūkojot visus datus, arī redzams, ka platumam un augstumam ir neliels samazinājums pie ļoti maziem izmēriem – tas saistāms ar to, ka trenēšanā netika lietoti dati ar spriegumu vai deformāciju mazāku par $<0.1\text{mm}$ un $<0.1\text{MPa}$. Tik mazas vērtības ir tālu no nodaļā 6.2 aprakstītajiem apgabaliem, kas ir nozīmīgi normatīvu pārbaudēs.



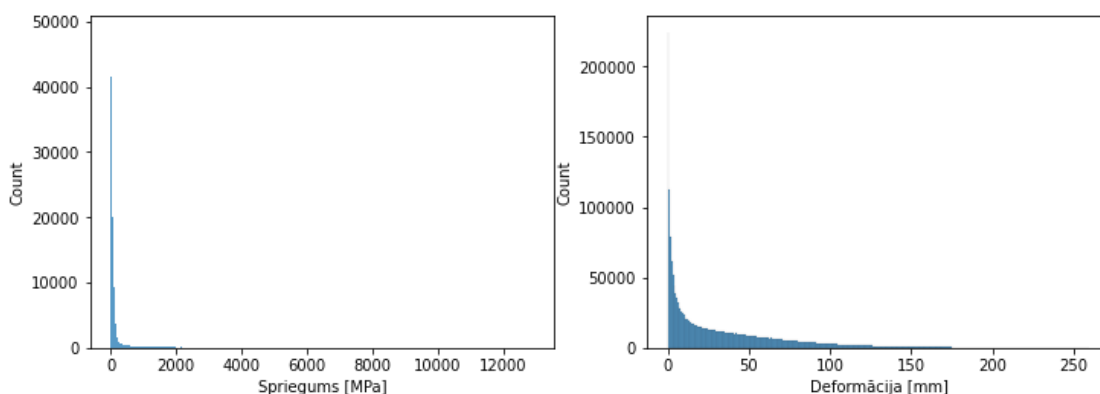
Att. 5.4 Histogrammas ar platumu un augstuma vērtību sadalījumu pie visiem datiem

Ieejas datiem tikai pārbaudītas arī to vidējās, maksimālās un minimālās vērtības – tās sakrīt ar nodaļā 3.2 aprakstītajām robežām, skat. Tabula 5.2.

Tabula 5.2 Veikto mēģinājumu ievades datu apkopojums

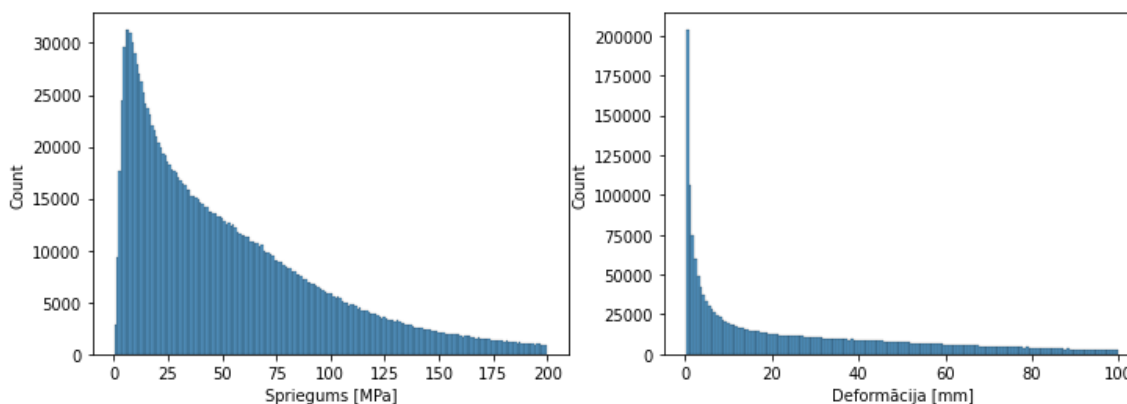
	Vidējā vērtība	Standartnovirze	Minimums	Mediāna	Maksimums
Spriegums [MPa]	268.38	756.92	0.12	54.15	12899.60
Deformācija [mm]	32.20	36.97	0.10	18.06	259.89
Platums [m]	1.72	0.74	0.40	1.72	3.00
Augstums [m]	2.72	1.32	0.40	2.73	5.00
Biezums [m]	0.0139	0.0057	0.0040	0.0138	0.0240
Izklidēta Ls. [N/m ²]	12646.31	7176.71	0.01	12687.95	24999.99
Punkta sl. [N]	699.87	749.55	0.00	439.39	2250.00
Līnijas sl. [N/m]	705.01	748.71	0.00	454.64	2250.00
Līnijas sl. augst. [m]	0.43	0.55	0.00	0.20	2.50

Apskatot iegūtos spriegumu un deformāciju datus, histogrammās redzam, ka spriegumiem ir izlēcēji līdz pat 12646MPa vērtībai, bet vairums datu ir diapazonā līdz 200MPa (mediāna 54MPa). Arī deformācijām maksimums 260mm ir tālu no mediānas 18mm.



Att. 5.5 Spriegumu un deformāciju histogrammas - visa datu kopa

Kā norādīts nodaļā 6.2, normatīvos maksimāli atļautie spriegumi ir līdz 88MPa un izlieces līdz 46mm. Histogrammas ar robežām līdz 200MPa un 100mm parādītas Att. 5.6. Kopumā 83% no spriegumu datiem un 93% no deformāciju datiem ir šajā diapazonā, kas nozīmē, ka vairums datu apraksta diapazonu, kurā praktiski notiek stikla pakešu nestspējas un izlieču pārbaudes.



Att. 5.6 Spriegumu un deformāciju histogrammas - spriegumi līdz 200MPa, deformācijas līdz 100mm

6. MAŠĪNMĀCĪŠANĀS MODEĻU TRENĒŠANA UN LIETOŠANA

Darba ievadā aprakstīti piedāvātie risinājumi projektēšanas procesa uzlabošanai. Šajā nodaļā aprakstīts trešais no šiem soļiem – mašīnmācīšanās modeļu izveide, lai būtiski paātrinātu vienas stikla paketes konfigurācijas aprēķinu. Iepriekšējā nodaļā ieguvām un analizējām datu kopu. Šajā nodaļā mēģināsim šo kopu “aprakstīt” ar mašīnmācīšanās modeļiem.

6.1 Kritēriji modeļu izvērtēšanai

Mašīnmācīšanās modeļiem izvirzītas divas galvenās prasības – precizitāte un prognozēšanas ātrdarbība.

Precizitāte. Modeļiem ir jābūt ar mazu relatīvo kļūdu. Turklāt jābūt ne tikai mazai vidējai relatīvajai kļūdai, bet kļūdām jābūt pēc iespējas vienmērīgāk sadalītām. Skatīt vairāk nodaļā 6.2.

Prognozēšanas ātrdarbība. Kā aprakstīts nodaļā 4.2.1, vienas paketes izvērtēšanai nepieciešamas aptuveni 40 – 240 prognozes. Viens GEM aprēķins, kurš tiek aizstāts ar ekvivalentu mašīnmācīšanās prognozi, aizņem aptuveni 3 sekundes (skat. nodaļu 6.7.1). Mašīnmācīšanās modelis, kura ātrdarbība būtu līdzīga GEM aprēķiniem, zaudētu savu jēgu.

6.2 Modeļu precizitātes izvērtēšana no būvinženierijas skatpunkta

Mašīnmācīšanās modeļu lietošanas mērķis ir izskatīt daudzus stikla paketes uzbūves variantus, lai varētu izvēlēties vienu, kurai veikt galīgo elementu metodes aprēķinu. Izvēlēta gala risinājuma atbilstība normatīvam ir jāpierāda klientam un/vai būvniecību regulējošām institūcijām – es, darba autors, uzskatu, ka šeit liela nozīme ir tradīcijām industrijā, kuras pārskatāmā nākotnē vēl uz mašīnmācīšanās modeļiem skatīsies ar piesardzību.

Iepriekš minētā nepieciešamība pēc viena, “fināla” aprēķina nemaina uzdevuma būtību – trenēt modeļus, kuru rezultāti pēc iespējas sakrīt ar GEM aprēķinu rezultātiem.

Pirmais aspekts, kas jāņem vērā – būvinženierijā noslogotība (*utilization*) tiek mērīta procentuāli. Tas cieši saistās ar robežstāvokļu (*limit state*) pieeju, kas aprakstīta EN1990 [51] un pielietota arī šajā darbā apskatītajos DIN18008 un EN16612. Praktiski šī pieeja nozīmē, ka tiek lietoti dažādi drošības koeficienti jeb reizinātāji. Drošības koeficienti tiek lietoti gan pieliktajām slodzēm, gan materiālu pretestībām. Pēc šī principa izriet, ka arī mašīnmācīšanās modeļu precizitāte ir jāmēra pēc to relatīvās kļūdas.

Otrais aspekts. Iepriekš pieminētie drošības koeficienti.

- Gan EN16612, gan DIN18008 lieto dalītāju 1.8 parastam stiklam. Koeficients rūdītam stiklam ir attiecīgi 1.2 un 1.5. Šie reizinātāji tiek piemēroti stikla īpašībām, par kurām drīkst būt zemākas tikai 5% no testētajām vērtībām, t.i., ir 95% iespējamība, ka vērtība būs lielāka.
- Drošības pārbaudēm slodzēm tiek piemēroti drošības koeficienti. Gan EN16612, gan DIN18008 patstāvīgās slodzes (pašsvars, spiediena izmaiņas ēkas augstuma dēļ) tiek reizinātas ar 1.35, mainīgās slodzes (vējš, iemītnieki) ar 1.50. Mainīgās slodzes tiek vērtētas ar iespējamību 1 reizi 50 gados. 50 gadi ir tipisks pieņemtais ēkas kalpošanas ilgums (kategorija “4” atbilstoši EN1990 [51] nodaļai 2.3).

Šis iezīmē kopīgu būvniecības industrijas tendenci – augstas “rezerves”, kas šajā gadījumā būs no $1.2 \times 1.35 = 1.62$ līdz $1.8 \times 1.5 = 2.7$ jeb 62% - 170%. Izvērtējot iepriekš minēto, mans, darba autora, sertificēta būvinženiera, vērtējums ir pieņemts par akceptējamu mašīnmācīšanās modeļu kļūdu 5% robežās. Piemērojot “trīs-sigma” principu, 99.7% no relatīvajām kļūdām būtu jābūt mazākām par 5%.

Trešais aspekts saistās ar relatīvo kļūdu sadalījumu. Ideālā gadījumā pilnīgi visas prognozes būtu zem 5% robežas un teorētiski atļautu 95% modeļa precizitāti, ja visos gadījumos kļūda būtu vienāda. Tomēr, kā redzēsīm nodaļās 6.5 un 6.6, relatīvās kļūdas būtiski variēs. Labs modelis būs tāds, kurā relatīvā kļūda variēs maz, jo tas nodrošinās skaidru uzticamības līmeni.

Ceturtais aspekts saistās ar maksimālo pieļaujamo kļūdu. Ideālā pasaulē tā nebūtu lielāka par 5%, tomēr praktiski, pie liela mēģinājuma skaita, būs atsevišķas prognozes-izlēcēji. Šajā gadījumā izvēlēts maksimālais limits, kas ir vienāds ar minimālo slodžu drošības koeficientu – neviena no kļūdām nedrīkst pārsniegt 35% robežu.

Piektais aspekts saistās ar apgabaliem, kuros svarīgi vērtēt precizitāti. Kā rāda Tabula 5.2, treniņu datos spriegumi variē no 0.1 – 12900MPa un izlieces 0.1 – 260mm. Atbilstība normatīvu prasībām ir binārs lēmums – atbilst/neatbilst. Pārsniedz noteiktu robežu vai nepārsniedz.

Ievērtējot visus normatīvos minētos drošības koeficientus un stikla tipus, stiklā atļaujamie spriegumi variē no 7.2MPa parastam stiklam pie pastāvīgajām slodzēm atbilstoši EN16612 līdz 88.0MPa rūdītam stiklam pie īstermiņa slodzēm atbilstoši DIN18008. Kļūdas, kas notiek tālu no šī nozīmīgā diapazona, nav būtiskas stikla paketes ievērtēšanai. Piemēram, ja 1MPa vietā tiek prognozēti 3MPa, tad šī 200% kļūda praktiski nav būtiska. Spriegumu prognožu vērtēšanā iekļauti tikai tie dati, kuru patiesās vai prognozētās vērtības ietilpst diapazonā 7.2-88MPa

Stiklam pieļaujamās deformācijas ir atkarīgas no tā īsākās malas izmēra L. Un ir attiecīgi L/65 atbilstoši EN16612 vai L/100 atbilstoši DIN18008. Atbilstoši nodaļā 3.2 aprakstītajam,

šajā darbā apskatītas paketes ar izmēru no 400mm līdz maksimālajam platumam 3000mm. Attiecīgi būtiskais diapazons ir no $400\text{mm}/100 = 4\text{mm}$ līdz $3000\text{mm}/65=47\text{mm}$. Izlieču prognožu vērtēšanā iekļauti tikai tie dati, kuri patiesās vai prognozētās vērtības ietilpst diapazonā 4mm-47mm.

6.3 Mašīnmācīšanās satvara izvēle

Lai sasniegtu darba mērķi, tiek izstrādāta programmatūra, kurā mašīnmācīšanās ir tikai viena no daļām kopējā procesā. Literatūras apskatā secinājām, ka nesenā mašīnmācīšanās risinājumu attīstība ir virzījusies uz automatizētiem risinājumiem, padarot tehnoloģiju pieejamu arvien plašākam cilvēku lokam un radot iespēju sasniegt labus precizitātes rezultātus relatīvi īstā laikā.

Kā potenciāli piemēroti šim darbam tika atlasīti divi jaunākie risinājumi no nodaļā 2.3 minētajiem – “Google AutoML Tables” un “Amazon AutoGluon”. Pirmais no šiem risinājumiem tika izvēlēts, jo Google kopā ar pašu risinājumu piedāvā relatīvi lielas jaudas trenēšanas infrastruktūru (skat. 6.3.1). Otrais no šiem risinājumiem tika izvēlēts, jo tas ir jaunākais no visiem un attiecīgi risinājumu pavadošajā pētījumā [16] ir parādīts, ka tas precizitātes ziņā ir līdzvērtīgs vai labāks par citiem, vecākiem risinājumiem.

Kā trešais no risinājumiem tika apskatīts “Microsoft Azure AutoML”. Lai gan tas nav aprakstīts/salīdzināts citos pētījumos, arī Azure AutoML ir relatīvi jauns (2019. gada) satvars, kurš spēj darboties ar tabulāriem ievades datiem. Microsoft risinājums regresijas problēmām nepiedāvā lietot neironu tīklus – turpretī lieto “tradicionālākas” metodes, kā lēmumu kokus un *random forest*, kombinējot tos ar dažādiem ievades datu normalizācijas veidiem.

6.3.1 Teorētisks iespēju salīdzinājums

Pirmkārt, tā kā mašīnmācīšanās modeļi ir lietoti, lai noteiktu maksimālās izlieces un spriegumus – tad lietotajam satvaram ir jāatbalsta modeļi, kas veic regresiju (nevis klasifikāciju). Visi trīs izvēlētie to atbalsta.

Visi izvēlētie risinājumi darbojas ne tikai ar pašu modeļu arhitektūras izvēli un trenēšanu, bet arī datu sagatavošanu – tiem tiek padoti ievades dati .CSV (vai citā) neapstrādātā formātā un satvari paši veic problēmu identifikāciju un datu priekšapstrādi lai sāktu trenēt modeļus. Jāpiebilst, ka šī darba ietvaros liela daļa no ievades datu apstrādes ir savā ziņā ne-aktuāla. Tā kā dati ir speciāli ģenerēti lietošanai mašīnmācīšanās modeļos, tad tiem nav iztrūkstošu vai kļūdainu datu. Līdzīgi – nav sagaidāms, ka kādus no parametriem varēs atmest vai apvienot –

no būvinženierijas perspektīvas jau zināms, ka katrs parametrs ietekmē rezultātu. Savukārt nodaļā 5.2.2 jau apskatījām, ka ievades dati savstarpēji nekorelē.

Katram no trim satvariem ir dažāds “caurspīdīguma” līmenis. “Amazon AutoGluon” ir atvērtā koda risinājums, viss kods pieejams “Github” repozitorijā [57], kā arī plaša lietotāja dokumentācija atsevišķā mājas lapā [58], to apraksta arī zinātniska publikācija [16]. “Microsoft Azure AutoML” ir pieejama dokumentācija mājaslapā [59], lietotie algoritmi regresijas veikšanai nav inovatīvi, savukārt šo modeļu ansambļu veidošanai ir atsauce uz Caruana algoritmu [60]. Visnoslēpumainākais no trim ir “Google AutoML Tables”. Tā dokumentācija mājaslapā [25] ir tendēta uz tā iespēju aprakstu un ļoti maz pastāsta par pašu modeļu veidošanas procesu. Tā kods nav atvērts pieejams, bet, tā kā rezultātā tiek iegūti “Tensorflow” modeļi, tad noprotams, ka modeļu veidošanai tiek izmantots šis atvērtā koda satvars.

Visi trīs risinājumi ir pieejami kā API, kurus var vadīt ar Python skriptiem. Google risinājuma gadījumā pieejams arī API, kur komandas var dot kā HTTP pieprasījumus. Google un Microsoft risinājumi ir pieejami arī caur tīmekļa vietnēm un tos var lietot, neuzrakstot nevienu koda rindiņu. Amazon risinājuma gadījumā ir tas ir pieejams kā Python bibliotēka, ko var instalēt ar PIP palīdzību.

No resursu prasīguma viedokļa būtiski atšķiras ieteiktās aprēķina jaudas. Google savam risinājumam piedāvā fiksēta izmēra infrastruktūru - 92 mašīnas paralēli ar 4x vCPU un 15GB operatīvo atmiņu. Amazon risinājumu var uz gandrīz jebkuras darbstacijas. Lietojot “Amazon Web Services” datoru jaudas, rekomendācija [61] ir lietot “m5.24xlarge” mašīnu ar 96 vCPU un 384GB RAM. Kā liecina nosaukums, šī virtuālā mašīna ir līdzīga 24 mašīnām ar 4x vCPU un 16GB RAM. Microsoft Azure AutoML rekomendē būtiski mazāku jaudu – 1-2 mašīnas ar 4x vCPU un 14 GB RAM. Tātad, ieteiktā jauda starp Google/Amazon/Microsoft risinājumiem, ir attiecībā 92/24/2.

Katram no satvariem regresijas uzdevumiem ir atbalstītas dažādas zaudējuma funkcijas (*loss functions*). Interesanti, ka tiešā veidā nav nevienas zaudējuma funkcijas, kuru atbalsta visi satvari. Var uzskatīt, ka visi piedāvā vidējo absolūto kļūdu un vidējo kvadrātisko kļūdu, lai gan Microsoft risinājums šos parametrus piedāvā kā normalizētus, nevis absolūtos lielumos. Papildus šīm divām kopējām funkcijām, Amazon piedāvā arī R^2 determinācijas koeficientu un centrālo (*median*) absolūto kļūdu. Microsoft piedāvā Spīrmana (*Spearman*) korelācijas koeficientu un arī R^2 . Google papildus piedāvā vidējo kvadrātisko logaritmisko kļūdu, kas izrādīsies ļoti piemērota mūsu uzdevumam (skat nodaļā 6.4.1).

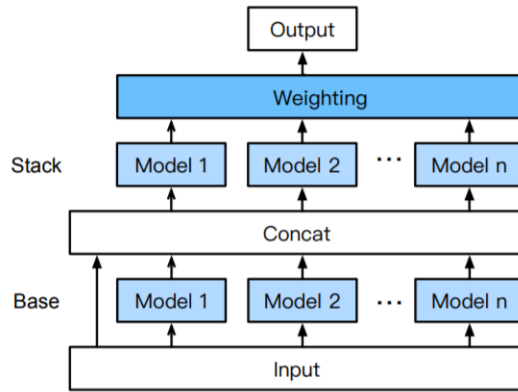
Kā minēts, visi trīs izvēlētie satvari veido modeļu ansambļus. Bet veids, kā tiek iegūts šis ansamblis, ir dažāds. Tālāk aprakstīti modeļu tipi un piegājiens tieši regresijas tipa modeļiem, kādus mēs esam paredzējuši veidot.

Microsoft Azure AutoML sākumā trenē vairākus modeļus atsevišķi. Šajos modeļos iekļaujas šādi tipi: “Elastic Net”, “Light GBM”, “Gradient Boosting”, “Lēmumu koki”, “Tuvāko kaimiņu metode”, “LARS Lasso”, “Stohastiskā gradienta metode”, “Random Forest”, “Xgboost”, “Fast Linear Regressor”. Kā redzam, netiek iekļauti neironu tīkli. Katrs no šiem modeļiem tiek mēģināts ar vienu vai vairākām skaitlisko ievades datu mērogošanas metodēm – normalizēšana, principiālo komponentu metode, mērogošana apskatot diapazonu starp min. un maks. vērtībām un citām (skat. pilnu sarakstu [59]). Pēc tam, kad atsevišķie modeļi ir satrenēti, tiek izveidoti divi ansamblī:

- “Balsošanas” ansamblis – kas atgriež svērtu vidējo vērtību no atsevišķo modeļu prognozētajām vērtībām.
- “*Stacking*” ansamblis – meta-modelis, kas tiek vēlreiz trenēts un kura ieejas dati ir atsevišķo modeļu prognozētās vērtības.

