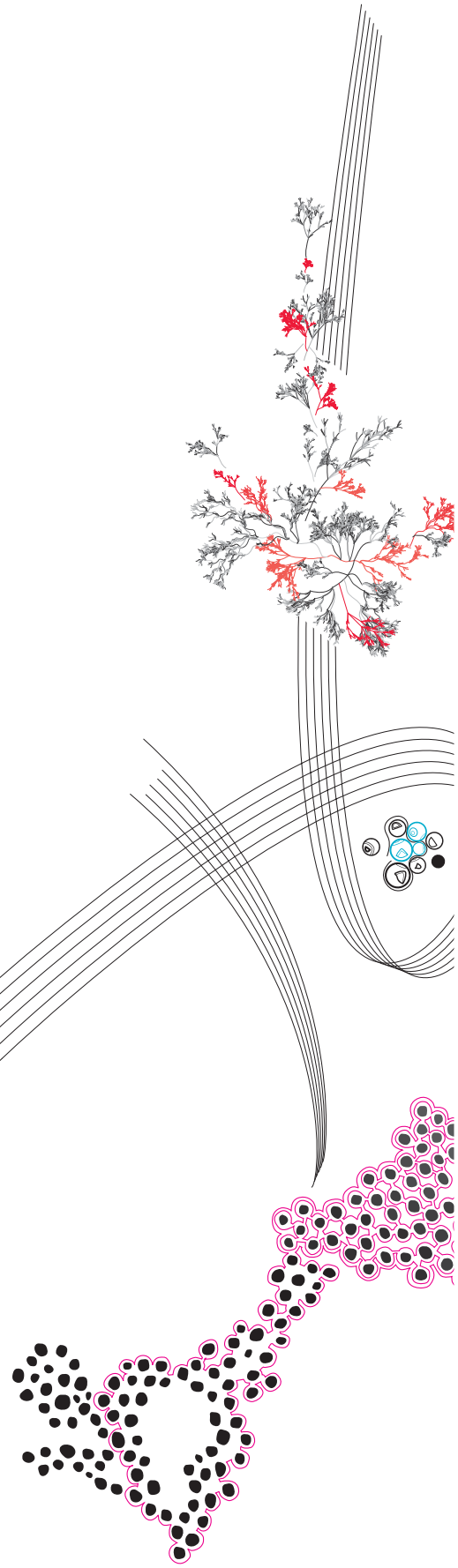
BSc Thesis Gezondheidswetenschappen



Een Monte Carlo-simulatie naar het verval van de effectiviteit van cybersecurity trainingsmaterialen in de zorg

Max Thijmen Podt

Begeleiders:
Jan-Willem Bullee
Christina Kolb

2 juli 2025

Faculteit Behavioural, Management- & Social Sciences
Universiteit Twente

UNIVERSITY OF TWENTE.

Een Monte Carlo-simulatie naar het verval van de effectiviteit van cybersecurity trainingsmaterialen in de zorg

M.T. Podt

2 juli 2025

Abstract

Om de tijdsinvestering van zorgmedewerkers efficiënter te benutten, richt dit onderzoek zich op de optimale frequentie van cybersecuritytrainingen, gebaseerd op literatuur over geheugenretentie. Het doel is te bepalen of alternatieve trainingsplanningen leiden tot betere retentie dan een jaarlijkse training, en welke frequentie nodig is om geheugenretentie boven een bepaalde drempelwaarde te houden. Hiervoor is een Monte Carlo-simulatie uitgevoerd waarin de vergeetcurve van Ebbinghaus parametrisch werd aangepast aan demografische en werkdrukfactoren. In totaal zijn 48 trainingsstrategieën vergeleken over een periode van vijf jaar: zeventien vooraf gedefinieerde en 31 dynamische experimenten op basis van drempelwaarden. Voor elk experiment werd de mediaan van de oppervlakte onder de retentiecurve berekend, met *bootstrapped* betrouwbaarheidsintervallen. De beste resultaten kwamen voort uit strategieën met een hogere trainingsfrequentie aan het begin van de bekeken periode. Gelijkmatig verdeelde trainingen bleken minder effectief dan twaalf van de zestien andere experimenten. De dynamische experimenten toonden aan dat vier en vijf trainingen respectievelijk nodig zijn voor 27% en 33% geheugenretentie. Deze bevindingen suggereren dat zorginstellingen met een vast aantal trainingen de geheugenretentie kunnen verbeteren door sessies te spreiden met afnemende frequentie en toenemende intervallen.