Abos šajos piegājienos modeļi tiek atlasīti pēc Caruana 2004. gadā izstrādāta algoritma [60] – mēģinot īsumā pārstāstīt tā būtību – tiek sākts ar tukšu ansambli, pēc tam modeļi likti pa vienam klāt un apskatīts, vai ansamblis ar modeli ir labāks nekā bez modeļa vai arī ne. Ansambļu pārbaudēs tiek lietoti validācijas kopas dati.

“Amazon AutoGluon” ir būtiski citādāks piegājiens kā citiem satvariem, jo uzmanība ir nevis uz pašu modeļu hiperparametriem, bet tieši modeļu ansambļa trenēšanu. Ansambļa vispārīgā shēma parādīta Att. 6.1 [16]. Katrā no modeļu slāņiem modeļi ir dažāda tipa modeļi – “LightGBM”, “CatBoost”, “XGBoost”, *Random forest*, *Extremely randomized trees*, tuvāko kaimiņu metode, lineārā regresija, neironu tīkls (viena konkrēta neironu tīkla arhitektūra). Attiecīgi, tiek trenēti šie modeļi pirmajā slānī un pēc tam vēlreiz otrajā slānī – bet otrajā slānī kā ievades dati ir gan sākotnējie parametri, gan pirmā slāņa modeļu rezultāti. Tālāk var būt vēl vairāki slāņi (šī darba ietvaros gan netika sasniegti vairāk kā 3 slāņi). Tā kā katrs atsevišķais modelis ir relatīvi vienkāršs, tad katram no tiem tiek lietots “Bagging” jeb “bootstrapping un aggregating” jeb vienkāršoti sakot – trenēti vairāki modeļi katru reizi kādu daļu no datiem atstājot kā “testa datus”. Tādējādi metode katra modeļa trenēšanā izmanto visus datus.

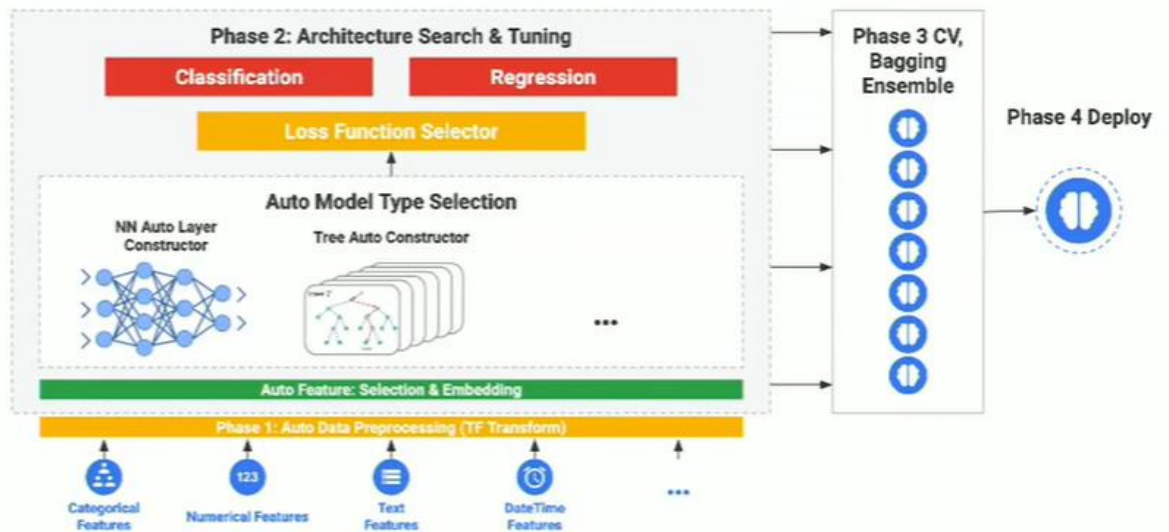


AutoGluon's multi-layer stacking strategy, shown here using two stacking layers and n types of base learners.

Att. 6.1 - AutoGluon modeļu ansambļa shēma [16]

Savādi, bet precīzākā informācija par Google AutoML Tables veidotajiem modeļiem ir atrodama kādā "YouTube" video, kurā izstrādātāji to prezentē [15]. Vispārīgā darbības shēma ir parādīta Att. 6.2.

Deepdive into multi-phases AutoML pipeline



Att. 6.2 Google AutoML Tables darbības plūsmas diagramma [15]

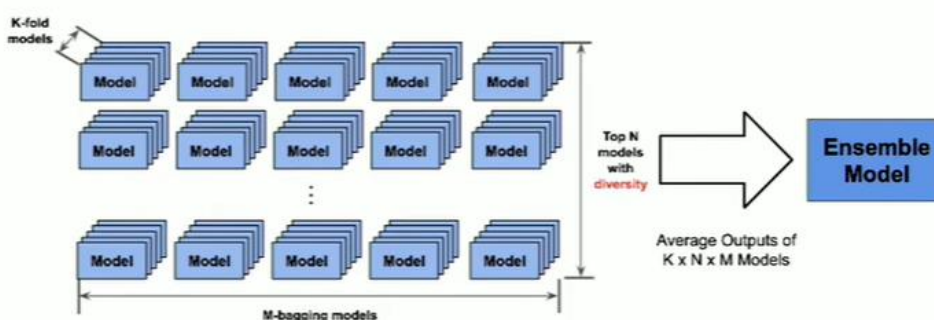
Skaidri ir aprakstīta datu priekšapstrāde, ko Google veic [25].:

- Datu normalizēšana;
- Vērtības pārvēršana uz *float32* tipu;
- Skaitliskās vērtības aizstāt ar vērtībām no 1 līdz 100 atbilstoši kvantīles vērtībai;
- Pārveidot lietojot funkciju $\log(\text{value}+1)$, ja vērtības ir lielākas par 0;
- Pārveidot lietojot funkciju $\log(\text{value}+1)$ un normalizēt pēc tam;
- Ja kolonnā, kurā ir skaitliskas vērtības, parādās teksts, tad šī datu rinda tiek izlaista.

- Izlecēju vērtības – lielas/mazas salīdzinoši ar citām netiek speciāli apstrādātas.

Kā arī lietotie modeļu tipi – “Adanet”, “Adanet” ansambļi, GBDT un neironu tīkli [25]. Pēc publiski pieejamās informācijas, neskaidrs ir veids, kā tiek izveidots ansambļa modelis. Arī pētījumā [16] šis norādīts kā neskaidrs. Tuvāko informāciju varam iegūt no Att. 6.3 [15]. kur redzams, ka tiek izvēlēta vidējā vērtība no dažādiem modeļiem, katra modeļa kross-validācijas modeļiem un “bootstrapping” (jeb “bagging”) modeļiem. Tomēr, kā vēlāk parāda mūsu pētījumā lietoto modeļu dati, netiek izvēlēti visi labākie modeļi ar dažādām arhitektūrām – tāpat notiek kāda sarežģītāka atlase.

Auto CV, Bagging and Ensemble



Att. 6.3 Google AutoML Tables modeļu ansambļa izveide [15]

Apkopojot uzzināto, pirmkārt jāmin, ka katrs no satvariem būtu vērts, lai tā priekšrocības un ēnas puses pētītu pamatīgi atsevišķos darbos. Google risinājums tiek darbināts ar vislielāko jaudu un pamatā balstās uz dažādiem neironu tīkliem. Microsoft risinājums ir diametrāli pretējs – ar metodēm bez neironu tīkliem un skaidri paskaidrotu ansambļu veidošanas loģiku. Amazon risinājums ir savā ziņā “alternatīvs” ar savu uzstādījumu, ka modeļi tiek izvēlēti no limitēta skaita “bibliotēkas”. Lai gan tas ir jaunākais, tas ir arī ar visbrīvāk pieejamo dokumentāciju/informāciju.

6.3.2 Precizitātes un ātrdarbības salīdzinājums, izvēlētais satvars

6.3.2.1 Precizitātes vērtējums

Mašīnmācīšanās satvara izvēlei tika izvēlēta datu kopa ar 500 tūkstošiem datu. Testēšanai tika lietotas 50 tūkstošu datu lielas kopas, kuras netika lietotas modeļu trenēšanā vai validācijā – t.i., vispār netika padodas kā ievades dati modeļa veidošanas laikā. Parametru izvēle tiks

veikta ar modeļiem priekš spriegumu prognozēšanas, jo jau sākotnējos testos tika identificēts, ka spriegumu prognozēm sasniegt vajadzīgo precizitāti ir daudz izaicinošāk nekā izlieču prognozēm. Kā aprakstīts nodaļā 6.2, tad modeļu precizitāte tiks vērtēta pēc vidējās relatīvās kļūdas (MAPE)

Sākumā apkoposim rezultātus no visiem trim satvariem. Microsoft risinājums tika testēts ar izstrādātāja ieteikto trenēšanas jaudu. Kā arī jāatzīmē, ka precīzākais no mēģinājumiem tika sasniegts, modelim beidzot darbību būtiski ātrāk.

Tabula 6.1 Microsoft Azure satvara rezultāti pie dažādiem iestatījumiem

Iestatījumi	Zaudējuma funkcija	Vidējā relatīvā kļūda	Trenēšanas ilgums	Trenēšanas jauda	Labākā modeļa tips
N/A	Normalizēta vidējā kvadrātiskā kļūda	32.89%	12 stundas	4x vCPU, 14 GB RAM	“Stack” ansamblis
N/A	Normalizēta vidējā absolūtā kļūda	32.97%	12 stundas		“Light GBM”
N/A	R ²	23.58%	12 stundas		“Stack” ansamblis
N/A	Spearman korelācija	15.32%	52 minūtes*		“Balsošanas ansamblis”
Labākais rezultāts:		15.32%			

* modeļa trenēšana tika palaista, norādot 12 stundas kā maksimālo darbības laiku, bet noslēdza darbību ātrāk.

Satvaram AutoGluon ir trīs pamata iestatījumi ‘best_quality’, ‘high_quality_fast_inference_only_refit’, ‘good_quality_faster_inference_only_refit’. Būtībā šie iestatījumi balansē modeļa precizitāti pret prognožu veikšanas ātrumu. Testam tika izvēlēti divi galējie iestatījumi – “best_quality” un “good_quality_faster_inference_only_refit”. Testi tika veikti ar būtiski mazāku trenēšanas jaudu, nekā ieteiktā. Tika secināts, ka jauda ir atbilstoša zemākās kvalitātes modeļa trenēšanai – starp 10 stundu un 20 stundu ilgu trenēšanu faktiski nebija atšķirības (tīri skaitliski skatoties – 20h deva sliktāku rezultātu). Augstākās kvalitātes modeļa trenēšanā bija būtisks uzlabojums starp 10 stundu un 20 stundu trenēšanas mēģinājumiem un izskatās, ka ir potenciāls sasniegt vēl augstāku precizitāti.

Tabula 6.2 Amazon AutoGluon satvara rezultāti pie dažādiem iestatījumiem

Iestatījumi	Zaudējuma funkcija	Vidējā relatīvā kļūda	Trenēšanas ilgums	Trenēšanas jauda	Labākā modeļa tips
“good_quality_faster_inference_only_refit”	Vidējā kvadrātiskā kļūda	5.40%	10 stundas	4x vCPU, 16 GB RAM	N/A
“good_quality_faster_inference_only_refit”	Vidējā absolūtā kļūda	6.01%	10 stundas		N/A
“good_quality_faster_inference_only_refit”	Vidējā kvadrātiskā kļūda	5.68%	20 stundas		N/A
‘best_quality’	Vidējā kvadrātiskā kļūda	9.63%	10 stundas		N/A
‘best_quality’	Vidējā kvadrātiskā kļūda	4.31%	20 stundas		N/A
Labākais rezultāts:		4.31%			

Satvaram Google AutoML Tables nav atsevišķu “kvalitātes” iestatījumu, iespējamie iestatījumi ir tikai trenēšanas ilgums un zaudējuma funkcija. Jāatzīmē, ka šim satvaram trenēšanas jaudu nevar mainīt un tādēļ jau pašā īsākajā laika posmā, kādu tas atļauj izmantot – 1 stundā – tiek kopā paralēli lietota lielāka jauda nekā mēģinājumos ar iepriekšējiem satvariem. Google vadlīnijas [25] iesaka datiem ar 100 tūkstoši – 1 miljons rindām izmantot trenēšanas laiku 1-6 stundas. Tādēļ tika trenēti arī modeļi arī ar 3h trenēšanas laiku un iegūsti būtiski labāki rezultāti.

Tabula 6.3 Google AutoML Tables satvara rezultāti pie dažādiem iestatījumiem

Iestatījumi	Zaudējuma funkcija	Vidējā relatīvā kļūda	Trenēšanas ilgums	Trenēšanas jauda	Labākā modeļa tips
N/A	Vidējā kvadrātiskā kļūda	10.65%	1 stunda	92 mašīnas ar 4x vCPU, 14 GB RAM	N/A
N/A	Vidējā absolūtā kļūda	4.47%	1 stunda		N/A
N/A	Vidējā kvadrātiskā logaritmiskā kļūda	7.03%	1 stunda		N/A
Labākais rezultāts:		4.47%			
N/A	Vidējā kvadrātiskā kļūda	5.39%	3 stundas		N/A
N/A	Vidējā absolūtā kļūda	2.71%	3 stundas		N/A
N/A	Vidējā kvadrātiskā logaritmiskā kļūda	2.20%	3 stundas		N/A
Labākais rezultāts:		2.20%			

Diskusija par precizitātes vērtējumu. Redzams, ka “Microsoft Azure AutoML” precizitāte būtiski atpaliek no pārējiem diviem satvariem. Kā arī šeit skaidri parādās automatizētās mašīnmācīšanās “ēnas puses” – šajā gadījumā faktiski nav skaidra veida, kā iegūt labāku precizitāti, jo parametri tiek izvēlēti automātiski. Tādēļ turpmāk darbā šis satvars netiks izmantots.

Gan AutoGluon, gan “Google AutoML Tables” uzrāda relatīvi labākus rezultātus. Lai gan augstākais sasniegtais rezultāts ir trenējot modeli 3 stundas ar Google risinājumu, tas nav tieši salīdzināms ar AutoGluon, jo redzams, ka, izvēloties “augstāko kvalitāti”, var sasniegt labu rezultātu ar relatīvi daudz mazāku kopējo aprēķina jaudu. Kā arī pie Google risinājuma labākā zaudējuma funkcija – vidējā kvadrātiskā logaritmiskā kļūda (skatīt arī vienādojumu 6.3) – nav pieejama AutoGluon pēc noklusējuma. AutoGluon ir iespējams papildināt ar citām zaudējuma funkcijām. Tādēļ nevar viennozīmīgi noteikt, kurš no šiem satvariem būtu vērtējams kā precizitātes ziņā pārrāks. Jebkurā gadījumā – labākā precizitāte šajos mēģinājumos tika sasniegta ar “Google AutoML Tables”, trenējot 6 stundas, relatīvā kļūda 2.20%, attiecīgi precizitāte 97.8%.

Tā kā testi tika veikti uz lielas datu kopas – 50 tūkstošiem datu, tad tika novērota cita īpašība – AutoGluon labākās kvalitātes modeļa prognozes bija ļoti lēnas. Tādēļ turpmāk tika apskatīts prognožu ātrums.

6.3.2.2 Ātrdarbības vērtējums

Kā aprakstīts nodaļā 4.2.1, vienas paketes izvērtēšanai nepieciešamas aptuveni 40 – 240 prognozes. Precizitātes ziņā labākie modeļi no AutoGluon un Google AutoML Tables ir salīdzināti pēc to prognožu ātrumiem. Tā kā paredzams, ka vidējais vienas prognozes ātrums modelim var mainīties atkarībā no tā, cik prognozes tiek pieprasītas vienā pieprasījumā, tad tiks mērīti prognožu ātrumi ir pie 1 prognozes, 40 prognozēm, 150 prognozēm un 250 prognozēm vienā pieprasījumā. Katrā no gadījumiem tiks veikti 10 prognožu kopu mērījumi, noteikti vidējie rādītāji. Rezultāti apkopoti Tabula 6.4.

Tabula 6.4 Ātrdarbības salīdzinājums starp AutoGluon "Labākās kvalitātes" modeli un Google AutoML Tables modeli, kas trenēts 3h.

Cik prognožu pieprasījumā?	AutoGluon Vidējais laiks prognožu kopas pieprasījumam	AutoGluon Vidējais laiks vienai prognozei	Google AutoML Vidējais laiks prognožu kopas pieprasījumam	Google AutoML Vidējais laiks vienai prognozei
1 prognoze	21.0s	21000ms	7ms = 0.007s	7ms
40 prognozes	22.1s	552ms	18ms = 0.018s	0.5ms
150 prognozes	26.4s	176ms	129ms = 0.129s	0.86ms
250 prognozes	29.1s	116ms	240ms = 0.240s	1.04ms

Ātrdarbība AutoGluon tika vērtēta lietojot virtuālo mašīnu ar 16GB RAM un 4x vCPU, Google AutoML Tables modeļi tika palaisti uz virtuālās mašīnas ar 2x vCPU un 4GB RAM. Līdz ar to, teorētiski Google risinājumam bija lietots mazāk jaudīgs aprīkojums.

AutoGluon prognozes vienam pilnam paketes aprēķinam (40-240 prognozes) aizņems 22-29 sekundes, kamēr Google AutoML Tables tikai 18-240 milisekundes. Ar šādu AutoGluon prognožu ātrumu stikla paketes optimizācija, un līdz ar to šī darba mērķa sasniegšana, būtu faktiski neiespējama.

Prognožu ātrums ir zināma šī satvara problēma, ir bijuši satvara izstrādātāju ieteikumi, kā modeļi paātrināt - "izņemt" no tā arhitektūras lēnāk strādājošus modeļus [62] (piem., - testētajā modelī ir lietoti arī "tuvāko kaimiņu" modeļi). Tomēr, sākot "rediģēt" modeļa arhitektūru, sāks samazināties arī kopējā modeļa precizitāte. Darba turpinājumā būtu risks nonākt "apburtajā lokā" – vai nu tik lēnas prognozes, kas neļauj izpildīt šī darba mērķi, vai arī ātrākas prognozes ar neprecīziem rezultātiem.

Diskusija. Izvēlētais satvars.

Amazon AutoGluon ir daudzsološs satvars, ar relatīvi mazu aprēķina jaudu tas sasniedza līdzīgus rezultātus kā Google AutoML Tables. Tas ir satvars ar atvērtu kodu, modeļa arhitektūras "rediģēšanas" iespējām, iespējām lietot paša veidotas zaudējuma funkcijas un plašu dokumentāciju. Tomēr šī darba ietvaros tiks izmantots Google AutoML Tables risinājums. Par spīti tā "necaurspīdīgumam" un izmaksām (t.i. nepieciešamo trenēšanas aprēķinu jaudu), tas

sasniedz labākos rezultātus. Gan augstāko precizitāti (šajā nodaļā apskatīto modeļu mazākā kļūda = 2.2%). Gan arī ļoti ātru prognožu laiku, kas pārsniedz AutoGluon sniegumu vairākus simtus reižu.

6.4 Modeļa parametru izvēle

Pēc tam, kad esam izvēlējušies modeļa trenēšanas ievāru, apskatīsim veidus kā varam izveidot pēc iespējas precīzāku modeli. Sākotnējiem testiem ievādes dati ir tādi paši kā mašīnmācīšanās satvara izvēlē – spriegumu prognozēšanas modelis, 500 tūkstoši treniņa datu, 50 tūkstoši testa datu.

6.4.1 Zaudējuma funkcijas izvēle

Google AutoML Tables piedāvā izvēlēties trīs dažādas zaudējuma funkcijas (loss functions):

- vidējo absolūto kļūdu (*MAE* – *mean average error*)

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (6.1)$$

- vidējo kvadrātisko kļūdu (*RMSE* – *root mean square error*)

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (6.2)$$

- vidējo kvadrātisko logaritmisko kļūdu: (*RMSLE* – *root mean square logarithmic error*)

$$RMSLE = \sqrt{\frac{\sum_{i=1}^n (\log(y_i+1) - \log(x_i+1))^2}{n}} \quad (6.3)$$

Vienādojumos 6.1, 6.2 un 6.3 lietoti šādi apzīmējumi:

n – datu skaits

y_i – prognozētās vērtības

x_i – patiesās vērtības

Tika trenēti modeļi ar vienādiem datiem, izvēloties katru no šīm zaudējuma funkcijām. Modeļi tika trenēti 3 stundas. Laiks izvēlēts atbilstoši Google vadlīnijām, kuras modeļus ar 100 tūkstošu – 1 miljonu datu rindām iesaka trenēt 1-6 stundas.

Atbilstoši nodaļā 6.2 aprakstītajiem precizitātes kritērijiem, labākā zaudējuma funkcija būs tā, kura sasniegs mazāko relatīvo/procentuālo kļūdu (*MAPE* – *mean average percentage error*), kā arī tiks sasniegts pēc iespējas vienmērīgāks relatīvo kļūdu sadalījums.

Faktiski šo modeļu trenēšana tika aprakstīta jau sadaļā 6.3.2, tomēr šajā nodaļā šo modeļu rezultāti apskatīti detalizētāk no zaudējuma funkcijas aspekta.

Tabula 6.5 Relatīvo kļūdu salīdzinājums pie dažādām zaudējuma funkcijām. Pilna testa datu kopa.

Zaudējuma funkcija	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Vidējā absolūtā kļūda (MAE)	2.71%	36.50%	299.4%	5.80%
Vidējā kvadrātiskā kļūda (RMSE)	5.39%	73.83%	968.9%	11.92%
Vidējā kvadrātiskā logaritmiskā kļūda (RMSLE)	2.20%	24.52%	154.8%	3.89%

Tabula 6.6 Relatīvo kļūdu salīdzinājums pie dažādām zaudējuma funkcijām. Tikai dati ar spriegumiem 7.2MPa – 88MPa

Zaudējuma funkcija	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Vidējā absolūtā kļūda (MAE)	1.64%	29.65%	165.12%	3.90%
Vidējā kvadrātiskā kļūda (RMSE)	4.14%	49.07%	241.72%	6.81%
Vidējā kvadrātiskā logaritmiskā kļūda (RMSLE)	1.23%	22.65%	114.56%	2.83%

Tabula 6.5 un Tabula 6.6 apkopo datus modeļiem ar dažādām zaudējuma funkcijām. Jau pēc šiem pirmajiem 3 modeļiem redzam nozīmīgas tendences. Pirmkārt, vidējā kvadrātiskā logaritmiskā kļūda kā zaudējuma funkcija iegūst vislabākos relatīvās kļūdas rādītājus – gan pašu kļūdu, gan mazāko standartnovirzi, gan mazāko maksimālo kļūdu. Otrkārt – redzam, ka lielākās relatīvās kļūdas notiek spriegumu reģions, kas ir ārpus mūsu interesējošā diapazona 7.2-88.0 MPa (skat nodaļu 6.2). Treškārt – varam novērtēt, ka, lai gan vidēji varam runāt par augstu precizitāti 98.8% interesējošajā diapazonā, maksimālā kļūda ir virs 100% un 99.7-percentīle ir tālu no vēlamajiem 5%.

Secinājums – modeļu trenēšanā lietosim vidējo kvadrātisko logaritmisko kļūdu kā zaudējuma funkciju.

6.4.2 Biežāk lietoto parametru apgabalu ievērtēšana

Pirmie modeļi tika trenēti ar datiem, kuros visiem ievades parametriem ir vienmērīgais sadalījums (skat. 5.2.3). Tomēr praktiski stikla paketēm šādu gadījumu būs diezgan maz – parasti līnijas/punkta slodze ir pielikta no vienas puses, pārējie paketes slāņi tiek slogoti tikai ar vienmērīgi izkliedētām klimatiskajām un vēja slodzēm. Tādēļ varam uzskatīt, ka puse vai 2/3 (attiecīgi divslāņu un trīsslāņu paketēm) būs gadījumi, kad 3 no 8 ievades parametriem būs vienādi ar 0. Šie trīs parametri ir – punkta slodzes spēks, līnijas slodzes spēks, līnijas slodzes augstums.

Tika apskatīts, kā mainās prognožu precizitāte, atkarībā no slogojuma veida:

Tabula 6.7 Relatīvo kļūdu salīdzinājums pie dažādiem slogojuma veidiem. Treniņu dati tikai ar visām slodzēm pieliktām. Pilna testa datu kopa.

Slogojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Visas slodzes pieliktas	2.20%	24.52%	154.81%	3.89%
Pielika izkliedēta un līnijas slodze	3.29%	28.73%	100.00%	4.49%
Pielika izkliedēta un punkta slodze	3.60%	46.09%	436.57%	7.01%
Pielikta tikai izkliedētā slodze	5.26%	99.11%	343.50%	9.50%

Tabula 6.7 skaidri parāda, ka modeļa precizitāte būtiski krītas, ja viens vai vairāki parametri (slodzes) ir vienādi ar 0. Ja tikai viens parametrs – punkta slodzes vērtība ir 0, tad kļūda palielinās par 50%. Ja trīs parametri vienādi ar 0 un pielikta tikai izkliedēta slodze, tad kļūda palielinās 2.4 reizes un ir 5.26%.

Secinājums – ir būtiski modeli trenēt ar papildu datiem apgabalos, kuri tiks bieži izmantoti. Kā risinājums šim tika lietots šāds treniņa datu sadalījums:

- 50% no datiem ar visām slodzēm pieliktām.
- 25% dati ar pieliktu tikai izkliedētu slodzi. Jāmin, ka šādam gadījumam ir pat izstrādātas empīriskas formulas [53] un ir paredzams, ka šis būs vienkāršs gadījums prognozēšanai.
- 12.5% dati ar pieliktu izkliedētu un līnijas slodzi;
- 12.5% dati ar pieliktu izkliedētu un punkta slodzi.

Trenējot ar šādiem datiem, iegūstam šādas relatīvās kļūdas:

Tabula 6.8 Relatīvo kļūdu salīdzinājums pie dažādiem slogojuma veidiem. Treniņu dati ar ievērtētiem dažādiem slogojuma pielikšanas veidiem. Pilna testa datu kopa.

Slogojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Visas slodzes pieliktas	2.22%	24.70%	108.99%	3.82%
Pielika izkliedēta un līnijas slodze	2.21%	24.31%	73.78%	3.77%
Pielika izkliedēta un punkta slodze	2.67%	31.13%	242.95%	5.13%
Pielikta tikai izkliedētā slodze	2.18%	27.21%	4103.77%	25.11%

Tabula 6.8 parāda, ka, lietojot treniņa datus ar dažādiem slogojuma tipiem, varam būtiski izlabot rezultātus. Redzam, ka tikai gadījums ar izkliedētu un punkta slodzi ir ar sliktāku precizitāti nekā iepriekš apskatītais modelis (Tabula 6.5), kas trenēts un testēts ar visu slodžu datiem.

Tabula 6.9 Relatīvo kļūdu salīdzinājums pie dažādiem slogojuma veidiem. Treniņu dati ar ievērtētiem dažādiem slogojuma pielikšanas veidiem. Tikai dati ar spriegumiem 7.2MPa – 88MPa

Slogojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Salīdzinājumam – rezultāti modelim, kurš gan trenēts, gan testēts tikai ar datiem, kuros pieliktas visas slodzes.	1.23%	22.65%	114.56%	2.83%
Visas slodzes pieliktas	1.27%	22.94%	82.15%	2.82%
Pielika izkliedēta un līnijas slodze	1.18%	23.34%	45.26%	2.73%
Pielika izkliedēta un punkta slodze	1.49%	28.44%	113.16%	3.51%
Pielikta tikai izkliedētā slodze	1.23%	26.11%	58.50%	3.39%
Vidēji starp visiem slodžu tiem	1.29%		Max: 113.16%	

Tabula 6.9 parāda datus tikai būtiskajā spriegumu diapazonā. Tas arī parāda, ka lielās maksimālās kļūdas un attiecīgi standartnovirzes Tabula 6.8 ir saistītas ar izlecējiem vai nu pie ļoti maziem, vai ļoti lieliem spriegumiem.

Tika veikts eksperiments, katram no slogojuma veidiem pieliekot kategorisku mainīgo – tādējādi modeļa trenēšanā skaidri norādot, kādu slodžu gadījumu katra no datu rindām pārstāv. Tabula 6.10 parāda rezultātus. Salīdzinot Tabula 6.9 ar Tabula 6.10 varam redzēt, ka precizitātes ir gandrīz vienādas. Vidējā kļūda starp visiem slodžu tiem mainās no 1.29% uz 1.32%, kas tūri skaitliski nozīmētu, ka šāda kategoriju parādīšana datiem pat pasliktina rezultātus.

Tabula 6.10 Relatīvo kļūdu salīdzinājums pie dažādiem slogojuma veidiem. Treniņu dati ar ievērtētiem dažādiem slogojuma pielikšanas veidiem un katrai rindai norādītu kategoriju (label), kādam slodžu pielikšanas tipam tā pieder. Tikai dati ar spriegumiem 7.2MPa – 88MPa

Slogojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Salīdzinājumam – rezultāti modelim, kurš gan trenēts, gan testēts tikai ar datiem, kuros pieliktas visas slodzes.	1.23%	22.65%	114.56%	2.83%
Visas slodzes pieliktas	1.25%	22.42%	84.53%	2.81%
Pielika izkliedēta un līnijas slodze	1.22%	22.82%	44.12%	2.71%
Pielika izkliedēta un punkta slodze	1.51%	28.27%	113.92%	3.49%
Pielikta tikai izkliedētā slodze	1.28%	26.02%	51.22%	3.39%
Vidēji starp visiem slodžu tiem	1.32%		Max: 113.92%	

Secinājumi – turpmākajā modeļu trenēšanā lietosim datus, kuros 50% ir ar visām pieliktām slodzēm, 25% tikai ar izkliedētu slodzi, pa 12.5% datiem gadījumiem, kad izklaidēta slodze kombinēta ar punkta vai līnijas slodzi. Šīs slodžu kategorijas datus atsevišķi neatzīmēsim.

6.4.3 Trenēšanas ilguma ievērtēšana

Nodaļās 6.4.1 un 6.4.2 veiktie eksperimenti ir veikti ar modeļa trenēšanas ilgumu 3 stundas. Nodaļā 6.3.2.1 salīdzinājām rezultātus modeļiem, kas trenēti 1 stundu, un tiem, kas trenēti 3 stundas. Kļūda samazinājās aptuveni uz pusi (Tabula 6.3). Tādēļ tālāk apskatīsim, vai, palielinot trenēšanas laiku vēl divas reizes, arī kļūda turpinās samazināties tikpat strauji. Tika apskatīts, kādu precizitātes uzlabojumu var iegūt, trenējot modeli 6 stundas.

Tabula 6.11 Relatīvo kļūdu salīdzinājums pie 3 stundu un 6 stundu trenēšanas ilguma. Tikai dati ar spriegumiem 7.2MPa – 88MPa

Slogojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
3h - Visas slodzes pieliktas	1.25%	22.42%	84.53%	2.81%
3h - Pielika izkliedēta un līnijas slodze	1.22%	22.82%	44.12%	2.71%
3h - Pielika izkliedēta un punkta slodze	1.51%	28.27%	113.92%	3.49%
3h - Pielikta tikai izkliedētā slodze	1.28%	26.02%	51.22%	3.39%
Vidējais starp slodžu veidiem – 3h:	1.31%		Maks: 113.92%	
6h – Visas slodzes pieliktas	1.37%	27.76%	106.79%	3.30%
6h - Pielika izkliedēta un līnijas slodze	1.07%	22.14%	42.50%	2.57%
6h - Pielika izkliedēta un punkta slodze	1.37%	27.76%	106.79%	3.30%
6h - Pielikta tikai izkliedētā slodze	1.09%	25.79%	57.85%	3.21%
Vidējais starp slodžu veidiem – 6h:	1.22%		Maks: 106.79%	

Tabula 6.11 parāda, ka pie 6 stundām treniņa laika uzlabojums ir neliels. Vidējā relatīvā kļūda starp visiem slogojumu veidiem samazinās no 1.31% uz 1.22%. Līdzīgi vērojami nelieli samazinājumi arī 99.7-percentīlēm, maksimālajām kļūdām un standartnovirzēm. Tieši gadījumam ar visām slodzēm pieliktām, ilgāks treniņa laiks pat parāda nedaudz sliktākus rezultātus, uzlabojumi ir pārējos slogojuma veidos.

Jāpiebilst, ka šajā salīdzinājumā nevaram skaidri komentēt, kurā brīdī varētu sākties modeļa pārmērīga pielāgošana (*overfitting*). Tas tādēļ, ka Google dokumentācija nav caurspīdīga attiecībā uz to, kā tā izvēlās katra atsevišķa modeļa trenēšanas ilgumu. Google sagatavotā dokumentācija apgalvo, ka modeļa trenēšana tiks apturēta, tiklīdz kā tas beigs uzlabot savu precizitāti [25]. Gan 3 stundu, gan 6 stundu trenēšanas gadījumā modeļa trenēšana pārtraukta netika. Maksimālais trenēšanas laiks netika izmēģināts Google pakalpojuma augsto izmaksu dēļ.

Secinājums – modeļa trenēšana lietojot maksimālo Google ieteikto laiku šāda izmēra datu kopai (6 stundas), sasniedz nelielu precizitātes palielinājumu (par 0.09%).

6.5 Modelis maksimālo spriegumu noteikšanai

Pēc modeļa parametru izvēles, kā aprakstīts nodaļā 6.4, tika trenēts modelis ar visiem pieejamajiem datiem. Šie dati ir analizēti nodaļā 5.2. Datu kopas izmēru ar 2 miljoniem datu praktiski noteica šī maģistra darba nodošanas termiņš. Praktiskam programmatūras pielietojumam, datu kopa tiek turpināta ģenerēt, un nākotnē tiks trenēti jauni modeļi ar lielāku datu skaitu.

Modelis tika trenēts 15 stundas, 92 virtuālajām mašīnām darbojoties paralēli, katras mašīnas parametri 4x vCPU, 15GB operatīvā atmiņa un SSD disks [63]. 1-10 miljonu rindu lielām datu kopām Google vadlīnijas [25] iesaka trenēšanas laiku 1-12 stundas. Eksperiments nodaļā 6.4.3 norādīja, ka pēc puses no Google ieteiktā laika izmantošanas var iegūt tikai ļoti nelielu uzlabojumu. Tomēr šajā gadījumā mērķis bija iegūt pēc iespējas labāku precizitāti, tādēļ arī salīdzinoši lielāks trenēšanas laiks.

6.5.1 Modeļa uzbūve

Pēc “Google AutoML Tables” vērtējuma, kurā 1/10 no datiem tika atstāti testēšanai, trenētais modelis sasniedz šādas vērtības:

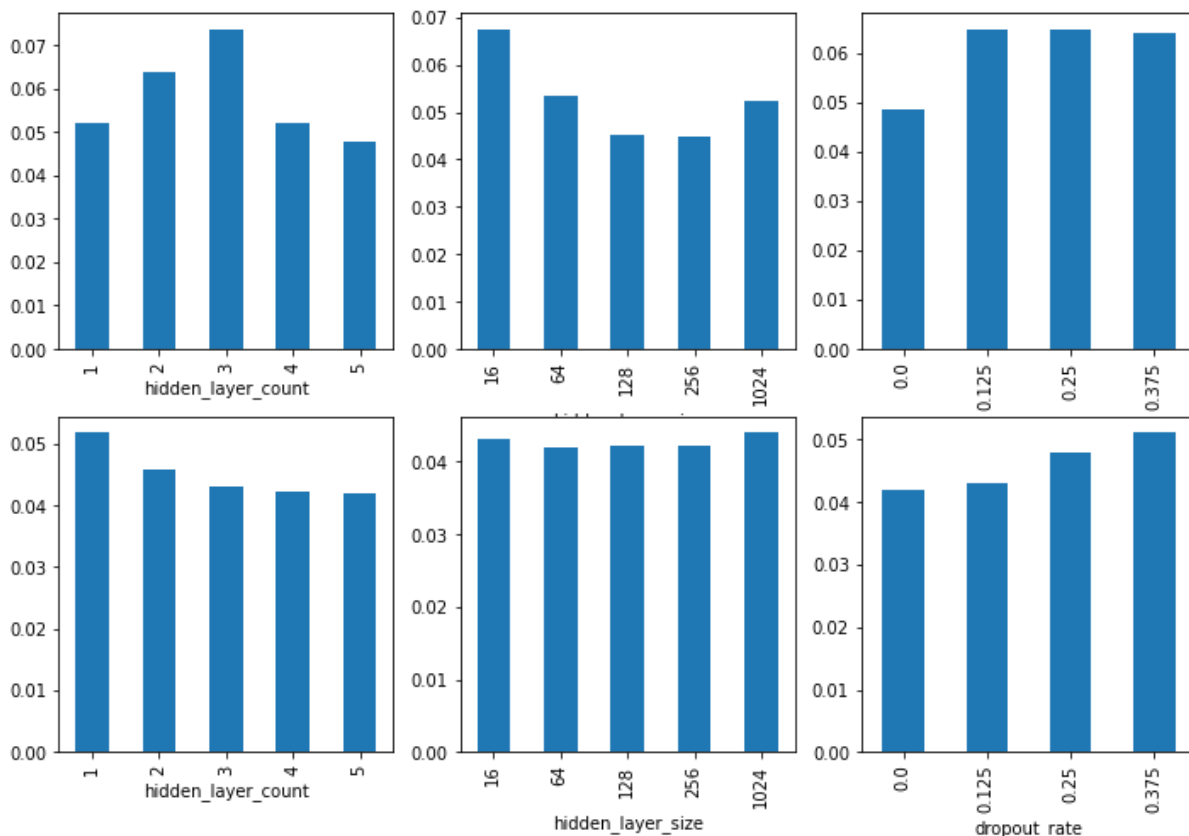
- Vidējā absolūtā kļūda (MAE) = 3.93MPa
- $R^2 = 0.994$
- Vidējā relatīvā kļūda (MAPE) = 1.75%. Uz lietotajiem testa datiem relatīvā kļūda tika aprēķināta kā 1.67% (Tabula 6.13). Šāda neliela atšķirība varētu tikt radusies konkrēto testa datu kopu atšķirību dēļ.
- Vidējā kvadrātiskā logaritmiskā kļūda (RMSLE) = 0.040. Šis parametrs tika lietots kā zaudējuma funkcija treniņu procesā un tādēļ ir ļoti būtisks.

Lai nonāktu pie šī gala modeļa, kopā tika trenēti 300 modeļu. Četri no šiem modeļiem bija gradientu pastiprināti lēmumu koki (*GBDT - gradient boosted decision trees*), pārējie 296 – “*feed-forward*” neironu tīklu modeļi. GBDT modeļu labākā precizitāte zaudējuma funkcijai $RMSLE = 2.48$, kas ir būtiski sliktāka nekā neironu tīkliem. Gala modelī neviens no šiem GBDT modeļiem nav ticis iekļauts, tādēļ tālāk šie modeļi nav apskatīti.

No šiem 296 neironu tīklu modeļiem, 25 tika automātiski iekļauti gala modelī, kur “Google AutoML Tables” atsevišķi trenēja ansambli ar šiem modeļiem. Jāpiebilst, ka labākais atsevišķais neironu tīklu modelis sasniedza $RMSLE = 0.042$, bet ansamblis 0.04. Samazinājums-uzlabojums par 5% šajā gadījumā ir neliels, bet ir. Jāpiebilst, ka Google piedāvātā informācija par trenētajiem modeļiem nav pilnīga – no 296 modeļiem, tikai 133 bija

ar unikālām pieejamo parametru kopām, par šo nav skaidra komentāra dokumentācijā. Kā aprakstīts 6.3.1 un Att. 6.3, tad varētu minēt, ka ne-unikālie modeļi bija radušies no krossvalidācijas un “bootstrapping”.

Turpmāk apskatīsim, kā dažādi neironu tīkla parametri ir ietekmējuši RMSLE. Jāatgādina, ka mazāka RMSLE vērtība ir labāks rādītājs.



Att. 6.4 Dažādu parametru ietekme uz spriegumu modeļa precizitāti. Augšējā rindā grafiki ar vidējām RMSLE vērtībām, apakšējā rindā - ar mazākajām sasniegtajām. Pa kreisi - slēpto slāņu skaits, pa vidu - slēpto slāņu izmērs, pa labi - lietotā "dropout" lielums

No Att. 6.4 varam redzēt, ka “Google AutoML Tables” ir apskatījis 1-5 slēptos slāņus, un labākie rezultāti sasniegti ar lielāku slāņu skaitu. Palielinot slāņu platumu, iegūstam labākas vidējās vērtības, bet labākās sasniegtās vērtības gandrīz nemainās. Jāpiemin, ka no 296 modeļiem, tikai 10 bija ar slāņu platumu 1024, līdz ar to par šo slāņa izmēru skaidru secinājumu nevar izdarīt.

Pretēji cerētajam, atmešanas (“dropout”) koeficientu palielinot, modeļa precizitāte samazinās.

Kā minēts, galīgais trenētais modelis saturēja ansambli no 25 iepriekš trenētiem modeļiem. Kritēriji, kā Google AutoML algoritms izvēlējies šos modeļus, ir neskaidri. Skatoties no iepriekš trenētajiem modeļiem, netiek izvēlēti 25 labākie modeļi – tiek izvēlēti modeļi, kuru parametri ranga ziņā bija 1,4,5,7,8 u.tml. Šiem modeļiem, no pieejamās informācijas, ir 4 dažādi uzbūves varianti:

Tabula 6.12 Spriegumu noteikšanas modeļu ansablī lietoto modeļu parametri

Nr.	Skaitis, cik reīzu tīkls lietots ansablī	Katra slāņa izmērs	Slāņu skaits	Skaitlisko ievades datu normalizācija	Šķērs-slāņi (cross layers)	Slāņu savienojumi “pa taisno” (skip connections)	Atkritēju koeficients (dropout)
1	5	64	5	Jā	1	Jā	0
2	5	128	4	Jā	2	Jā	0
3	10	128	5	Jā	0	Jā	0
4	5	128	5	Jā	1	Jā	0

Izveidotais modeļu ansablis lietotājam tiek nodots kā “Tensorflow” saglabāts modelis, kuru attiecīgi iespējams darbināt, servējot prognožu pieprasījumu caur HTTP.

Secinājums – par spīti tam, ka no “Google AutoML Tables” ir iespējams iegūt vispārīgus datus par modeļiem, dotā informācija ir nepietiekama, lai izdarītu skaidrus secinājumus par modeļu uzbūvi. Šo modeļu ansabli ir iespējams atvērt un pētīt lietojot rīku “Tensorboard”. Tas ir iespējams turpmāks pētījumu virziens, bet nav veikts šajā darbā, jo tiešā veidā nevirzās uz šī darba mērķi.

6.5.2 Modeļa precizitāte

Tabula 6.13 un Tabula 6.14 apkopo modeļu relatīvās kļūdas - visiem datiem un tikai datiem, kuru īstās vai prognozētās spriegumu vērtības atrodas normatīvu pārbaudēm nozīmīgajā diapazonā. Modeļa ar 2 miljoniem datu testēšanā tika lietotas 100 tūkstošus lielas datu kopas katram no sloģojuma veidiem – attiecīgi 2 reizes lielākas, kā iepriekšējo eksperimentu testiem. Šie, kopumā 400 tūkstoši testa datu nav tikuši lietoti nevienā no modeļu treniņu procesiem.