Trefwoorden: Geheugenretentie, cybersecurity, simulatie, trainingsfrequentie, Monte-Carlo, Ebbinghaus, gezondheidszorg, optimalisatie

Inhoudsopgave

1 Inleiding

"De zorg moet digitaal waar het kan en fysiek waar het moet"(ANP Producties, 2022). Hiermee gaf Ernst Kuipers, de toenmalige minister van Volksgezondheid, aan dat de zorg in Nederland onder druk staat en blikte hij vooruit op een digitale zorg als verlichting van deze werkdruk. Ook volgens onderzoek van het Centraal Bureau voor de Statistiek (2023) vindt 41,8% van alle zorgmedewerkers in Nederland de werkdruk te hoog.

Vergrijzing, stijgende personeelskosten en oplopende aantallen chronische patiënten zijn allemaal uitdagingen voor de zorg, waarbij digitalisering de potentie heeft om de zorg te verlichten van deze druk (Rijksinstituut voor Volksgezondheid en Milieu, 2022). Beveiliging en privacy van patiëntengegevens is één van de belangrijkste zorgen die bij digitalisering komt kijken, aangezien er strenge maatregelen nodig zijn voordat persoonlijke data veilig gesteld kunnen worden van ongewenste toegang en cyberaanvallen (Zorgbord, 2023).

Zorginstellingen in Nederland zijn in 2024 al meermaals slachtoffer geworden van cyberaanvallen, waarbij gevoelige persoonsinformatie werd gestolen en gelekt (Nationaal Coördinator Terrorismebestrijding en Veiligheid, 2024). Aangezien succesvolle cyberaanvallen in de zorg voornamelijk voortkomen uit fouten van medewerkers (Kruse e.a., 2017), is het fundamenteel voor het verminderen van cyberaanvallen om het zorgpersoneel bewuster te maken van de gevaren van een digitale zorg en er adequaat mee te leren handelen.

Cybersecurity trainingen dienen deze bewustwording bij medewerkers te verhogen en daarmee het aantal cyberaanvallen te verlagen. In de zorg is het zaak om deze trainingen te implementeren zonder een grote impact op de werkdruk te hebben. Om deze impact te minimaliseren, is het belangrijk met zo min mogelijk aantal trainingen de zorgmedewerkers zoveel mogelijk te laten onthouden van de inhoud. Dit leidt tot de vraag: “Wat is de optimale frequentie van cybersecurity trainingen in de zorg om geheugenretentie te bevorderen bij werknemers?” Om deze vragen te beantwoorden zal er in dit onderzoek een simulatie uitgevoerd worden om de consequenties op geheugenretentie en druk op het zorgpersoneel te vergelijken bij verschillende trainingsfrequenties en intervallen. In elk experiment zal een specifieke situatie gesimuleerd worden, waardoor deze verschillende trainingsfrequenties met elkaar vergeleken kunnen worden. Daarbij kunnen de uitkomstwaarden het effect op geheugenretentie per situatie weergeven.

Hierbij zullen de volgende deelvragen aan bod komen:

- *Wat is de optimale trainingsfrequentie voor de huidige hoeveelheid trainingen per jaar?*
- *Welke trainingsfrequentie is nodig om boven bepaalde drempelwaarden van geheugenretentie te blijven?*

De antwoorden op deze deelvragen zullen inzichten geven in de lange-termijn effecten van verscheidene indelingen van cybersecurity trainingen in de zorg.