Salīdzinot ar eksperimentu modeli ar 500 tūkstošiem datu (Tabula 6.15), redzam, ka būtiski samazinās vidējā kļūda – no 1.28% uz 0.88%, tātad par 31%. Savukārt 99.7-percentīle samazinās nedaudz līdz 25%. Maksimālā datu kopā sastaptā kļūda palielinās no 113% uz 132%. Skaidrojums tam varētu būt divas reizes lielākā testa kopa, kur tika sastapta kāda “neērta” datu kombinācija.

Tabula 6.13 Relatīvās kļūdas spriegumu prognozēm, modelim ar 2 miljonu treniņu datu kopu. Visi dati.

Sloojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Visas slodzes pieliktas	1.74%	22.62%	73.54%	3.45%
Pielika izkļiedēta un līnijas slodze	1.97%	25.02%	132.37%	3.85%
Pielika izkļiedēta un punkta slodze	1.68%	23.23%	54.91%	3.48%
Pielikta tikai izkļiedētā slodze	1.38%	22.14%	86.75%	3.13%
Vidējās-svērtās vērtības, ievērtējot treniņu datu kopas sadalījumu 50/12.5/12.5/25 starp sloojuma tiem	1.67%	Max: 25.02%	Max: 132.37%	3.42%

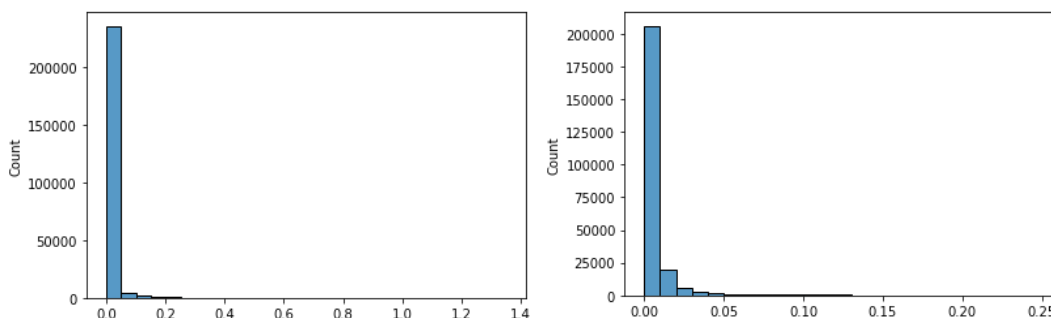
Tabula 6.14 Relatīvās kļūdas spriegumu prognozēm, modelim ar 2 miljonu treniņu datu kopu. Tikai dati ar spriegumiem 7.2MPa – 88MPa

Sloojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Visas slodzes pieliktas	0.88%	20.29%	73.54%	2.29%
Pielika izkļiedēta un līnijas slodze	1.05%	24.96%	132.37%	2.90%
Pielika izkļiedēta un punkta slodze	0.84%	20.67%	54.91%	2.32%
Pielikta tikai izkļiedētā slodze	0.84%	22.38%	86.75%	2.72%
Vidējās-svērtās vērtības, ievērtējot treniņu datu kopas sadalījumu 50/12.5/12.5/25 starp sloojuma tiem	0.88%	Max: 24.96%	Max: 132.37%	2.48%

Tabula 6.15 Relatīvās kļūdas spriegumu prognozēm, salīdzinājums starp modeļiem ar dažādu treniņu datu skaitu. Tikai dati ar spriegumiem 7.2MPa – 88MPa

Sloojuma veids	Vidējā relatīvā kļūda (MAPE) – 500 tūkstoši treniņu datu	Vidējā relatīvā kļūda (MAPE) – 2 miljoni treniņu datu	Relatīvās kļūdas 99.7-percentīle – 500 tūkstoši treniņu datu	Relatīvās kļūdas 99.7-percentīle – 2 miljoni treniņu datu
Visas slodzes pieliktas	1.27%	0.88%	22.94%	20.29%
Pielika izkļiedēta un līnijas slodze	1.18%	1.05%	23.34%	24.96%
Pielika izkļiedēta un punkta slodze	1.49%	0.84%	28.44%	20.67%
Pielikta tikai izkļiedētā slodze	1.23%	0.84%	26.11%	22.38%
Vidējās-svērtās vērtības, ievērtējot treniņu datu kopas sadalījumu 50/12.5/12.5/25 starp sloojuma tiem	1.28%	0.88%	Max: 28.44%	Max: 24.96%

Apskatot histogrammas relatīvajām kļūdām Att. 6.5, redzam jau skaitliski redzamo tendenci – lielākā daļa datu ir ar kļūdu <5%, bet ir atsevišķi izlecēji ar lielām kļūdām.



Att. 6.5 Histogramma relatīvo kļūdu vērtībām spriegumu prognozēšanas modelī. Visām kļūdu vērtībām (pa kreisi) un vērtībām līdz 25% (pa labi)

Modeļa vērtējums.

Vidējā precizitāte spriegumu prognozēm diapazonā, kurā notiek stikla paketes pārbaudes atbilstoši normatīviem, ir 99.1%. Visā testa datu diapazonā tā ir 98.3%.

Vērtējot modeli pēc nodaļā 6.2 uzstādītajiem kritērijiem:

- Maksimālā kļūda 99.7% no datiem nozīmīgajā diapazonā ir 25%, nevis 5%, kā noteikts kritērijā. Kļūda zem 5% ir 96% datu, kas kopumā vērtējams kā rezultāts tuvs vēlamajam.
- Maksimālā kļūda datos nozīmīgajā diapazonā ir 132%, kas ir būtiski lielāka par uzstādīto kritēriju 35%. Kļūda lielāka par 35% bija 0.034% no testētajiem datiem.

6.6 Modelis maksimālo deformāciju noteikšanai

Maksimālo deformāciju prognozēšanas modelim šajā darbā pievērsta būtiski mazāka uzmanība. Tas tādēļ, ka deformāciju prognozes pie mēģinātajām datu kopām uzrādīja būtiski labākus rezultātus. Gan ievades dati, gan lietotie modeļa trenēšanas parametri deformāciju prognožu modelim ir lietoti tādi paši kā spriegumu prognozēšanas modelim.

6.6.1 Modeļa uzbūve

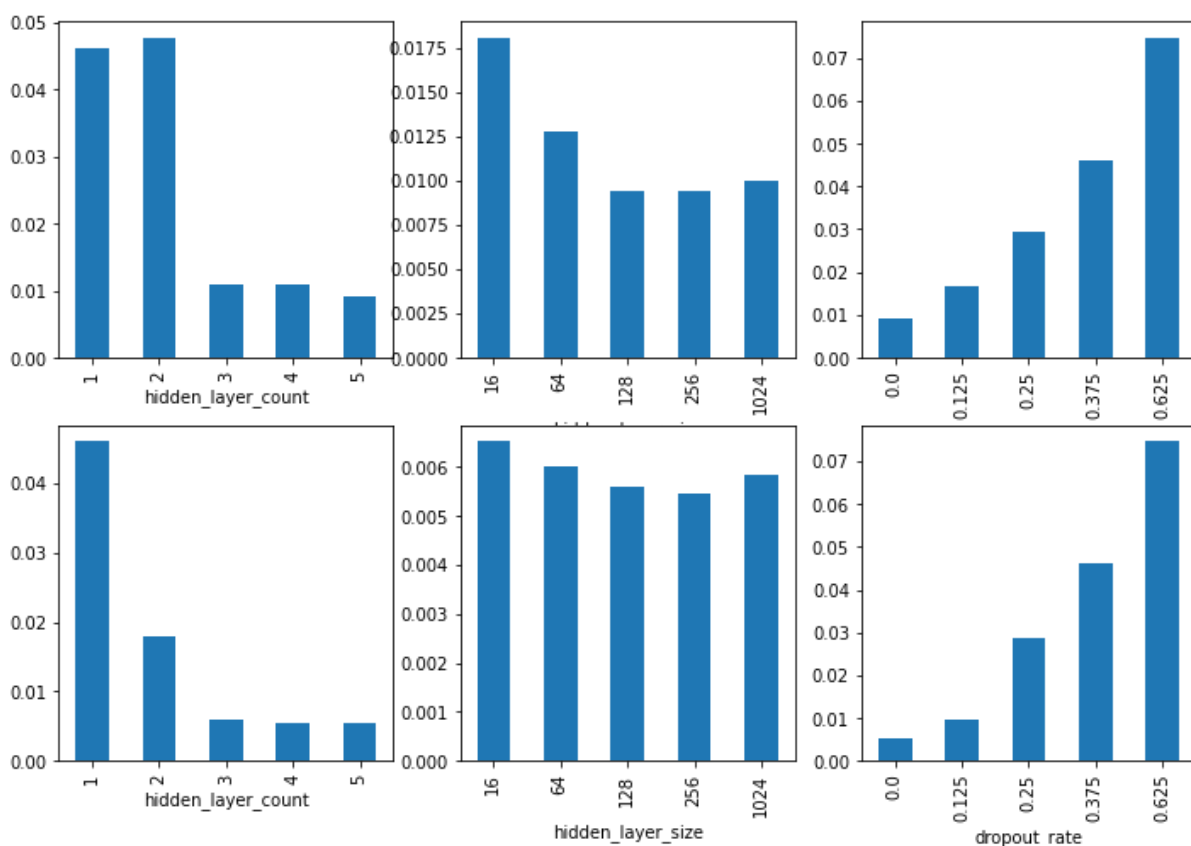
Trenētais modelis sasniedz šādas vērtības (dati, ko dod “Google AutoML Tables” satvars pēc modeļa trenēšanas):

- Vidējā absolūtā kļūda (MAE) = 0.039 mm
- $R^2 = 1.000$
- Vidējā relatīvā kļūda (MAPE) = 0.23%
- Vidējā kvadrātiskā logaritmiskā kļūda (RMSLE) = 0.003.

Tāpat kā spriegumu prognozēšanai, ar deformāciju prognozēm “Google AutoML Tables” sākumā trenēja 300 atsevišķu modeļu un pēc tam izvēlējās 25, no kuriem automātiski veidot galīgo modeli – modeļu ansambli. Visi modeļi bija vienvirziena plūsmas (*feed-forward*) neironu tīklu modeļi.

Atšķirībā no spriegumu modeļa trenēšanas, deformācijām ir būtiska starpība starp labāko atsevišķo modeli ar RMSLE=0.0055 un modeļu ansambli, kurš sasniedz RMSLE=0.003, 54% samazinājums.

Deformāciju modelim izteiktāk var redzēt, kā dažādi parametri ietekmē modeļa precizitāti.



Att. 6.6 Dažādu parametru ietekme uz deformāciju modeļa precizitāti. Augšējā rindā grafiki ar vidējām RMSLE vērtībām, apakšējā rindā - ar mazākajām sasniegtajām. Pa kreisi - slēpto slāņu skaits, pa vidu - slēpto slāņu izmērs, pa labi - lietotā "dropout" lielums

Att. 6.4 varam redzēt, ka, palielinoties slāņu skaitam, palielinās precizitāte. Līdzīgi, kā spriegumu modelim, viena slāņa platums sniedz uzlabojumus līdz 256 neironiem un lielāks atmešanas (“dropout”) koeficients samazina modeļa precizitāti.

Ansablī, pēc pieejamās informācijas, ir četrus dažādu tipu modeļi:

Tabula 6.16 Deformāciju noteikšanas modeļu ansablī lietoto modeļu parametri

Nr.	Skaitis, cik reizi tīkls lietots ansablī	Katra slāņa izmērs	Slāņu skaits	Skaitlisko ievades datu normalizācija	Šķērs-slāņi (cross layers)	Slāņu savienojumi "pa taisno" (skip connections)	Atkritēju koeficients (dropout)
1	10	256	5	Jā	0	Jā	0
2	5	256	4	Jā	2	Jā	0
3	5	128	4	Jā	1	Jā	0
4	5	256	5	Jā	1	Jā	0

Redzams, ka deformāciju prognozēšanai tiek lietoti platāki slāņi – ar 128 vai 256 neironiem (spriegumu modelī 64 vai 128). Arī šis potenciāli ir iemesls, kādēļ modelis dod precīzākus rezultātus.

Secinājumi ir līdzīgi kā par spriegumu modeli, skatīt nodaļas 6.5.1 noslēgumu.

6.6.2 Modeļa precizitāte

Deformāciju prognožu relatīvās kļūdas apkopo Tabula 6.13 un Tabula 6.14. Deformāciju modeļa trenēšanai un testēšanai tika lietotas tās pašas datu kopas, kas spriegumu modelim. Skatīt ieejas datu kopas analīzi nodaļā 5.2.

Līdzīgi kā spriegumu modelim, arī izliecēm tika atsevišķi vērtēts apgabals, kurā atrodas normatīvos noteiktie deformāciju limiti, tas ir no 4mm līdz 47mm.

Tabula 6.17 Relatīvās kļūdas deformāciju prognozēm, modelim ar 2 miljonu treniņu datu kopu. Visi dati.

Slojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Visas slodzes pieliktas	0.28%	6.05%	84.64%	0.83%
Pielika izkļiedēta un līnijas slodze	0.22%	4.41%	47.76%	0.58%
Pielika izkļiedēta un punkta slodze	0.32%	7.72%	218.77%	1.57%
Pielikta tikai izkļiedētā slodze	0.11%	1.64%	11.54%	0.21%
Vidējās-svērtās vērtības, ievērtējot treniņu datu kopas sadalījumu 50/12.5/12.5/25 starp slojuma tiem	0.24%	Max: 7.72%	Max: 218.77%	0.74%

Tabula 6.18 Relatīvās kļūdas deformāciju prognozēm, modelim ar 2 miljonu treniņu datu kopu. Tikai dati ar izliecēm diapazonā 4-47mm

Sloojuma veids	Vidējā relatīvā kļūda (MAPE)	Relatīvās kļūdas 99.7-percentīle	Maksimālā relatīvā kļūda	Relatīvās kļūdas standartnovirze
Visas slodzes pieliktas	0.18%	3.32%	21.48%	0.41%
Pielika izkļiedēta un līnijas slodze	0.14%	2.32%	13.44%	0.30%
Pielika izkļiedēta un punkta slodze	0.18%	3.13%	23.08%	0.43%
Pielikta tikai izkļiedētā slodze	0.08%	0.68%	3.20%	0.10%
Vidējās-svērtās vērtības, ievērtējot treniņu datu kopas sadalījumu 50/12.5/12.5/25 starp sloojuma tipiem	0.15%	Max: 3.32%	Max: 23.08%	0.32%

Modeļa vērtējums.

Vidējā precizitāte spriegumu prognozēm diapazonā, kurā notiek stikla paketes pārbaudes atbilstoši normatīviem, ir 99.85%. Visā testa datu diapazonā tā ir 99.7%.

Vērtējot modeli pēc nodaļā 6.2 uzstādītajiem kritērijiem:

- Maksimālā kļūda 99.7% no datiem nozīmīgajā diapazonā ir 3.3%, mazāka par noteikto kritēriju 5%. 99.8% no mēģinājumiem kļūda ir mazāka par 5%.
- Maksimālā kļūda datos nozīmīgajā diapazonā ir 23%, kas ir mazāka par uzstādīto kritēriju 35%.

6.7 Ātrdarbības salīdzinājums ar GEM aprēķiniem

Mašīnmācīšanās un galīgo elementu modeļu ātrdarbība ir apskatīta no diviem aspektiem – gan salīdzinot nepieciešamo laiku, lai iegūtu vienu aprēķinu/prognozi, gan arī salīdzinot laiku priekš pilnas stikla paketes izvērtēšanas.

Šajā nodaļā veiktajos ātrdarbības salīdzinājumos ir izvēlēti pēc iespējas līdzīgi datoru parametri. Galīgo elementu aprēķini tiek veikti uz darbstacijas ar Intel Core i5-2520M 2-kodolu procesoru un 4GB operatīvās atmiņas. Mašīnmācīšanās prognozes tiek veiktas uz virtuālās mašīnas ar 2x virtuālajiem CPU un 4GB operatīvās atmiņas.

Mašīnmācīšanās modeļa prognozēm laiks tika novērots uz servera – netika ņemts vērā HTTP pieprasījuma nosūtīšanas/saņemšanas laiks. Gluži kā visi iepriekšējie galīgo elementu modeļi, arī ātruma testi notika lietojot programmatūru Dlubal RFEM 5.19. Galīgu elementu modelī tika ņemts laikā tikai aprēķinu laiks – netika ņemts vērā modeļa ģeometrijas definēšanas un tīkla (*mesh*) ģenerēšanas laiks.

6.7.1 Nepieciešamais laiks vienam aprēķinam un prognozei

Tika veikti 1000 aprēķinu mēģinājumiem ar pieliktām visām no slodzēm. Tabula 6.19 parāda, ka vidējais aprēķina laiks deformāciju prognozēšanai ir 123 reizes mazāks par GEM aprēķinu laiku. Spriegumu prognozēm tas ir 368 reizes mazāks. Redzams arī, ka deformāciju prognozēšana aizņem aptuveni 3x vairāk laika nekā spriegumu. Tas bija sagaidāms rezultāts, jo deformāciju modelī neironu tīklu platums ir lielāks (skatīt Tabula 6.12 un Tabula 6.16).

Tabula 6.19 Nepieciešamais aprēķina laiks atsevišķi aprēķinot/prognozējot katru mēģinājumu individuāli – dati no 1000 nejaušiem mēģinājumiem

Modelis	Vidējais aprēķina laiks	Centrālais (median) aprēķina laiks	Maksimālais aprēķina laiks	Minimālais aprēķina laiks
Galīgo elementu modelis	2944ms	2766ms	4247ms	2590ms
Modelis spriegumu prognozēšanai	8ms	7ms	16ms	6ms
Modelis deformāciju prognozēšanai	24ms	20ms	87ms	12ms

Tabula 6.20 Nepieciešamais aprēķina laiks, prognozējot 150x mēģinājumus katrā pieprasījumā – dati no 75000 nejaušiem mēģinājumiem

Modelis	Vidējais aprēķina laiks	Centrālais (median) aprēķina laiks	Maksimālais aprēķina laiks	Minimālais aprēķina laiks
Modelis spriegumu prognozēšanai 1x prognoze vienā pieprasījumā	8ms	7ms	16ms	6ms
Modelis spriegumu prognozēšanai 150x prognozes vienā pieprasījumā	50ms	48ms	116ms	37ms
Modelis spriegumu prognozēšanai 150x prognozes vienā pieprasījumā vidējais laiks 1 prognozei	0.33ms	0.32ms	0.77ms	0.25ms
Modelis deformāciju prognozēšanai 1x prognoze vienā pieprasījumā	24ms	20ms	87ms	12ms
Modelis deformāciju prognozēšanai 150x prognozes vienā pieprasījumā	141ms	125ms	373ms	80ms
Modelis deformāciju prognozēšanai 150x prognozes vienā pieprasījumā vidējais laiks 1 prognozei	0.94ms	0.83ms	2.49ms	0.54ms

Neironu tīklu modelim, kas tiek darbināts caur Google piedāvāto modeļu servēšanas konteineru, ir iespējams vienā reizē padot sarakstu ar prognožu pieprasījumiem. Kā norādīts nodaļā 4.2.1, viena paketes pilnam izvērtējumam būs nepieciešamas ~40-240 prognozes. Tādēļ tālāk tika apskatīts, kāds būs vienas prognozes laiks, ja vienā pieprasījumā tiek padoti dati 150x prognozēm. Tika veikti 500 pieprasījumu ar 150x prognozēm katrā. Tabula 6.20 parāda, ka aprēķina laiks priekš viena mēģinājuma prognozes samazinās ~25 reizes. Tātad ir būtiski programmatūrā apkopot vairākus mēģinājumus vienā prognožu pieprasījumā.

6.7.2 *Nepieciešamais laiks vienas stikla paketes pilnam novērtējumam*

Līdz šim šajā darbā mašīnmācīšanās modeļus esam vērtējuši, balstoties uz to atsevišķām prognozēm. Šajā nodaļā tiks apskatīti 4 stikla pakešu piemēri un to pilns aprēķinu/prognožu laiks. Praktisko mašīnmācīšanās modeļu paātrinājumu salīdzinājumā ar GEM modeļiem var noteikt dažādi parametri:

- Galīgo elementu skaits GEM modelī. Mašīnmācīšanās modeļa precizitāti nav atkarīga no paketes izmēriem un tā, cik “smalki” ģeometrija sadalīta galīgajos elementos.
- Paketes slāņu skaits – vienā GEM modelī tiek veikti aprēķini vairākām stikla loksniem, savukārt mašīnmācīšanās modelī katrai loksnei nepieciešama atsevišķa prognoze.
- Kombināciju skaits. Kā parādīts nodaļā 6.7.1, vidējais vienas prognozes ilgums būtiski samazinās, ja tiek pieprasītas reizē daudzas prognozes (kas būtu līdzīgi daudzām kombinācijām). Galīgo elementu aprēķinos nav sagaidāms vairāk-kārtīgs uzlabojums.

Kā aprakstīts nodaļā 4.3.3, galīgo elementu racionāls izmērs ir 1/10 no mazākās dimensijas izmēra. Tas nozīmē, ka šaurām un augstām stikla paketēm būs relatīvi vairāk galīgo elementu.

Ņemot vērā iepriekš identificētās potenciālās atšķirības, salīdzinājumam tika izvēlētas šādas stikla paketes ar atbilstošu slodžu kombināciju skaitu:

- A. 1150mm plata, 3500mm augsta stikla pakete, 2-slāņi, 20 slodžu kombinācijas;
- B. 1150mm plata, 3500mm augsta stikla pakete, 2-slāņi, 88 slodžu kombinācijas;
- C. 550mm plata, 3500mm augsta stikla pakete, 3-slāņi, 20 slodžu kombinācijas;
- D. 550mm plata, 3500mm augsta stikla pakete, 3-slāņi, 88 slodžu kombinācijas;

Tabula 6.21 - Īpašības izvēlētajām stikla paketēm ātrdarbības testos

	Liels kombināciju skaits	Mazs kombināciju skaits
Liels galīgā elementu modeļa izmērs	D	C
Mazs galīgo elementu modeļa izmērs	B	A

Visām izvēlētajām paketēm ir pieliktas vienādas slodzes. Visām paketēm ārējais un iekšējais stikls ir lamināti. Tie tiek apskatīti ar lamināta plēvi, vai nu pilnībā stingu, vai bez stinguma (pieeja normatīvā DIN 18008 [44]). Nemainīgie ievades parametri stikla paketēm:

Tabula 6.22 Stikla pakešu pilna novērtējuma ātrdarbības salīdzināšanā lietotie pakešu parametri

Parametrs	Vērtība	Komentāri
Platums	550mm vai 1150mm	Gadījumiem C un D = 550mm, A un B = 1150mm
Augstums	8ms	
Lamināta plēves stingums	Pilns / Bez stinguma	Visām stikla paketēm apskatīti abi gadījumi
Ārējais stikls	6mm / 0.76mm plēve/ 6mm	Laminēts stikls
Vidējais stikls	4mm monolīts stikls	Tikai 3-slāņu paketēm, gadījumi C un D
Iekšējais stikls	5mm / 0.76mm plēve/ 5mm	
Vēja spiediens uz ārējo slāni	0.78 kPa	
Vēja spiediens uz iekšējo slāni	0.26 kPa	
Vēja sūce uz ārējo slāni	-1.04 kPa	
Vēja sūce uz iekšējo slāni	-0.17 kPa	
Punkta slodze	Nav pielikta	
Līnijas slodzes vērtība	1.1 kN/m	
Līnijas slodzes augstums	1100mm	
Līnijas slodzes pielikšanas puse	Uz iekšējo stikla slāni	
Temperatūras starpība vasarā	38°C	
Meteoroloģiskais spied. vasarā	2 kPa	
Temperatūras starpība ziemā	-25°C	
Meteoroloģiskais spied. ziemā	-4 kPa	
Augstuma starpība	+35m	Starpība starp būves atrašanās vietu un ražotnes vietu
Galīgo elementu tīkla izmērs	55mm vai 1150mm	1/10 no mazākās stikla paketes dimensijas

Tabula 6.23 GEM aprēķinu ātruma un rezultātu salīdzinājums ar mašīnmācīšanās modeļu rezultātiem

	A 1150x3500mm, 2-slāņi, 20 komb.	B 1150x3500mm, 2-slāņi, 88 komb.	C 550x3500mm, 3-slāņi, 20 komb.	D 550x3500mm, 3-slāņi, 88 komb.
GEM Aprēķinu ātrums	12 300ms	45 500ms	20 900ms	80 600ms
Maksimālais spriegums starp visiem slāņiem un kombinācijām GEM aprēķinā	17.19 MPa	21.25 MPa	31.33 MPa	32.07 MPa
Maksimālā deformācija starp visiem slāņiem un kombinācijām GEM aprēķinā	7.45 mm	10.36 mm	3.21 mm	3.43 mm
Deformāciju mašīnmācīšanās modelis - ātrums	41ms	271ms	98ms	320ms
Deformāciju mašīnmācīšanās modelis – prognozētā deformācija	7.29 mm	10.14 mm	3.21 mm	3.41 mm
Spriegumu mašīnmācīšanās modelis - ātrums	25ms	167ms	71ms	178ms
Spriegumu mašīnmācīšanās modelis – prognozētā deformācija	17.21 MPa	21.28 MPa	31.35 MPa	32.01 MPa
Atšķirība starp <u>deformācijām</u> GEM un mašīnmācīšanās modeļos	2.1%	2.1%	0.0%	0.6%
Atšķirība starp <u>spriegumiem</u> GEM un mašīnmācīšanās modeļos	0.1%	0.1%	0.1%	0.2%
Cik reizi <u>deformāciju</u> mašīnmācīšanās modelis ir ātrāks par GEM aprēķinu?	300	168	213	252
Cik reizi <u>spriegumu</u> mašīnmācīšanās modelis ir ātrāks par GEM aprēķinu?	492	272	294	453

Tabula 6.23 satur informāciju par GEM aprēķinu ātrumu, rezultātiem un atbilstošām vērtībām mašīnmācīšanās modeļos. No tabulas datiem varam izdarīt vairākus secinājumus:

- Pilns stikla paketes izvērtējums ar mašīnmācīšanās modeļiem aizņēma 168-492 reizes mazāk nekā galīgo elementu aprēķins. Tas ir līdzīgā diapazonā, kā atšķirības atsevišķu prognožu/aprēķinu veikšanai.
- Tā kā GEM modelis kā rezultātus atgriež gan spriegumus, gan deformācijas, bet mašīnmācīšanās modeļus var darbināt paralēli, tad faktiski ātruma vērtējumā ir jāņem lēnākais no spriegumu/deformāciju modeļiem. Kā noskaidrots iepriekš, tas būs deformāciju prognozēšanas modelis. Tādēļ varam teikt, ka praktiski ātruma uzlabojums būs 168-300 reizes.
- Atšķirība spriegumu un deformāciju maksimālajām vērtībām apskatītajām stikla paketēm starp GEM un mašīnmācīšanās modeļiem bija Max. 2.1%, kas ir labs rādītājs, mazāks par 5% (kritērijs skat. 6.2). Tas nav pārsteidzoši, jo zinām, ka 96% no spriegumu prognozēm un >99.7% no deformāciju prognozēm būs ar atšķirību mazāku par 5%.
- Pie kombināciju skaita palielināšanas no 20 uz 88 (4.4 reizes), galīgo elementu modeļu aprēķina laiks pieauga gandrīz proporcionāli – 3.7x starp A un B, 3.8x starp C un D. Mašīnmācīšanās modeļiem ātruma izmaiņas variēja – no 2.5 līdz 6.7 reizēm.

7. STIKLA PAKETES PARAMETRU OPTIMIZĀCIJA

7.1 Vispārīgi

Darba ievadā aprakstīti piedāvātie risinājumi projektēšanas procesa uzlabošanai. Šajā nodaļā aprakstīts ceturtais – noslēdzošais no šiem soļiem – stikla paketes parametru optimizācija. Iepriekšējās nodaļās esam izveidojuši risinājumi, kas spēj vienu stikla paketi pārbaudīt ļoti ātri. Izmantojot šo risinājumu, varam pārbaudīt daudzus variantus un veikt optimizāciju, lai minimizētu patērēto materiālu, kas izteikts kā visu stikla slāņu summa.

Vispirms apskatīsim, kādi ir optimizācijas parametri, konstantes un mērķi. Katras stikla paketes pārbaudes dati optimizācijas aspektā tiks saukti par “indivīdu”. Indivīda konstantes nemainīsies optimizācijas procesā. Šī datu daļa izriet no projekta prasībām un apkārtējā klimata. Indivīda parametri būs tie, kurus mainīsim optimizācijas procesā, lai minimizētu derīguma funkcijas lielumu. Jāpiebilst, ka šeit būtiska nozīme ir “konstantēm” – pieļaujamajiem spriegumiem un izliecēm atbilstoši normatīvam.

Tabula 7.1 Katra optimizācijas indivīda dati – konstantes, parametri un derīguma funkcijas rezultāts

Viena optimizācijas indivīda konstantes	Viena optimizācijas indivīda parametri	Viena optimizācijas indivīda derīguma (<i>fitness</i>) funkcijas rezultāts
Paketes platums	Ārējā stikla 1. slāņa stikla biezums	Visu stikla slāņu kopējais biezums
Paketes augstums	Ārējā stikla 2. slāņa stikla biezums	
Paketes tips – 2slāņu vai 3slāņu	Pirmās gāzes šķirkārtas biezums	
Pieļaujamais maksimālais spriegums atbilstoši normatīvam.	Vidējā stikla 1. slāņa stikla biezums	
Pieļaujamā maksimālā izliece atbilstoši normatīvam.	Vidējā stikla 2. slāņa stikla biezums	
Vēja spiediens uz ārējo slāni	Pirmās gāzes šķirkārtas biezums	
Vēja spiediens uz iekšējo slāni	Iekšējā stikla 1. slāņa stikla biezums	
Vēja sūce uz ārējo slāni	Iekšējā stikla 2. slāņa stikla biezums	
Vēja sūce uz iekšējo slāni	Vai ārējais stikls ir laminēts (jā/nē)	
Punkta slodze	Vai vidējais stikls ir laminēts (jā/nē)	
Līnijas slodzes vērtība	Vai iekšējais stikls ir laminēts (jā/nē)	
Līnijas slodzes augstums		
Līnijas slodzes pielikšanas puse		
Temperatūras starpība vasarā		
Meteoroloģiskais spied. vasarā		
Temperatūras starpība ziemā		
Meteoroloģiskais spied. ziemā		
Augstuma starpība		
Slodžu kombinācijas, kuras jāpārbauda.		

Katrs no optimizācijas parametriem ir apskatīts kā kategorisks mainīgais, jo lietoti tikai ražošanā pieejami stikla lokšņu un šķirkārtu biezumi.

Papildus Tabula 7.1 parādītajām konstantēm, bieži ir papildu ierobežojumi, ko inženieris grib iekļaut optimizācijas “ierobežojumos”. Šie ierobežojumi attiecās uz stikla uzbūvi, parasta/rūdīta stikla izmantošanu, kopējo šķirkārtas biezumu un šie ierobežojumi faktiski izriet no vizuālajām, drošības un siltum-pretestības prasībām. To uzskaitījums:

- Ierobežot kopējo gāzes šķirkārtu biezumu noteiktās robežās.
- Atļaut/aizliegt lietot laminētus stiklus. Šo iestatījumu var atsevišķi mainīt katram no stikla slāņiem.
- Atļaut/aizliegt lietot rūdītus stiklus. Šo iestatījumu var atsevišķi mainīt katram no stikla slāņiem.
- Atļaut tikai tādas pakešu uzbūves, kur ārējais stikls ir stingāks par iekšējo. Šī ir bieža prasība un labā prakse, lai neradītu optiskus defektus ēkas iemītniekiem.
- Atļaut lietot tikai tādus laminētus stiklus, kura kopā līmētajiem slāņiem abiem ir vienāds biezums.

7.2 Ģenētiskā optimizācija

7.2.1 Vispārīgi

Literatūras apskatā nodaļā 2.4 jau secinājām, ka līdz šim būvinženierijas jomā populārākie ir bijuši evolucionējošie un tieši ģenētiskie algoritmi. Šajā gadījumā varam izmantot to stiprās puses, jo mums ir kategoriski mainīgie, ātrs viena mēģinājuma laiks un derīguma funkcija nav konvekta (rezultāts ir stiklu biezumu summa izteikta milimetros). Tādēļ pētījumā lietosim ģenētiskās optimizācijas algoritmu.

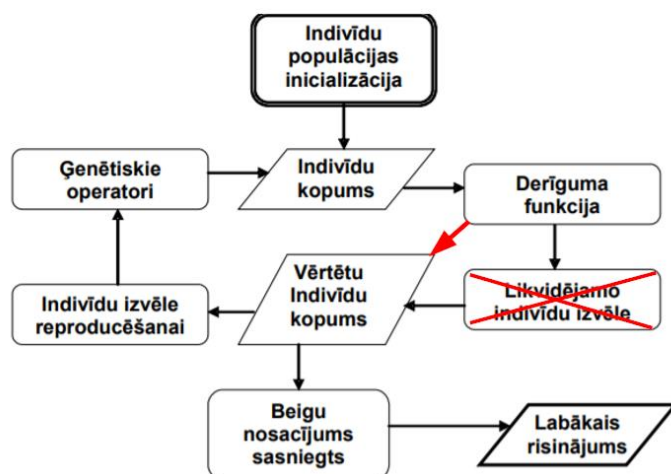
Ģenētiskie algoritmi izmanto tādas no bioloģijas aizgūtas metodes kā iedzimtība (*inheritance*), mutācija (*mutation*), izlase (*selection*) un krustošana (*crossover*) [64]. Populāciju veido potenciālie risinājumi jeb “indivīdi”, kurus reprezentē “hromosomas”. Katra “hromosoma” savukārt sastāv no vairākiem “gēniem” [64]. Tātad mūsu gadījumā vienam “gēnam” atbilst kāds no optimizācijas parametriem.

Tabula 7.2 Vienu optimizācijas “hromosomu” veidojošie gēni

Nr.	Gēni, kas veido hromosomu	Gēna iespējamās vērtības
1	Ārējā stikla 1. slāņa stikla biezums	4mm / 5mm / 6mm / 8mm / 10mm / 12mm
2	Ārējā stikla 2. slāņa stikla biezums	4mm / 5mm / 6mm / 8mm / 10mm / 12mm
3	Pirmās gāzes šķirkārtas biezums	8mm / 10mm / 12mm / 14mm / 15mm / 16mm / 18mm
4	Vidējā stikla 1. slāņa stikla biezums	4mm / 5mm / 6mm / 8mm / 10mm / 12mm
5	Vidējā stikla 2. slāņa stikla biezums	4mm / 5mm / 6mm / 8mm / 10mm / 12mm
6	Pirmās gāzes šķirkārtas biezums	8mm / 10mm / 12mm / 14mm / 15mm / 16mm / 18mm
7	Iekšējā stikla 1. slāņa stikla biezums	4mm / 5mm / 6mm / 8mm / 10mm / 12mm
8	Iekšējā stikla 2. slāņa stikla biezums	4mm / 5mm / 6mm / 8mm / 10mm / 12mm
9	Vai ārējais stikls ir laminēts	Jā / Nē
10	Vai vidējais stikls ir laminēts	Jā / Nē
11	Vai iekšējais stikls ir laminēts	Jā / Nē

Tabula 7.2 parāda gēnus, kas veido “hromosomu” un to iespējamās vērtības. Šeit arī varam redzēt būtiskas iezīmes, kas noteiks iespējas algoritma implementācijā. Pirmkārt, katrā hromosomā ir fiksēts skaits gēnu. Otrkārt, ir svarīga šo gēnu secība, jo katrs apzīmē kādu konkrētu parametru, Treškārt – gēnu vērtības nevar savstarpēji mainīt vietām, jo, piemēram., gāzes šķirkārtu iespējamie biezumi ir citādi (un citāds skaits) nekā stiklu iespējamie biezumi.

Vispārīga ģenētiskā algoritma darbības shēma parādīta Att. 7.1. Tā adaptēta no [64] un šajā avotā ir nosaukta par “paaudžu nomaiņas shēmu” – vienā solī pie “individu izvēles reproducēšanai” attiecīgi notiek labāko atlase, bet neviens indivīds netiek apzināti “likvidēts”. Pie “Ģenētiski operatori” daļas notiek vairākas darbības – gan izvēlēto indivīdu hromosomu krustošana (*crossover*), gan mutācija.



Att. 7.1 Vispārīga ģenētiskā algoritma shēma. Adaptēta no [64], labojumi ar sarkanu

7.2.2 Izvēlētais opcijas dažādu optimizācijas algoritma daļu implementācijā

Šajā nodaļā apskatīsim dažādas iespējas un izvēlētos piegājienu katrai no algoritma daļām – populācijas inicializācijai, derīguma funkcijai, indivīdu izvēlei reproducēšanai, krustošanai un mutācijai.

Populācijas inicializācija. Ir divi galvenie piegājieni – nejauša vai heuristiska [65]. Heuristiskas inicializācijas gadījumā, sākotnējā populācija tiek izvēlēta pēc kāda iepriekš noteikta likuma. Mūsu gadījumā tika izvēlēta nejauša inicializācija, lai izvairītos no “iekrišanas” kādos lokālos apgabalos. Šajā gadījumā visas gēnu vērtības ir kategoriskas, līdz ar to apskatāmais apgabals ir ierobežots un nav pamata uztraukties, ka optimums tiks meklēts pārāk plašā apgabalā.

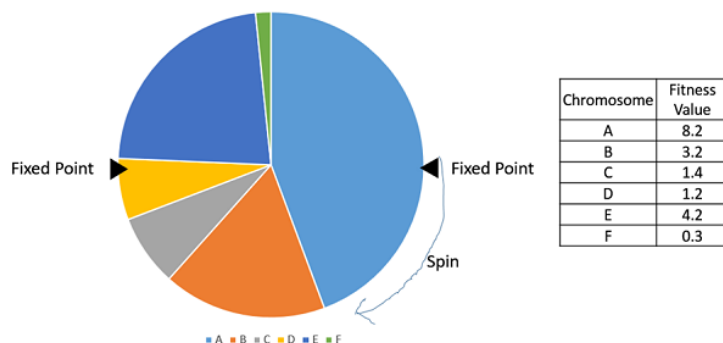
Indivīdu vērtēšana. Šī ir algoritma būtiskākā daļa, kam veltītas visas citas šī darba nodaļas. Att. 7.3 parāda shēmu, kā notiek viena indivīda vērtēšana. Šī shēma pēc būtības ir detalizētāka versija Att. 3.1 vidusdaļā redzamajām darbībām.

Indivīdu izvēle. Labāko indivīdu izvēlei var būt dažādas pieejas [64]:

Elitisms – jaunās paaudzes “sastāvu” veido no iepriekšējās paaudzes labākajiem indivīdiem. Mūsu gadījumā derīguma funkcijai ir noteikts skaits vērtību (minimālais kopējais biežums ir 2slāņi $\times 4 = 8\text{mm}$, maksimālais 6slāņi $\times 12 = 72\text{mm}$). Līdz ar to būs daudzi relatīvi tuvi rezultāti, un šādi mēs riskējam “iestrēgt” kādā lokāli optimālā risinājumā.

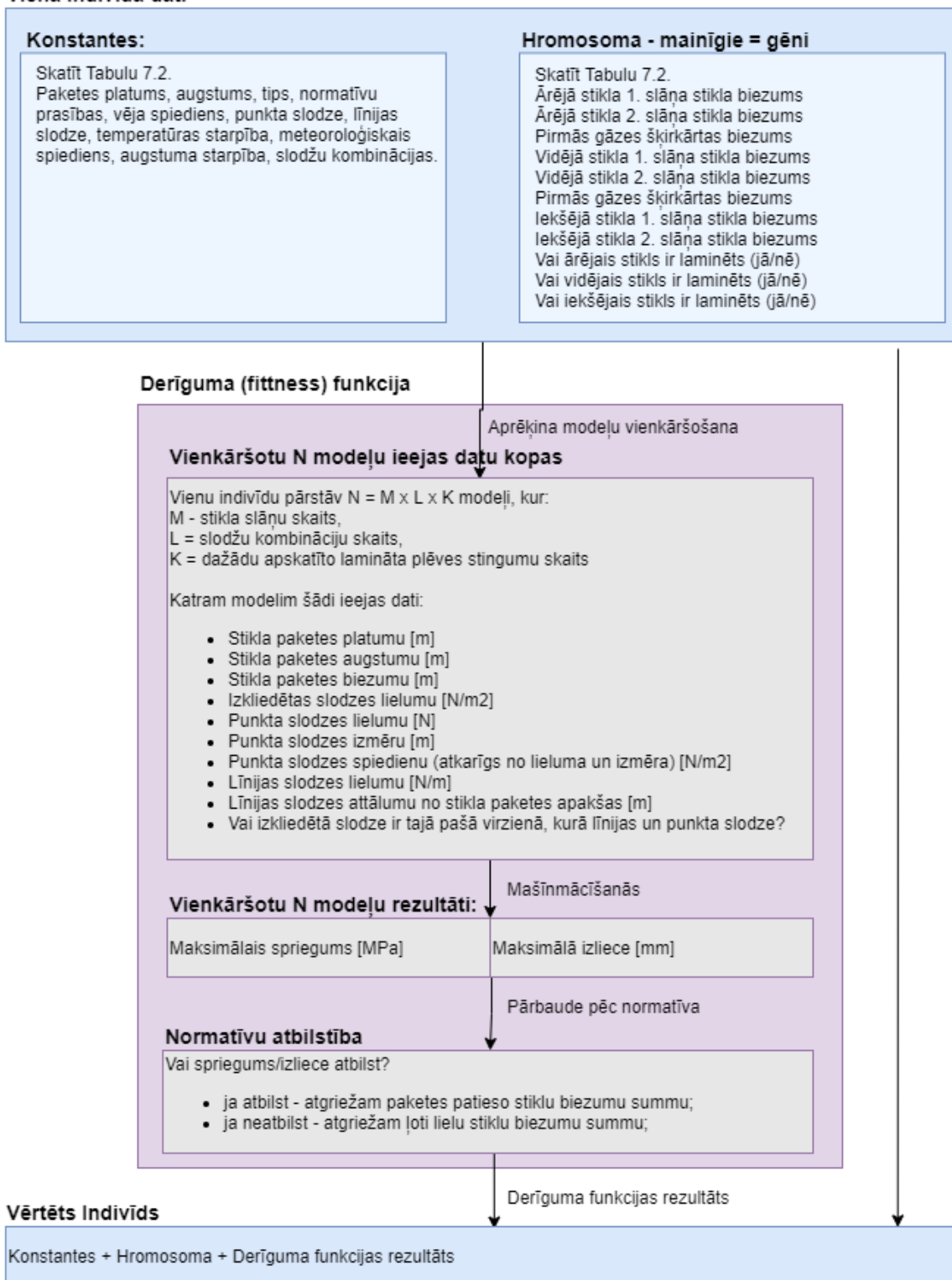
Derīgumam proporcionāla izvēle – pie šī tipa pieder gan “ruletes” izvēle, gan “stohastiskā universālā izlases ņemšana” (*Stochastic Universal Sampling*). Abi šie paveidi ir līdzīgi – galvenā atšķirība ir tajā, ka stohastiskā metode nodrošina, ka viens indivīds netiek iekļauts nākamajā paaudzē vairākas reizes. Skat. principa attēlojumu

Att. 7.2. Tieši šī metode ir izmantota šajā darbā, jo tā nodrošina gan priekšrocības labākajiem indivīdiem, gan nodrošina, ka tiek iekļauti mazāk veiksmīgi indivīdi, kuru apvidū var izrādīties kāds vēl labāks risinājums



Att. 7.2 Stohastiskās metodes principa grafisks attēlojums. Dažādu hromosomu “šķēļu” platumi tiek izvēlēti atbilstoši to derīgumam, un šis “pīrāgs” darbojas ar kā nejaušs ritenis. “Fixed point” skaits ir atkarīgs no nākamās paaudzes indivīdu skaita.