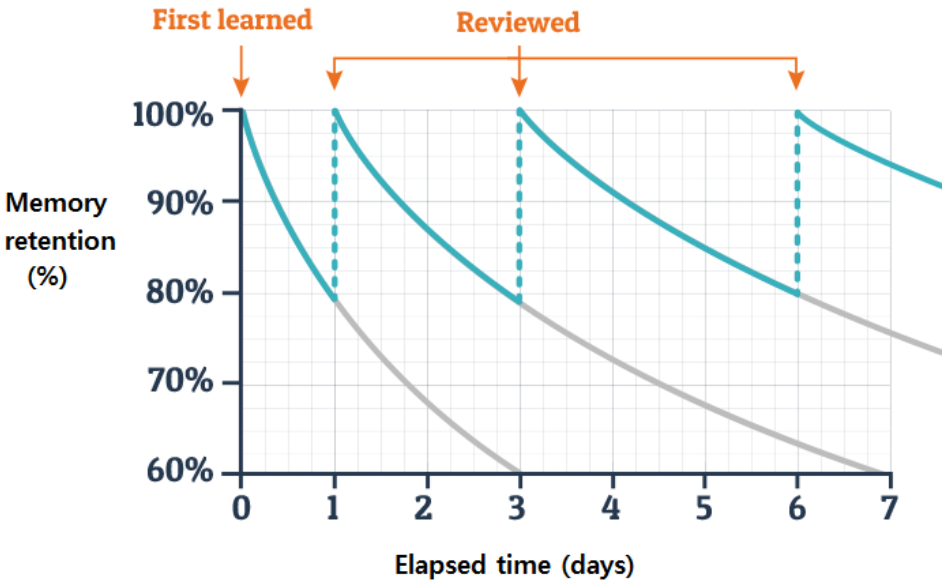
1.1 Relevantie

Aangezien dit onderzoek het doel heeft om onderbouwde richtlijnen neer te zetten over de frequentie van cybersecurity trainingen in de zorg, zal het een bijdrage leveren aan efficiëntere en duidelijkere cybersecurity strategieën voor ziekenhuizen en andere zorginstellingen. Daarnaast vult het onderzoek een gat in de literatuur op, aangezien er nog geen onderbouwde optimale frequentie voor cybersecuritytrainingen in de zorg bekend is.

2 Literatuur

2.1 Geheugenretentie

De simulatie zal berusten op wetenschappelijke kennis over algemene geheugenretentie, waarvoor de vergeetcurve van Ebbinghaus (1885) als basis gebruikt zal worden. Deze curve omschrijft de afname van geheugenretentie en het effect van een extra leermoment op de snelheid van de afname. In Figuur 1 is te zien dat een *review* (*booster*) van de stof resulteert in een functie die minder snel afneemt, waarbij de geheugenretentie over een langere periode strekt.



FIGUUR 1: Vergeetcurve van Ebbinghaus (Chun & Heo, 2018)

Ebbinghaus heeft in zijn originele studie slechts zijn eigen geheugenretentie getest, en deze datapunten vervolgens proberen te verklaren met een wiskundige functie. Hierbij heeft hij in zijn onderzoek de volgende twee functies benoemd waarbij x en $Q(t)$ de geheugenretentie weergeven, en t de tijd in minuten:

$$x = 1 - \left(\frac{2}{t}\right)^{0.099-0.51} \tag{1}$$

$$Q(t) = \frac{1.84}{log(t)^{1.25} + 1.84} \tag{2}$$

Tegenwoordig is het onderzoek echter vaker gerepliceerd, en zijn er functies geformuleerd die al deze datasets beter zouden verklaren dan functie 1 en 2. Uit het onderzoek van Murre en Dros (2015) blijken functie 3 en 4 beter in staat om alle meegenomen data te verklaren. Dit is gebasseerd op de *Sum of Squared Difference*(SSD), *Proportion Variance Explained*(R^2) en *Akaike Information Criterion*(IAC).

$$Q(t) = \mu_1 e^{a_1} + \mu_2 e^{a_2} \tag{3}$$

$$Q(t) = \mu_1 e^{-a_1 t} + \frac{\mu_1 \mu_2 (e^{-a_2 t} - e^{-a_1 t})}{a_1 - a_2} \tag{4}$$

Om onderscheid te maken tussen functie 3 en 4 wordt er in het onderzoek van Murre en Dros (2015) aangegeven dat functie 3 geen onderliggend model heeft om de functie te verklaren, waar functie 4 wordt verklaard door de *Memory Chain Model* (MCM), wat resulteert in een duidelijkere interpretatie van de parameters. Daarom zal functie 4 gebruikt worden om de Vergeetcurve van Ebbinghaus te simuleren, waarbij $Q(t)$ de geheugenretentie, t de verlopen tijd in minuten na het leermoment en μ_1, μ_2, a_1 en a_2 parameters zijn. Aangezien de functie een vast verval heeft, zal de functie aangepast moeten worden om het verval na een *review* te omschrijven. De parameters zullen daarom een rol spelen in het "verbuigen"van de functie bij een *review* om een langzamer verval te omschrijven, zoals te zien is in Figuur 1.