Viena indivīda dati



Att. 7.3 Derīguma funkcijas blokshēma

Mērogojamā izvēle – derīguma funkcijas diskriminējošais raksturs tiek palielināts atkarībā no indivīdu derīguma. Šī nav mūsu gadījumā pielietojama metode, jo indivīdu derīgums tiek “izlemts” pārbaudēs atbilstoši normatīvam un ir noteikts skaits iespējamo derīguma funkcijas iznākumu;

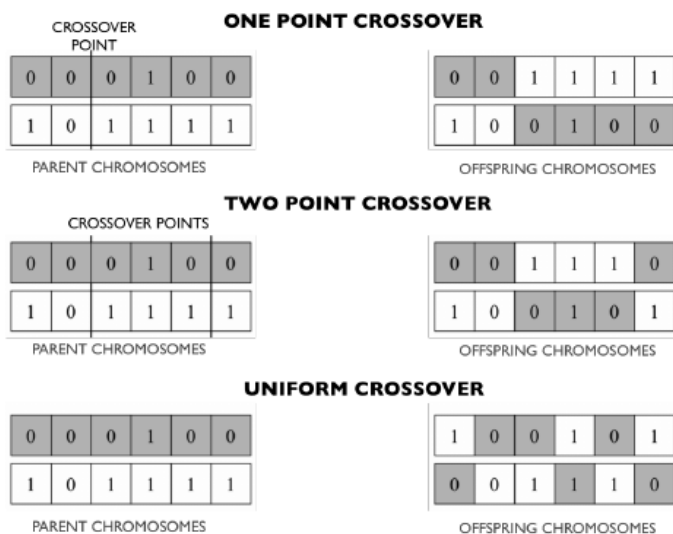
Spēļu turnīra izvēle – populācija tiek sadalīta grupās, kuras tiek atkal sadalītas grupās, līdz iegūstam pārus, no kura katra tiek izvēlēts “uzvarētājs”, kurš nākamajā kārtā “sacenšas” ar citu kārtu uzvarētājiem. Šī metode mūsu gadījumā nebūs efektīva, jo daudz indivīdiem ir paredzama vienāda derīguma funkcijas vērtība.

Kārtas numura izvēle – visi indivīdi tiek sakārtoti rindā pēc derīgum un izvēle ir atkarīga tikai no šī sakārtojuma (nevis derīguma funkcijas vērtībām). Arī šī metode mūsu gadījumam nebūs piemērota, jo daudz indivīdiem ir paredzama vienāda derīguma funkcijas vērtība.

Tātad – šī darba ietvaros kā piemērotākā indivīdu izvēles pieeja tika izvēlēta “stohastiskā universālā izlases ņemšana” (*Stochastic Universal Sampling*).

Genētiskie operatori. Krustošanas pieejas izvēle. Krustošanas galvenais uzdevums ir no diviem (vai vairāk) vecākiem ar labu derīguma funkcijas vērtību izveidot potenciāli vēl “derīgāku” jaunu indivīdu. Gan krustošanas, gan mutācijas gadījumā ir svarīgi ņemt vērā to, ka mūsu gadījumā nevaram mainīt gēnus vietām. Līdz ar to dažādas uz secības maiņu balstītas krustošanas pieejas nebūs piemērotas un praktiski ir trīs populārākās metodes, kas piemērotas mūsu gadījumam [66], kas vizuāli attēlotas Att. 7.4 [67].

Mūsu gadījumā lietojot “viena punkta” vai “divu punktu” krustošanu, tas nozīmētu, ka, piemēram, ārējā stikla uzbūve no viena “vecāka” tiktu kombinēta ar iekšējā stikla uzbūvi no cita. No inženierijas viedokļa tas nav loģisks risinājums, jo vēlamies krustošanu arī viena stikla slāņa ietvaros. Piemēram, ja apskatam 2 gēnus, kas reprezentē laminēta stikla 2 slāņus - ja vienam “vecākam” tie ir 6mm+4mm, un otram vecākam tie ir 5mm+5mm, tad mūsu vēlamais krustojums būtu 5mm+4mm. Tādēļ šajā darbā izvēlēta “vienmērīga” (*uniform*) krustošanas pieeja.



Att. 7.4 Krustošanas metodes - grafisks attēlojums [67]

Ģenētiskie operatori. Mutācijas pieejas izvēle. Mutācijas galvenais uzdevums ir ieviest ģenētisku “dažādību” jaunas indivīdu paaudzes veidošanā. Ir pieejama informācija par daudzām metodēm [68]. Tomēr vairums no tām nav piemērojamas mūsu gadījumam, jo mēs nevaram ģēnu vērtības mainīt vietām un pašas vērtības ir kategoriskas.

Piemēram, tādas metodes kā “Flip Bit” un “Bit string” darbojas tikai ar bināriem ģēniem. Daudzas citas – “Ievietošana” (*insertion*), “Pārvietošana” (*displacement*), “Daļēja sajaukšana” (*partial shuffle*), “Secības apgriešana” (*reverse sequence*) darbojas ar ģēnu secības maiņu. Vēl citas metodes – “Gaussian” un “Shrink” nevaram izmantot, jo mūsu vērtības ir kategoriskas un tām nevaram pieskaitīt/atņemt skaitļus.

Tādēļ tika izvēlēta vienīgā no metodēm, kas ir tiešā veidā piemērojama mūsu gadījumam – vienmērīgā (*Uniform*), kas praktiski nozīmē, ka ar noteiktu varbūtību esošais ģēns tiek aizstāts ar citu, nejaušu, vērtību. Varbūtības lieluma izvēli apskatīsim vēlāk šajā nodaļā.

Beigu nosacījums. Tā kā ģenētiskais algoritms ir heuristisks, tad tas pats par sevi negarantē optimālā risinājuma atrašanu, kā arī tas pats nezina, vai optimālais risinājums ir atrasts. Šī pētījuma ietvaros beigu nosacījumi netiek atsevišķi pētīti. Lietotājam – inženierim – tiek dota iespēja vai nu norādīt maksimālo algoritma darbības laiku, vai nu maksimālo paaudžu skaitu, pie kurām netiek panākts uzlabojums. Ir iespēja arī algoritmu apturēt manuāli, nospiežot pogu.

Paaudžu izmērs, mutācijas un krustošanās varbūtības izvēle. Šis ir plašs pētījumu lauks, kas jānorāda kā tālāk pētāms. Īpaši tādēļ, ka šie trīs parametri optimizācijas gaitā var tikt dinamiski mainīgi, lai sākumā apskatītu pēc iespējas plašāku optimizācijas parametru diapazonu, bet beigās atrastu pašu labāko risinājumu. Lai gūtu vispārīgu ieskatu šo parametru izmēros, apskatīsim literatūru. Avots [69] satur plašu citu pētījumu apskatu. Att. 7.5 parādīts [69] publicētais pārskats par 28 citos pētījumos lietotajiem paaudžu izmēriem, mutācijas un krustošanās varbūtību izvēlēm.

Redzam, ka paaudžu izmērs vispārīgi mainās no 15 līdz 100 (mediāna = 50), trijos gadījumos tas ir lielāks par 100, attiecīgi 330, 500 un 4000.

Lai labāk redzētu krustošanās un mutācijas datu sadalījumi, Att. 7.6 izveidots kastu grafiks. Krustošanās varbūtība mainās no 0.45 līdz 1, mediāna 0.8. Mutācijas varbūtība mainās no 0.001 līdz 0.5 ar vienu gadījumu, kurā tā ir 1. Mediānas vērtība = 0.05. Atsevišķi šis pētījums ar mutācijas varbūtību 1.0 netika apskatīts, bet, nezinot kontekstu, varētu spriest, ka pie šādas varbūtības katra paaudze ir neatkarīga no citām.

Kā minēts – optimālu šo parametru izvēle ir tālāks pētījumu virziens – veicot testus un skaitlisku iegūto rezultātu salīdzināšanu. Šī darba ietvaros būtiski ir atrast optimālo risinājumu,

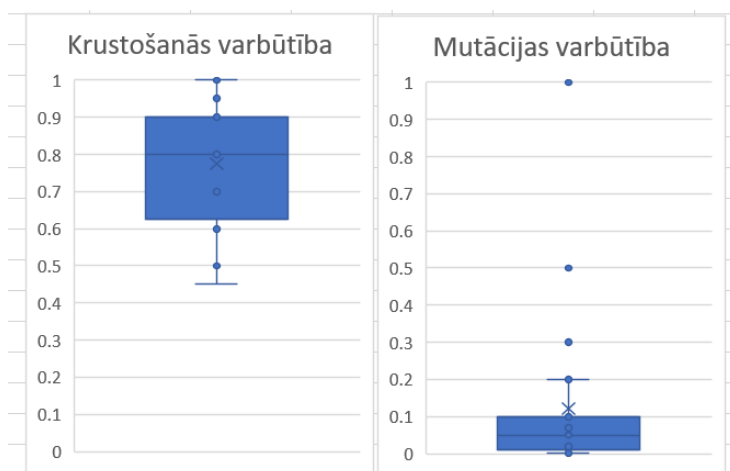
pat ja tā sameklēšanai nepieciešams nedaudz lielāks laiks nekā ar labākajiem hiperparametriem. Izvērtējot literatūrā pieejamo informāciju izvēlēti šādi parametri:

- Paaudzes izmērs = 50, mediānas vērtība no citos pētījumos lietotajām.
- Krustošanās varbūtība = 0.80, mediānas vērtība no citos pētījumos lietotajām.
- Mutācijas varbūtība = 0.10. Mutācijas varbūtības būtiski svārstās starp pētījumiem. Būtiska nozīme ir arī gēnu skaitam – mūsu gadījumā ir 11 gēni, kas nozīmē, ka ļoti mazas varbūtības var neievieš gandrīz nekādu mutāciju (piemēram, ar paaudzes izmēru 50 un 11 gēniem katram indivīdam, kopā būs 550 gēni, ja mutēsim ar varbūtību 0.001, tad var gadīties, ka neviens gēns netiek mutēts). Izvēlēta mutācijas varbūtība ir lielāka par citos pētījumos lietoto vērtību mediānu 0.05. Izvēlēta varbūtība 0.1 nozīmē, ka vidēji katram indivīdam tiks mutēts viens gēns.

Table 1. The GA parameters, PS - population size, CP - crossover probability, MP - mutation probability, used in previous works.

Reference	PS	CP	MP
[66]	30	0.95	0.01
[66]	80	0.45	0.01
[67]	100	0.9	1
[68]	100	0.8	0.005
[69]	50	0.9	0.03
[69]	50	1	0.03
[70]	50	0.8	0.01
[71]	15	0.7	0.05
[72]	100	1	0.003
[73]	30	0.8	0.07
[74]	100	0.8	0.01
[75]	500	0.8	0.2
[76]	40	0.6	0.1
[77]	100	0.8	0.3
[78]	100	0.6	0.02
[79]	30	0.6	0.05
[80]	76	0.8	0.05
[67]	4000	0.5	0.001
[81]	30	0.6	0.001
[82]	50	0.8	0.2
[83]	50	0.9	0.1
[84]	30	0.9	0.1
[85]	20	0.8	0.02
[86]	50	0.9	0.01
[87]	20	0.9	0.3
[88]	330	0.5	0.5
[89]	100	0.8	0.1

Att. 7.5 Citos pētījumos lietoto paaudžu izmērs, mutācijas un krustošanas varbūtības vērtības [69]

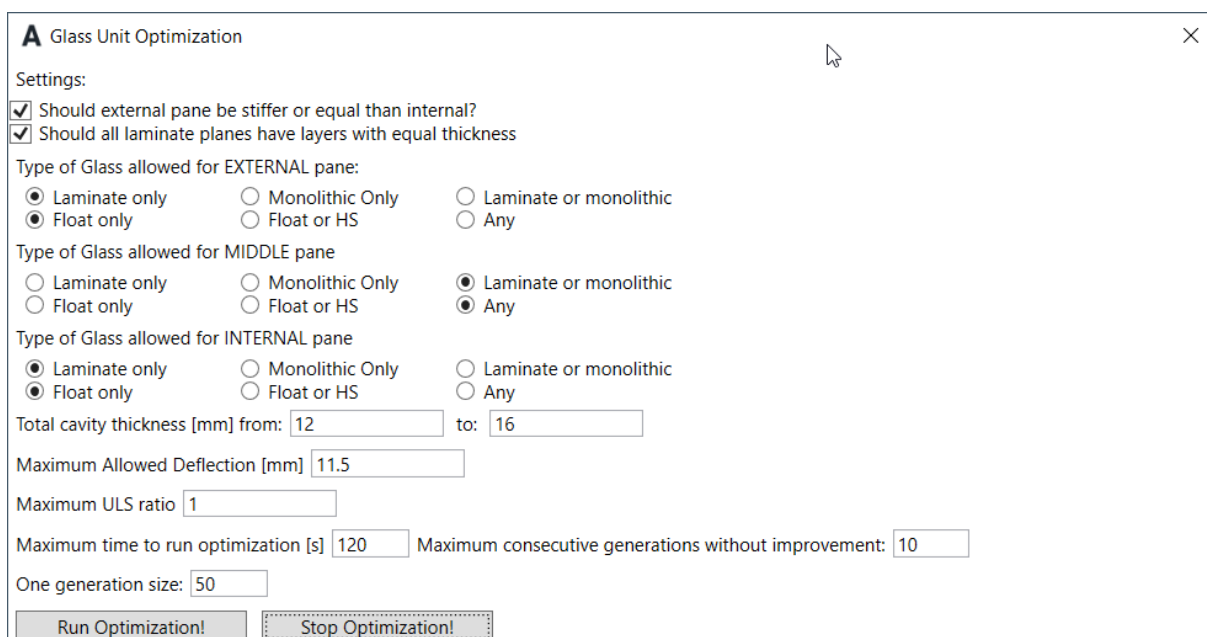


Att. 7.6 Citos pētījumos lietoto krustošānās un mutācijas varbūtību kastu grafiki, dati no [69]

7.3 Optimizācijas bibliotēkas izvēle un implementācija

Lai implementētu ģenētisko optimizāciju, tika izmantots ievars “GeneticSharp” [70]. Tā kā izveidotā lietojumprogramma ir veidota Windows videi, valodā C#, tad tika meklēts saderīgs ievars. Genetic Sharp izvēlēts, jo tas atbalsta visas nodaļā 7.2.2 veiktās paņēmieni/parametru izvēles, tas ir ērti lietojums un tā mājas lapa satur atsauces uz 24 citiem publicētiem pētījumiem, kuri izmantojuši šo bibliotēku, kas veido uzticību satvaram.

Pirmkods ar ievara implementāciju šī pētījuma ievaros pievienots 3. pielikumā. Izveidotais lietojumprogrammas optimizācijas logs ar iestatījumiem parādīts Att. 7.7.



Att. 7.7 Optimizācijas papildus ierobežojumi, ko var norādīt lietotājs

Galvenie implementācijas aspekti. No “GeneticSharp” tika lietotas šādas jau gatavas implementācijas:

- Individu izvēle – “Stohastiskā universālā izlases ņemšana” (Stochastic Universal Sampling). Pirmkods: <https://git.io/Js71U>
- Vienmērīga (*uniform*) krustošanās. Pirmkods: <https://git.io/Js71G>
- Vienmērīga (*uniform*) mutācija. Pirmkods: <https://git.io/Js711>
- Beigu nosacījums – noslēdzam vai nu pie noteikti patērētā laika, vai paaudžu skaita, kurās nenotiek neviens uzlabojums. Papildus implementēta iespēja lietotājam manuāli pārtraukt procesu. Pirmkods: <https://git.io/Js71X> un <https://git.io/Js7Mt>

Praktiski pilnībā no jauna (implementējot “GeneticSharp” satvara “interfeisus” - saskarnes) tika veidotas implementācijas šādiem aspektiem:

- Individu vērtēšana – skatīt Att. 7.3 un 3. pielikuma failu. Pirmkods: <https://github.com/mluksevics/Thesis/blob/master/Optimization/FitnessGlassUnit.cs>
- Hromosoma ar konstantu gēnu skaitu un katra gēna nozīmi. Pirmkods: <https://github.com/mluksevics/Thesis/blob/master/Optimization/ChromosomeGlass.cs>

7.4 Optimizācijas darbības izvērtējums

Ģenētiskais algoritms ir heuristisks algoritms – tāpat varam runāt tikai par labāko atrasto, nevis “absolūti labāko” stiklu uzbūvi, kas atrasta. Atsaucoties uz darba šī darba mērķi - algoritmam pēc ievades datiem jāspēj būvinženierim parādīt vairākus iespējamus stikla paketes uzbūves variantus, kas atbilst konstruktīvajām prasībām normatīvos.

Tā kā šī darba procesā esam izvēlējušies ģenētisko algoritmu, nevis pilnu variantu pārslasi (neiespējami veikt racionālā laikā) vai kādu citu optimizācijas metodi, tad praktiski “vairāki iespējamie stikla varianti” ir labākie varianti pēc katras ģenētiskā algoritma “paaudzes”.

Vērtēšanai ir izvēlētas 4 dažādas stikla pakešu uzbūves, kas reāli tikušas lietotas projektos un radušās pieredzējušiem inženieriem manuāli “meklējot” labāko risinājumu, pārbaudot dažādus variantus. Tabula 7.3 apkopo ievades datus un rezultātus. Redzams, ka visos gadījumos

Komentāri par katru no projektiem, kurā lietotas stiklu uzbūves:

1. Biroju ēka Londonā.

- a. Ievades dati: Divslāņu pakete ar relatīvi tipisku izmēru 1150x3500. Pieliktas gan līnijas, gan punkta slodzes. Uztādīta prasība pēc ne-rūdīta stikla un laminēta stikla gan iekšpusē, gan ārpusē;

- b. Rezultāti: Optimizācijas algoritms spēja atrast “neparedzētu” optimālāko variantu, izvēloties ārējam slānim laminātu ar dažāda biezuma stikliem, tas notika ~30 sekundēs. Protams, nevar viennozīmīgi teikt, ka šāda “netradicionāla” laminēta stikla lietojums būtu lētāks, bet optimizācija pat pārsniedza cerības, parādot iepriekš neapskatītu variantu.
2. Viesnīca Liepājā.
- a. Ievades dati. Trīsslāņu pakete. Īpaša ar savu nelielo izmēru, kas nosaka, ka noteicošās ir nevis no ārpusē pieliktās slodzes, bet gāzes šķirkārtu izplešanās/saraušanās temperatūras un gaisa spiediena iespaidā. Klients ir akceptējis risinājumu, kurš 15% pārsniedz normatīvā atļautās vērtības. Šī atļautā spriegumu pārsniegšana ir lietota optimizācijas iestatījumos (Att. 7.7, iestatījums “*Maximum ULS ratio*”).
- b. Rezultāti. Algoritms pēc aptuveni 3.5 minūtēm atrada Risinājumu Tuvu izvēlētajam – ar 5mm+4mm laminātu kā ārējo slāni (nevis 8mm monolītu), taču turpmākajās 10 minūtēs (pēc kurām optimizācija tika apturēta) tā arī nespēja atrast manuāli izvēlēto biezumu. No šādiem gadījumiem nākotnē varēs izvairīties, tālāk pētot ģenētiskā algoritma parametrus. Tomēr arī šajā gadījumā rezultāts vērtējams kā labs – tika atrasta pakete ar kopējo stiklu biezumu 21mm, ļoti tuvu inženieru projektētajiem 20mm.
3. Slimnīca Norvēģijā.
- a. Ievades dati. Divslāņu pakete. Stikla pakete īpaša ar to, ka tās uzbūvi noteica deformācijas limiti, kuri turklāt bija relatīvi lieli – līdz ar to aprēķinos būtiski izpaužas nelineārais “lielu pārvietojumu” raksturs, sāk darboties spēki stikla plaknē.
- b. Rezultāti. Algoritms pēc 21 sekundēm atrada līdzīgu risinājumu, kā projektētais – ar nedaudz citādu biezumu sadalījumu starp iekšējo/ārējo slāni – bet tādu pašu kopējo biezumu.
4. Privātmāja Siguldā. Trīsslāņu pakete.
- a. Ievades dati. Šis objekts izvēlēts, jo šeit lietota Latvijai un Skandināvijai “tipiska” trīsslāņu pakete, kur tikai iekšējam slānim jābūt laminētam, kā arī paketei ir relatīvi liels izmērs – 2100mm platums un 2500mm augstums.

- b. Rezultāti. Aptuveni pēc minūtes algoritms atrada uzbūvi ar tādu pašu biezumu (kopējais biezums 22mm), kā manuāli izvēlētu, pēc ~2minūtēm par 1mm plānāku kopējo uzbūvi un pēc~4minūtēm vēl par 1mm labāku. Tātad algoritma atrastais kopējais biezums bija 20mm – labāks nekā inženiera izvēlētais.

Diskusija/vērtējums. Divos no četriem gadījumiem ģenētiskās optimizācijas algoritms, darbinot mašīnmācīšanās modeļus, atrada plānāku stikla biezumu, nekā projektēts. Vienā gadījumā algoritms “iestrēga”, bet tuvu labākajam rezultātam. Vienā no gadījumiem algoritms parādīja iepriekš neapzinātu veidu (no inženiera perspektīvas), kā samazināt kopējo stikla biezumu.

Kopumā es, kā darba autors un inženieris, vērtēju optimizācijas risinājumu kā praktiski izmantojamu. Visos gadījumos algoritms mazāk nekā 5 minūtēs atrada tādu pašu, labāku vai līdzīgu risinājumu. Šis laiks ir mazāks par ~25 minūtēm, kas nepieciešamas vienas paketes pilna aprēķina laiku programmā SJ Mepla (skat. 14. lpp.).

Uz šī darba novitāti norāda aspekts, ka šobrīd nav pieejamas publiskas informācijas par kādu rīku, ar ko varētu salīdzināt šī darba laikā izveidoto sistēmu vai etalonuzdevumiem stiklu biezumu optimizācijas sfērā.

Tabula 7.3 Optimizācijas algoritma rezultātu salīdzinājums ar inženiera izvēlētām stikla uzbūvēm

Parametrs	Biroju ēka Londonā	Viesnīca Liepājā	Slimnīca Norvēģijā	Privātmāja Siguldā
Stikla paketes izmērs	3500 x 1150	416 x 482	1440 x 2550	2500 x 2100
Stikla paketes tips	Divslāņu	Trīs-slāņu	Div-slāņu	Trīs-slāņu
Inženieru izvēlētais ārējais stikls	6mm+6mm	8mm	6mm	8mm
Inženieru izvēlētā gāzes šķirkārta 1	14mm	14mm	10mm	16mm
Inženieru izvēlētais vidējais stikls	-	4mm	-	6mm
Inženieru izvēlētā gāzes šķirkārta 2	-	12mm	-	16mm
Inženieru izvēlētais iekšējais stikls	5mm+5mm	4mm+4mm	6mm	4mm+4mm
Inženieru izvēlētā stikla kopējais biezums	22mm	20mm	12mm	22mm
Inženieru izvēlēto gāzes šķirkārtas kopējais biezums	14mm	26mm	10mm	32mm
Automatizēti izvēlētais ārējais stikls	4mm + 8mm	5mm + 4mm	8mm	8mm
Automatizēti izvēlētā gaisa šķirkārta 1	15mm	10mm	14mm	18mm
Automatizēti izvēlētais vidējais stikls	-	4mm	-	4mm
Automatizēti izvēlētā gaisa šķirkārta 1	-	18mm	-	14mm
Automatizēti izvēlētais iekšējais stikls	4mm + 4mm	4mm + 4mm	4mm	4mm + 4mm
Automatizēti izvēlētā stikla kopējais biezums	20mm	21mm	12mm	20mm
Automatizēti izvēlēto gāzes šķirkārtas kopējais biezums	15mm	28mm	14mm	32mm
Algoritma uzstādījumi				
Vai atļauts rūdīts stikls?	Nē	Nē	Nē	Nē
Vai ārējam slānim jābūt monolītam/lamin.	Laminētam	Nav noteikts	Nav noteikts	Nav noteikts
Vai vidējam stiklam jābūt monolītam/lamin.	-	Nav noteikts	Nav noteikts	Nav noteikts
Vai iekšējam slānim jābūt monolītam/lamin.	Laminētam	Laminētam	Nav noteikts	Laminētam
Maksimāli atļautā gāzes šķirkārtu summa	18mm	30mm	14mm	36mm
Mīnīmāli atļautā gāzes šķirkārtu summa	14mm	26mm	10mm	32mm
Konstantes:				
Normatīvs, pēc kura veiktas pārbaudes	DIN18008	DIN18008	DIN18008	EN16612
Maksimāli pieļautā izliece	L/100=11.5mm	L/100=4mm	L/65 = 22.1mm	L/65 = 32.3mm
Vēja sūce	-1.21kPa	-1.77kPa	-1.73kPa	-0.84kPa
Vēja spiediens	+1.04kPa	+1.33kPa	-1.90kPa	+0.66kPa
Punkta slodze	0.5kN	Nav	Nav	Nav
Līnijas slodze	1.1kN/m@1.1m	Nav	Nav	0.5kN/m@1.1m
Klimats vasarā Temp./Augstums/Meteo spied.	38C/35m/2kPa	29C/12m/1kPa	20C/61m/2kPa	29C/100m/1kPa
Klimats ziemā Temp./Augstums/Meteo spied.	-25C/35m/-4kPa	-25C/12m/-4kPa	-25C/0m/-4kPa	-25C/100m/-4kPa
Laminēta stikla plēves stingums	Pilnīgi sting un bez stinguma	Pilnīgi sting un bez stinguma	Lietoti monolīti stikli	Konstants 0.35GPa
Algoritma rezultāti:				
Vai tika atrasta inženieru izvēlētai uzbūve atbilstoša, vai labāka?	Jā	Nē	Jā	Jā
Cik ilgi, aizņēma atrast uzbūvi ar kopējo stikla biezumu tādu kā inženiera izvēlētā.	29 sek.	Tuvāko atrada pēc 209 sek.	21 sek.	65 sek.
Vai algoritms bija spējīgs atrast uzbūvi par izvēlēto labāku (ar mazāku kopējo biez.)?	Jā	Nē	Nē	Jā
Ja jā, cik ilgs laiks pagāja, līdz tik atrasta labāka uzbūve	29.1sek.	N/A	N/A	114 sek.

8. IZVEIDOTĀS LIETOJUMPROGRAMMAS APRAKSTS

Izmantojot šajā darbā izpētīto par aprēķina vienkāršojumiem, mašīnmācīšanās modeļiem un parametru optimizāciju, ir izveidota lietojumprogramma stikla pakešu aprēķinam. Lai gan lietojumprogramma izstrādāta priekš iekšējās lietošanas A/S UPB meitas uzņēmuma SIA Aile Grupa ietvaros, to pilnībā esmu veidojis es, šī darba autors. Šī programma tiek lietota, lai veiktu sākotnējās stikla pakešu konstruktīvās pārbaudes. Programma faktiski sastāv no divām daļām:

- Windows WPF izpildāmas programmas, rakstītas valodā C#, izmantojot “.NET Framework”;
- Uz virtuālajām mašīnām izvietotiem mašīnmācīšanās modeļiem, kuri saņem un atgriež prognožu pieprasījumus caur HTTP.

Lai darbotos galīgo elementu aprēķini, ir jābūt instalētai programmatūrai Dlubal RFEM, jo galīgo elementu aprēķina modeļiem tiek izmantota šī programma caur tās API;

Windows programmas grafiskā saskarne parādīta Att. 8.1. Īss video ar programmas darbību ir pieejams saitē: <https://youtu.be/Q8qqFw6gQCA>

Input and Overview | Deflection Results | Stress Results

General

Load data | Save as...

Width [mm]: 1150

Height [mm]: 3500

Glass Unit Type:

Single Pane (no cavity)

Double Glass Unit

Triple Glass Unit

Balustrade

Design Code:

DIN18008

EN16612

Interlayer Stiffness Type

Constant stiffness

None/Full stiffness

Variable stiffness

Set Interlayer properties

Glass Buildup

Ext1 [mm]: 12 Monolithic

PVB ext: 0.76

Ext2 [mm]: 6

Cavity1 [mm]: 14

Mid1 [mm]: 5 Monolithic

PVB middle: 0.76

Mid2 [mm]: 5

Cavity2 [mm]: 14

Int1 [mm]: 5 Monolithic

PVB int: 0.76

Int2 [mm]: 5

Edit load combinations

External Loads

Wind pressure [kPa]: 0.78 ext. face, 0.26 int. face

Wind suction [kPa]: -1.04 ext. face, -0.17 int. face

Point Load [kN]: 0.5

Point size [mm]: 100

Point height [mm]: 1750

Point load side: External Internal

Line load [kN/m]: 1.1

Line height [mm]: 1100

Line load side: External Internal

Climate Loads

Summer: 38 [C] temperature, 35 [m] altitude, 2 [kPa] meteo

Winter: -25 [C] temperature, -35 [m] altitude, -4 [kPa] meteo

Calculations

Machine learning predictions | Optimization

Mesh size [mm]: 115

Generate model and linear calculations

Run non-linear calculations on this number of linear result rows: 0 Calculate All

Run non-linear calculations

Results Overview

Float: TVG: ESG:

Ext1 stress: 15.28 MPa @ CO3 44% 30% 17%

Ext2 stress: 7.64 MPa @ CO3 22% 15% 9%

Mid1 stress: 0% 0% 0%

Mid2 stress: 0% 0% 0%

Int1 stress: 16.57 MPa @ CO41 82% 36% 21%

Int2 stress: 16.57 MPa @ CO41 82% 36% 21%

Ext. pane deflection: 2.64mm @ CO92 W/435

Middle pane deflection: 0mm @ CO0 W/0

Int. pane deflection: 5.31mm @ CO129 W/216

Create Calculation Report

Status bar: *Mesh size is larger than 0.5x point load size and may result in unreliable stress results in vicinity of point load!

1.1.0.0

Att. 8.1 "Glass Assistant" - lietojumprogramma, kas balstās uz šajā maģistra darbā veikto pētījumu

Programma piedāvā šādas iespējas:

- Stikla pakešu vienkāršotu galīgo elementu modeļu izveidošanu, implementējot Feldmeier [45] pieeju, kas lietota arī DIN 18808 [44] un EN16612 [43]. Gaisa šķirkārtas netiek modelētas – bet aizstātas ar to darbību simulējošām izkļiedētajām slodzēm;
- Laminēta stikla slāņu kopdarbības vienkāršošanu atbilstoši “Enhanced Effective Thickness” [47]
- Galīgo elementu modeļi tiek automatizēti veidoti un aprēķināti programmā Dlubal RFEM, lietojot tās API.
- Spriegumu un deformāciju pārbaudes atbilstoši normatīviem DIN 18808 [44] un EN16612 [43].
- Stikla pakešu pārbaudes, lietojot mašīnmācīšanās modeļus, kas izveidoti šī pētījuma laikā (skatīt nodaļu 6).
- Stikla pakešu automatizētu uzbūves izvēli, lietojot mašīnmācīšanās modeļus un ģenētisko optimizāciju (skatīt nodaļu 7). Šis “iedzīvina” šajā darbā sasniegto mērķi praktiskā pielietojumā.

Windows programma sastāv no šādām galvenajām daļām/moduļiem:

- Grafiskās saskarnes un tās vadīšanas;
- Objektiem, kas reprezentē stikla paketi, tās atsevišķus slāņus;
- Slodžu pielikšanas – gaisa šķirkārtu vienkāršošanas daļa;
- Saskarnes ar mašīnmācīšanās modeļiem. Pirmkods pieejams šī maģistra darba 3. pielikumā.
- Optimizācijas modulis. Pirmkods pieejams šī maģistra darba 3. pielikumā.
- Aprēķinu atskaišu ģenerēšanas modulis.

Mašīnmācīšanās modeļi ir izvietoti uz virtuālajām mašīnām ar “Ubuntu” 20.04 operētājsistēmu, 1GB operatīvo atmiņu un 1x vCPU. Lai gan šie ir mazāki parametri nekā šī darba testos lietotie, praktiski šis nodrošina ātru prognožu atgriešanu lietotājam. Modeļi ir izvietoti uz A/S UPB serveriem Latvijā – praktiskā izmantošanā tas ir būtiski, jo pieprasījuma laiks samazinājās no ~50ms (turp-atpakaļ “RTT”) uz “DigitalOcean” serveriem Londonā līdz ~10ms uz serveriem Latvijā. Nodaļā 6.7 prognožu laiks deformācijām un spriegumiem netika skaitīts kopā, jo katrs mašīnmācīšanās modelis tiek palaists uz atsevišķas virtuālās mašīnas un pieprasījumi veikti paralēli.

Šis pētījums un tā rezultātā radusies programma palīdz virzīties uz šādu procesu, lai izvēlētos stikla paketes uzbūvi:

- 1) Gāzes šķirkārtu aptuvenais biezums tiek izvēlēts atbilstoši siltum-pretestības prasībām. Šis ir arī galvenais faktors, kas nosaka izvēli starp 2-slāņu un 3-slāņu paketēm, jo vienas šķirkārta biezums nav lielāks par 18mm.
- 2) Tiek izvērtētas prasības pēc laminētiem stikliem – tādas ir plaknēs, kurām var piekļūt cilvēki.
- 3) Tiek apkopota informācija par stikla paketes izmēriem, slodzēm, normatīvu;
- 4) Tiek lietota izveidotā lietojumprogramma “Glass Assistant”:
 - a. Tiek palaista stikla paketes biezuma optimizācija, lietojot mašīnmācīšanās modeļus.
 - b. Pēc optimizācijas noslēgšanās, inženieris izvērtē optimizācijas rezultātu. Ja nepieciešams, pārbauda vēl vairākus “tuvus” uzbūves variantus, tādējādi gūstot pārlicību par optimizācijas rezultātu.
 - c. Izvēlētajam variantam tiek veikts nelineārs galīgo elementu aprēķins, lietojot programmā iebūvēto sasaisti ar GEM programmu Dlubal RFEM. Tādā veidā “izslēdzam” mašīnmācīšanās prognožu iespējamo kļūdu.
- 5) Ja rezultāti ir jākomunicē ar kādu ārpus kompānijas (klientu, būvvaldi utml.) – tad tikai šis viens “fināla” variants tiek pārreķināts ar kādu atpazīstamu, industrijā uzticību guvušu, programmatūru (piemēram “SJ Mepla”). Šis aprēķins kalpo kā formāla pārbaude, nevis galvenais veids priekš stikla paketes uzbūves noteikšanas.

Lai šo procesu ieviestu dzīvē, inženieriem ir jāgūst pārlicība par izstrādātas programmatūras uzticamību. Tādēļ šobrīd aprēķini pēdējos projektos ir notikuši paralēli gan ar šī darba rezultātā veidoto programmatūru, gan komerciālām programmām “SJ Mepla” un “Glastik”.

2021. gada maija beigās (pirms šī darba nodošanas) izveidotās lietojumprogrammas iespējas ir dažādās stadijās. Galīgu elementu aprēķins, lietojot nodaļā 4 aprakstītos vienkāršojumus tika palaists februārī un tiek lietots paralēli citām komerciālām programmām. Mašīnmācīšanās iespējas ir pieejamas kopš maija sākuma, optimizācijas iespējas tiks darītas pieejamas inženieriem tuvākajās nedēļās.

9. REZULTĀTI

Literatūras apskatā tika izpētīts, ka divi Eiropā biežāk lietotie normatīvi DIN18008 un EN16612 stikla paketes gaisa šķirkārtas vienkāršošanai pielieto vienu un to pašu teorētisko bāzi – aprakstītu Feldmeier publikācijā [45]. Šī metodoloģija tika implementēta šī darba ietvaros izstrādātajā lietojumprogrammā. Tika identificēts, ka iepriekš minētie vienkāršoējumi ne tikai paātrina GEM modeļu aprēķinu, bet atļauj trenēt mašīnmācīšanās modeļus, kas pielietojami plašam stikla pakešu spektram. Skat. nodaļas 4 un 8.

Tika izveidoti kritēriji tādu mašīnmācīšanās modeļu novērtēšanai, kuri lietoti spriegumu un deformāciju prognozēšanai būvkonstrukcijas. Rezultāts ir uzskatāms kā labs priekš spriegumu un deformāciju paredzēšanas konstrukcijās, ja 99.7% no iegūtajiem rezultātiem ir ar precizitāti vismaz 95%, un maksimālā kļūda jebkurā no prognozēm nav lielāka par 35%. Skat nodaļu 6.2.

Darbā tika apskatīti trīs automatizēti mašīnmācīšanās risinājumi – no Google, Amazon un Microsoft. Kā piemērotākais spriegumu un deformāciju prognozēšanai tikai izvēlēts “Google Cloud AutoML Tables”, jo tas piedāvā labu prognožu precizitāti apvienojumā ar ātru prognozēšanas laiku. 150 prognožu atgriešana (aptuvenais skaits vienas stikla paketes aprēķinam) Google risinājumam aizņēma ~160ms, “Amazon AutoGluon” ~26000ms. Google un Amazon sasniedza līdzīgu precizitāti, bet “Microsoft Azure AutoML” sasniegtā precizitāte bija vairākas reizes zemāka. Skat nodaļu 6.3.

Salīdzinājums parādīja, ka vidējā kvadrātiskā logaritmiskā kļūda (RMSLE) kā modeļa trenēšanas metrika dod mazākas un vienmērīgāk sadalītas relatīvās kļūdas, nekā vidējā absolūtā kļūda (MAE) vai vidējā kvadrātiskā kļūda (RMSE). Veiktajiem testa modeļiem pie vienādām ievades un testa datu kopām, relatīvā kļūda sprieguma prognozēm lietojot RMSLE bija 2.20%, MAE, tā bija 2.71% un RMSE attiecīgi 5.39%. Līdz ar to RMSLE kā metrika ir piemērotāka konstrukciju spriegumu/deformāciju prognozēšanai. Skat nodaļu 6.4.

Tika secināts, ka mašīnmācīšanās modeļa precizitāti var būtiski uzlabot, ja treniņa datu ģenerēšanā ņem vērā datu kopas apgabalus, kuri paredzami kā bieži izmantojami prognozēšanai. Gadījumam, ja punkta un līnijas slodze nav pielikta (trīs no 8 ievades parametriem ir vienādi ar 0), relatīvā kļūda veiktajā izmēģinājumā samazinājās no 5.26% uz 2.18%, ja 25% treniņa datu aprakstīja tieši šādu gadījumu. Skat nodaļu 6.4.2.

Labākā šajā darbā sasniegtā precizitāte tika sasniegta ar ~2 miljoniem ievades datu, trenējot ar Google Cloud AutoML Tables risinājumu, izvēloties kā kritēriju RMSLE:

- 98.3% spriegumu prognozēšanai. Apskatot apgabalu, kurā prognozēm ir ietekme konstrukciju projektēšanā, 96.0% gadījumos kļūda bija mazāka par 5%, maksimālā testa kopā novērotā kļūda bija 132%. Kļūda lielāka par 35% bija 0.034% no testētajiem datiem. Skatīt nodaļu 6.6.
- 99.7% deformāciju prognozēšanai. Apskatot apgabalu, kurā prognozēm ir ietekme konstrukciju projektēšanā, 99.8% gadījumos kļūda bija mazāka par 5%, maksimālā testa kopā novērotā kļūda bija 23%. Skatīt nodaļu 6.5.

Iegūtie mašīnmācīšanās modeļi paātrināja vienas stikla paketes rezultātu iegūšanu ~160 vai vairāk reizes salīdzinot ar galīgo elementu aprēķinu. Skat nodaļu 6.7. Deformāciju un spriegumu prognozēšanas modeļi tika izvietoti uz atsevišķām virtuālajām mašīnām mākoņpakalpojumā. Prognožu pieprasījumi tika sūtīti paralēli, tādējādi paātrinot vienas stiklu paketes pilna aprēķina veikšanas laiku.

Literatūras apskatā tika secināts, ka provizoriski šāda veida uzdevumu (ar lielu viena eksperimenta ilguma laiku) optimizācijai būtu piemērotas Beijesa optimizācijas metodes. Tika arī secināts, ka darba rakstīšanās laikā nav pieejama gatava Beijesa optimizācijas bibliotēka lietošanai C# .NET vidē. Tādēļ tika lietota līdz šim konstrukciju optimizācijā visbiežāk lietotā metode – ģenētiskie algoritmi un implementēts “GeneticSharp” bibliotēkas pielāgojums. Skatīt nodaļu 7.

Lietojot “GeneticSharp” bibliotēku, tika apskatītas 4 dažādas stikla paketes, kuras jau tikušas lietotas projektos un kurām racionālākais risinājums izvēlēts pēc inženieru pieredzes un vairākām manuālām aprēķinu iterācijām. Trijos no gadījumiem optimizācijas algoritms, lietojot mašīnmācīšanās, modeļus atrada tādus pašus vai labākus risinājumus, kā tika izvēlēti manuāli un tam bija nepieciešamas 0.5 līdz 4 minūtes. Skatīt nodaļu 7.4.

Darba rezultātā ir izveidota C# WPF programma, kura veic stikla pakešu aprēķinus gan ar galīgo elementu metodi (izmantojot Dlubal RFEM API), gan lietojot mašīnmācīšanās metodes. Skatīt nodaļu 9. Šo programmu šobrīd izmēģina A/S UPB meitas-kompānijas SIA Aile Grupa inženieri, un pirmās atsauksmes ir pozitīvas, jo mašīnmācīšanās metode ļauj ātri atrast “aptuveno” paketes uzbūvi un to pārbaudīt ar precīzu galīgo elementu aprēķinu. Praktiski šobrīd tiek izmantoti mašīnmācīšanās modeļi ātrai stikla paketes uzbūves pārbaudei, bet optimizācijas iespējas vēl netiek izmantotas.

SECINĀJUMI

Izvirzot darba mērķi, bija skaidrs, ka darbs aptver plašu problēmu loku un tā vērtība ir divu nozaru zināšanu apvienojumā, tādējādi veicot pētījumu iepriekš maz pētītā virzienā.

Izvērtējot rezultātus un atskatoties uz darba gaitā iegūto pieredzi, es, kā darba autors, veicu šādus secinājumus:

- Šī darba novitāte ir pieejā, kādā stikla pakešu GEM modeļu vienkāršošana tiek kombinēta ar mašīnmācīšanās pielietojumu. Vienkāršojumi ir unikāli tieši priekš taisnstūrveida tikla paketēm. Tie atļauj apmācītos modeļus lietot plašam ievades datu diapazonam un iegūt rezultātus katrai slodžu kombinācijai. Populāri ir mašīnmācīšanās modeļus izmantot kā optimizācijas dzinēju = klasifikatoru, kas šajā gadījumā būtu neiespējami eksperimentu aprēķinu ilguma un daudzo iespējamo rezultātu dēļ.
- Mašīnmācīšanās risinājumu lietošanai un relatīvi labas precizitātes sasniegšanai ir pieejami ērti lietojami gatavi risinājumi. Kopš 2018. gada gan Google, gan Amazon, gan Microsoft ir gatavi risinājumi šajā segmentā, kas ir ļoti aktuāli prasmīgiem tehnoloģiju lietotājiem, kādi parasti ir būvinženieri.
- Darba gaitā autoram nostiprinājās viedoklis, ka ar tipiskiem mašīnmācīšanās modeļu rādītājiem, piemēram, absolūtā/relatīvā kļūda, ir par maz, lai noteiktu modeļa piemērotību praktisku būvinženierijas problēmu risināšanā un tādēļ tika izveidota metodika, lai vērtētu rezultātus pēc to relatīvās kļūdas sadalījuma.
- Veids, kādā tiek darbināti mašīnmācīšanās modeļi, dod ātrdarbības priekšrocības, salīdzinot ar galīgo elementu modeļiem. Viens mašīnmācīšanās modelis tiek palaists un sniedz prognozes plašam ievades datu klāstam. Turpretī, galīgo elementu aprēķinu gadījumā, katrai ieejas datu kopai tiek veidots jauns modelis, kas patērē papildu laiku sagatavošanas procesā.
- Maģistra darba izstrādes laikā gūtā pieredze un risinājumi ir derīgi praktiskam pielietojumam un jau tiek pamazām ieviesti procesos autora darba vietā - A/S UPB fasāžu biznesa līnijā.

Potenciālie turpmākie pētījumu virzieni.:

- Mašīnmācīšanās modeļu precizitātes uzlabošana stikla paketes maksimālo spriegumu prognozēšanai.
- Pētīt iespējamās optimizācijas ātrdarbības uzlabojumus, izvēloties dažādus “ģenētiskā” optimizācijas algoritma parametrus vai aizstājot to ar “Beijesa” optimizācijas algoritmu.

- Šis darbs apskata taisnstūrveida stikla paketes, to ir iespējams attīstīt, lai apskatītu trapecveida, trijstūra vai citu formu stikla paketes.
- Paplašināt optimizācijas iespējas, ievērtējot stikla pakešu siltuma pretestības un akustiskās īpašības.

IZMANTOTĀS LITERATŪRAS SARAKSTS

- [1] M. A. Kraus un M. Drass, «Artificial intelligence for structural glass engineering applications — overview, case studies and future potentials.,» *Glass Struct Eng*, sēj. 5, pp. 247-285, 2020.
- [2] J. Griffith, V. Marinov un G. Antonutto, «Applied Machine Learning in Structural Glass Design,» [Tiešsaiste]. Pieejams: <https://www.glassonweb.com/article/applied-machine-learning-structural-glass-design>. [Piekļūts 10. Janvārī 2021].
- [3] «Shenzhen Sun Global Glass,» [Tiešsaiste]. Pieejams: <https://www.sggglassmanufacturer.com/products/28-76mm-low-e-heat-strengthened-laminated-insulated-glass.html>. [Piekļūts 23. Janvārī 2021].
- [4] McKinsey & Company, «Decoding digital transformation in construction,» [Tiešsaiste]. Pieejams: <https://www.mckinsey.com/business-functions/operations/our-insights/decoding-digital-transformation-in-construction#>. [Piekļūts 2021 Janvārī 21.].
- [5] H. Liu, Z. Fu, K. Tang, X. Xu un M. Bauchy, «Machine learning for glass science and engineering: A review,» *Journal of Non-Crystalline Solids: X*, sēj. 4, 2019.
- [6] J. Ferreira-Cabello un E. Fraile-Garcia, «Metamodel-based design optimization of structural one-way slabs based on deep learning neural networks to reduce environmental impact,» *Engineering Structures*, sēj. 155, pp. 91-101, 2018.
- [7] T. Gulikers, «An Integrated Machine Learning and Finite Element Analysis Framework, Applied to Composite Substructures including Damage,» Delft University of Technology, Delft , 2018.
- [8] C. Bisagni un L. Lanzi, «Post-buckling optimisation of composite stiffened panels using neural networks,» *Composite Structures*, sēj. 58, nr. 2, pp. 237-247, 2002.
- [9] W. Ruijter, R. Spallino, L. Warnet un A. de Boer, «Optimization of composite panels using neural networks and genetic algorithms,» %1 *Proceedings Second MIT Conference on Computational Fluid and Solid Mechanics*, Cambridge (USA), 2003.
- [10] P. Hajela un L. Berke, «Neural networks in structural analysis and design: An overview,» *Computing Systems in Engineering*, sēj. 3, nr. 1-4, pp. 525-538, 1992.

- [11] O. S. A. Yetkin, «Design Space Exploration of Initial Structural Design Alternatives via Artificial Neural Networks,» %1 *Proceedings of the 37th eCAADe and 23rd SIGraDi Conference*, Porto, 2019.
- [12] L. Liang, M. Liu, C. Martin un W. Sun, «A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis,» *Journal of The Royal Society Interface*, sēj. 15, nr. 138, 2018.
- [13] Wikipedia, «Automated machine learning,» [Tiešsaiste]. Pieejams: https://en.wikipedia.org/wiki/Automated_machine_learning. [Piekļūts 18. Maijā 2021.].
- [14] L. Tuggener, M. Amirian un K. Romach, «Automated Machine Learning in Practice: State of the Art and Recent Results,» %1 *6th Swiss Conference on Data Science (SDS)*, Berne, 2019.
- [15] Kaggle, «A Deepdive into AutoML Tables | Tin-Yun Ho & Da Huang | Kaggle Days,» YouTube, [Tiešsaiste]. Pieejams: <https://www.youtube.com/watch?v=63HlsDDmvxo>. [Piekļūts 18. Maijā 2021].
- [16] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, L. Mu un A. Smola, «AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data,» 2020. [Tiešsaiste]. Pieejams: <https://arxiv.org/pdf/2003.06505v1.pdf>. [Piekļūts 23. Maijā 2021].
- [17] J. Mueller, «AutoGluon: achieve top 1% scores with 3 lines of code via AutoML,» [Tiešsaiste]. Pieejams: <https://www.kaggle.com/getting-started/140221>. [Piekļūts 23. Maijā 2021].
- [18] A. Truong, A. Walters, J. Goodsitt, K. Hines, C. B. Bruss un R. Farivar, «Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools,» %1 *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, Portlenda, 2019.
- [19] AI Multiple, «Top 24 AutoML software of 2021: In-Depth Guide,» [Tiešsaiste]. Pieejams: <https://aimultiple.com/automl-software/>. [Piekļūts 25. Martā 2021].
- [20] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd un J. H. Moore, «Automating Biomedical Data Science Through Tree-Based Pipeline Optimization,» %1 *Applications of Evolutionary Computation: 19th European Conference*, Porto, 2016.

- [21] L. Kotthoff, C. Thornton, H. H. Holger, F. Hutter un K. Leyon-Brown, «Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA,» *Journal of Machine Learning Research*, nr. 17, pp. 1-5, 2016.
- [22] M. Feurer, K. Eggenberger, S. Falkner, M. Lindauer un F. Hutter, «Auto-Sklearn 2.0: The Next Generation,» 8. Jūlijā 2020. [Tiešsaiste]. Pieejams: <https://arxiv.org/pdf/2007.04074.pdf>. [Piekļūts 23. Maijā 2020].
- [23] A. Arora, A. Candel, J. Lanford, E. LeDell un V. Parmar, «Deep Learning with H2O,» H2O.ai, Inc., Mountain View, 2016.
- [24] Google, «Release Notes | AutoML Tables | Google Cloud,» [Tiešsaiste]. Pieejams: <https://cloud.google.com/automl-tables/docs/release-notes>. [Piekļūts 23. Maijā 2021].
- [25] Google, «Google Cloud AutoML Tables,» [Tiešsaiste]. Pieejams: <https://cloud.google.com/automl-tables>. [Piekļūts 18 maijā 2021].
- [26] Wikipedia, «Mathematical Optimization,» [Tiešsaiste]. Pieejams: https://en.wikipedia.org/wiki/Mathematical_optimization. [Piekļūts 10. Janvārī 2021].
- [27] V. Tomei, M. Imbimbo un E. Mele, «Optimization of structural patterns for tall buildings: The case of diagrid,» *Engineering Structures*, sēj. 171, pp. 280-197, 2018.
- [28] J. Lim, C. You un I. Dayyani, «Multi-objective topology optimization and structural analysis of periodic spaceframe structures,» *Materials & Design*, sēj. 190, 2020.
- [29] A. Lemonge, J. Carvalho, H. P. un D. Vargas, «Multi-objective truss structural optimization considering natural frequencies of vibration and global stability,» *Expert Systems with Applications*, sēj. 165, 2021.
- [30] M. Esfandiari un G. Urgessa, «Progressive collapse design of reinforced concrete frames using structural optimization and machine learning,» *Structures*, sēj. 28, pp. 1252-1264, 2020.
- [31] C. Mitropoulou, G. Kazakis, S. Sotiropoulos, N. Lagaros un N. Kallioras, «Structural Optimization Computing Platform (SOCP) and SCIA Software,» *Procedia Manufacturing*, sēj. 44, pp. 410-416, 2020.
- [32] S. Barg, F. Flager un M. Fischer, «A design-focused, cost-ranked, structural-frame sizing optimization,» *Journal of Building Engineering*, sēj. 30, 2020.

- [33] M. Saka, O. Hasancebi un Z. Geem, «Metaheuristics in structural optimization and discussions on harmony search algorithm,» *Swarm and Evolutionary Computation*, sēj. 28, pp. 88-97, 2016.
- [34] D. Rutten, «Evolutionary Principles applied to Problem Solving using Galapagos,» [Tiešsaiste]. Pieejams: <https://www.grasshopper3d.com/profiles/blogs/evolutionary-principles>. [Piekļūts 20. Janvārī 2021].
- [35] «Grasshopper - algorithmic modeling for Rhino,» McNeel & Associates, [Tiešsaiste]. Pieejams: <https://www.grasshopper3d.com/>. [Piekļūts 20. Janvārī 2021].
- [36] C. Preisinger, «<https://www.karamba3d.com/>,» [Tiešsaiste]. Pieejams: <https://www.karamba3d.com>. [Piekļūts 23. Janvārī 2021].
- [37] V. R., «Octopus: Multi-Objective Evolutionary Optimization.,» [Tiešsaiste]. Pieejams: <https://www.food4rhino.com/app/octopus>. [Piekļūts 22. Janvārī 2021].
- [38] Wikipedia, «Bayesian optimization,» [Tiešsaiste]. Pieejams: https://en.wikipedia.org/wiki/Bayesian_optimization. [Piekļūts 22. Janvārī 2021].
- [39] «Hyperopt: Distributed Asynchronous Hyper-parameter Optimization,» [Tiešsaiste]. Pieejams: <http://hyperopt.github.io/hyperopt/#algorithms>. [Piekļūts 23. Janvārī 2021].
- [40] A. Mathern, O. Steinholtz un A. Sjöberg, «Multi-objective constrained Bayesian optimization for structural design,» *Structural and Multidisciplinary Optimization*, 2020.
- [41] C. Brecque, «The intuitions behind Bayesian Optimization with Gaussian Processes,» 26. Septembrī 2018. [Tiešsaiste]. Pieejams: <https://towardsdatascience.com/the-intuitions-behind-bayesian-optimization-with-gaussian-processes-7e00fcc898a0>. [Piekļūts 20. Janvārī 2021].
- [42] M. Feldmann un R. Kasper, Guidance for European Structural Design of Glass Components, Luxembourg: European Commission Joint Research Centre, 2014.
- [43] Technical Committee CEN/TC 129 “Glass in building”, «EN 16612 Glass in building - Determination of the load resistance of glass panes by calculation and testing,» EUROPEAN COMMITTEE FOR STANDARDIZATION (CEN), 2019.

- [44] Building and Civil Engineering Standards Committee, «DIN18808-1: Glass in building – Design and construction rules,» DIN Deutsches Institut für Normung e. V., Berlin, 2010.
- [45] F. Feldmeier, «Klimabelastung und Lastverteilung bei Mehrscheiben-Isolierglas,» *Stahlbau*, sēj. 75, nr. 6, pp. 467-478, 2006.
- [46] L. Galuppi un G. Carfagni, «The effective thickness of laminated glass: Inconsistency of the formulation in a proposal of EN-standards,» *Composites Part B: Engineering*, sēj. 55, pp. 109-118, 2013.
- [47] L. Galuppi, G. Manara un G. R. Carfagni, «Practical expressions for the design of laminated glass,» *Composites Part B: Engineering*, sēj. 45, nr. 1, pp. 1677-1688, 2013.
- [48] Wikipedia, «Finite element method,» [Tiešsaiste]. Pieejams: https://en.wikipedia.org/wiki/Finite_element_method. [Piekļūts 19. Janvārī 2021].
- [49] D. Bohmann, Mepla user manual - theory, Aachen: SJ Software GmbH, 2015.
- [50] mkt GmbH, User manual GLASTIK Professional 3.0, Alsdorf: mkt GmbH.
- [51] Technical Committee CEN/TC 250 "Structural Eurocodes", «EN 1990:2002+A1:2005 Eurocode — Basis of structural Design,» EUROPEAN COMMITTEE FOR STANDARDIZATION (CEN), 2010.
- [52] Trosifol, «PHYSICAL PROPERTIES OF SENTRYGLAS®AND BUTACITE®,» [Tiešsaiste]. Pieejams: https://www.trosifol.com/fileadmin/user_upload/Kuraray_4_1_Physical_Properties_of_Sentryglas.pdf. [Piekļūts 24. Janvāris 2021].
- [53] CNR, CNR-DT 210. Guide for the Design, Construction and Control of Buildings with Structural Glass Elements, Roma: CNR – Advisory Committee on Technical Recommendations for Construction , 2013.
- [54] L. Skotny, «What are the Types of Elements Used in FEA?,» [Tiešsaiste]. Pieejams: <https://enterfea.com/what-are-the-types-of-elements-used-in-fea/>. [Piekļūts 17 Janvārī 2021].
- [55] D. S. GmbH, «RFEM5 Program Description,» Dlubal Software GmbH, Tiefenbach, 2016.
- [56] O. Brooker, How to design reinforced concrete flat slabs using Finite Element Analysis, London: The Concrete Centre, 2006.

- [57] AutoGluon, «GitHub - awslabs/autogluon: AutoGluon: AutoML for Text, Image, and Tabular Data,» [Tiešsaiste]. Pieejams: <https://github.com/awslabs/autogluon>. [Piekļūts 23. Maijā 2021].
- [58] AutoGluon, «Tabular Prediction — AutoGluon Documentation 0.0.1 documentation,» [Tiešsaiste]. Pieejams: https://auto.gluon.ai/tutorials/tabular_prediction/index.html. [Piekļūts 23. Maijā 2021].
- [59] Microsoft, «What is automated ML? AutoML - Azure Machine Learning | Microsoft Docs,» [Tiešsaiste]. Pieejams: <https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml>. [Piekļūts 11. Maijā 2021].
- [60] R. Caruana, A. Niculescu-Mizil, G. Crew un A. Ksikes, «Ensemble selection from libraries of models,» %1 *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, Ņujorka, 2004.
- [61] AutoGluon, «FAQ — AutoGluon Documentation 0.2.0 documentation,» [Tiešsaiste]. Pieejams: https://auto.gluon.ai/stable/tutorials/tabular_prediction/tabular-faq.html. [Piekļūts 23. Maijā 2021].
- [62] AutoGluon, «TabularPredictor's predict is very slow! · Issue #376,» [Tiešsaiste]. Pieejams: <https://github.com/awslabs/autogluon/issues/376>. [Piekļūts 23. Maijā 2021].
- [63] Google, «Machine types | Compute Engine Documentation | Google Cloud,» [Tiešsaiste]. Pieejams: <https://cloud.google.com/compute/docs/machine-types>. [Piekļūts 23. Maijā 2021].
- [64] J. Zuters, « Ģenētiskie algoritmi un neironu tīkli,» [Tiešsaiste]. Pieejams: <http://home.lu.lv/~janiszu/courses/eanns/annsgenetic.pdf>. [Piekļūts 23. Maijā 2021].
- [65] TutorialsPoint, «Genetic Algorithms Tutorial - Tutorialspoint,» [Tiešsaiste]. Pieejams: https://www.tutorialspoint.com/genetic_algorithms/. [Piekļūts 23. Maijā 2021].
- [66] Wikipedia, «Crossover (genetic algorithm),» [Tiešsaiste]. Pieejams: [https://en.wikipedia.org/wiki/Crossover_\(genetic_algorithm\)](https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)). [Piekļūts 23. Maijā 2021].

- [67] U. Mehboob, J. Qadir, S. Ali un A. Vasilakos, «Genetic Algorithms in Wireless Networking: Techniques, Applications, and Issues,» *Soft Computing*, sēj. 20, p. 06, 2016.
- [68] Wikipedia, «Mutation (genetic algorithm),» [Tiešsaiste]. Pieejams: [https://en.wikipedia.org/wiki/Mutation_\(genetic_algorithm\)](https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm)). [Piekļūts 23. Maijā 2021].
- [69] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri un V. Prasath, «Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach,» *Information*, sēj. 10, nr. 12, p. 390, 2019.
- [70] D. Giacomelli, «GitHub - giacomelli/GeneticSharp: GeneticSharp is a fast, extensible, multi-platform and multithreading C# Genetic Algorithm library that simplifies the development of applications using Genetic Algorithms (GAs),» [Tiešsaiste]. Pieejams: <https://github.com/giacomelli/GeneticSharp/>. [Piekļūts 23. Maijā 2021].

PIELIKUMI

1. Pielikums. Programmas pirmkods GEM aprēķinu ievades datu ģenerēšanai

```

1. def generate_input():
2.     random.seed(round(time.time() * 1000 * 1))
3.     width = random.uniform(0.4, 3.0)
4.
5.     random.seed(round(time.time() * 1000 * 2))
6.     height = random.uniform(0.4, 5.0)
7.
8.     random.seed(round(time.time() * 1000 * 3))
9.     thickness = random.uniform(0.004, 0.024)
10.
11.    random.seed(round(time.time() * 1000 * 4))
12.    udl_magnitude = random.uniform(0, 25000)
13.
14.    random.seed(round(time.time() * 1000 * 5))
15.    point_load_magnitude = random.uniform(0, 2250)
16.    point_load_size = 0.1
17.    point_load_pressure = point_load_magnitude / (point_load_size * point_load_size)
18.
19.    random.seed(round(time.time() * 1000 * 7))
20.    line_load_magnitude = random.uniform(0, 2250)
21.
22.    random.seed(round(time.time() * 1000 * 8))
23.    line_load_height = random.uniform(0.01 * height, 0.5 * height)
24.
25.    random.seed(round(time.time() * 1000 * 9))
26.    line_point_direction = 1 if random.uniform(0, 1) > 0.5 else -1
27.
28.    mesh_size_typ = min(width, height) / 10
29.    mesh_size_point = min(mesh_size_typ, point_load_size / 2)
30.
31.    input_data = [
32.        width,
33.        height,
34.        thickness,
35.        udl_magnitude,
36.        point_load_magnitude,
37.        point_load_size,
38.        point_load_pressure,
39.        line_load_magnitude,
40.        line_load_height,
41.        line_point_direction,
42.        mesh_size_typ,
43.        mesh_size_point,
44.    ]
45.    data_string = " ".join([str(i) for i in input_data])
46.    data_hash = hash(data_string)
47.
48.    output_string = data_string + " " + str(data_hash)
49.    return output_string

```

2. Pielikums.

2. Pielikums. Programmas pirmkods palīg-programmai GEM aprēķinu veikšanai

Visual studio 2017 projekts un pilns C# pirmkods ir ievietots piekļuvei adresē:
<https://github.com/mluksevics/GlassExperimentRunner>

3. Pielikums.

3. Pielikums. Pirmkods mašīnmācīšanās un optimizācijas algoritmu implementācijai

C# pirmkods, kas parāda mašīnmācīšanās un optimizācijas algoritmu pielietojumu šī maģistra darba ietvaros ir ievietots piekļuvei adresē:

<https://github.com/mluksevics/Thesis/>

DOKUMENTĀRĀ LAPA

Maģistra darbs “**Stikla pakešu konstruktīvo parametru automatizēta noteikšana**”
izstrādāts LU Datorikas fakultātē.

Darba teksta galīgā versija izgatavota 24.05.2021.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____
(Autora paraksts un datums)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **p i e m ē r o t u / n e p i e m ē r o t u** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: _____
(Vadītāja paraksts un datums)

Darbs iesniegts **maģistratūras sekretariātā** _____.
(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: _____.
(Metodiķes paraksts)

Recenzents: _____ profesors, Dr. Math. Kārlis Podnieks _____
(Akad.amats, zin.grāds, vārds, uzvārds)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____
(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)