2.2 Monte Carlo Methode

Om een conclusie te kunnen trekken over alle zorgmedewerkers in Nederland zonder meer dan een miljoen mensen te hoeven betrekken bij het onderzoek, wordt een Monte Carlo simulatie gebruikt. Een Monte Carlo simulatie berust op aselecte steekproeven, waarbij de *Law of Large Numbers* aangeeft dat het gemiddelde van aselecte steekproeven richting de daadwerkelijke waarde zullen gaan, zolang er maar genoeg aselecte steekproeven genomen worden (Cevallos-Torres & Botto-Tobar, 2019). Als een dobbelsteen bijvoorbeeld 18 keer wordt gegooid, is het nog relatief realistisch dat er bijvoorbeeld 6 of 7 keer één bepaald getal wordt gegooid, in plaats van elk getal drie keer. Als de dobbelsteen echter 18.000 keer wordt gegooid, zal elk getal ongeveer 3.000 keer gegooid worden, en dit effect zal toenemen met meer worpen. Dit fenomeen kan in een simulatie gebruikt worden, door een bepaald aantal zorgmedewerkers te simuleren wat resulteert in een foutmarge vergeleken met de daadwerkelijke populatie. Doordat de steekproef aselect is en het aantal herhalingen groot genoeg, kunnen we op een gegeven moment aannemen dat de karakteristieken van deze populatie “dicht genoeg” bij de daadwerkelijke samenstelling ligt (MarbleScience, 2021) om aan een bepaalde foutmarge te voldoen. Dit maakt het mogelijk om met een beperkt aantal iteraties, toch een conclusie te kunnen trekken over de gehele populatie zorgmedewerkers.

2.3 Determinanten van geheugencapaciteit

Om de simulatie zo veel mogelijk op de realiteit te laten lijken, is het van belang om te kijken welke determinanten er door de literatuur worden omschreven over geheugencapaciteit. Zo kan er in de simulatie rekening gehouden worden met het feit dat zorgmedewerkers (en mensen in het algemeen) niet allemaal op dezelfde manier informatie onthouden. Hiervoor wordt er naar demografische kenmerken (geslacht, leeftijd en opleiding) en kenmerken die de druk op de zorgmedewerker omschrijven (arbeidsduur, werkdruk en werktevredenheid) gekeken.

2.3.1 Demografische kenmerken

Onder de demografische kenmerken wordt in dit onderzoek gefocust op geslacht, leeftijd en opleiding. Daarna is er gekeken of er interactie bestaat tussen deze kenmerken, om een compleet beeld te schetsen van de invloed van de demografie. Het verschil in geheugenretentie tussen mannen en vrouwen blijkt genuanceerd te liggen. Bloomberg et al. (2021) geeft aan dat vrouwen over alle leeftijdscohorten beter presteren op geheugenscore dan mannen. Pliatsikas et al. (2019) ondervonden echter dat mannen een marginaal significant

voordeel in geheugen hadden, waarbij de afname in geheugenscore bij mannen daarintegen sterker was dan bij vrouwen.

Het onderzoek van Pliatsikas et al. (2019) geeft ook inzichten in de relatie tussen werkgeheugen en opleiding. Het werkgeheugen blijkt uit de resultaten namelijk positief te worden beïnvloed door hogere opleiding.

Daarnaast vermeldden ze dat er een interactie bestaat tussen zowel leeftijd als opleiding met geslacht. Op oudere leeftijd hebben mannen namelijk een grotere afname in werkgeheugen dan vrouwen (zoals Bloomberg et al. ook concludeerden), maar daarnaast hebben vrouwen ook een groter positief effect door opleiding. Zowel het effect van opleiding als van leeftijd bleken lineair.

2.3.2 Kenmerken over werkdruk

Uit de relaties tussen de druk van de zorgmedewerker en geheugencapaciteit blijkt dat lange werkuren een negatief effect hebben op cognitieve vaardigheden. Virtanen et al. (2009) onderzocht namelijk het verschil tussen groepen die minder of gelijk aan 40 uur, tussen de 41 en 55 uur, of meer dan 55 uur per week werkten, waarbij de verschillen tussen alle groepen significant waren gevonden. Daarnaast is er ook gekeken naar werkuren als continue schaal (per stap van 10 uur) in plaats van groeperingen, en ook daar is er significant verschil aangetoond. Daarnaast voegt het onderzoek van Schoofs et al. (2008) toe dat psychosociale stress de prestatie van het werkgeheugen verminderd. Werkdruk en werktevredenheid waren in de literatuur niet direct te koppelen aan een positief of negatief effect op geheugencapaciteit. Daarom zal het effect van werkdruk in de simulatie voornamelijk afhangen van de arbeidsduur per week.

2.4 Arbeidsmarktprofiel van zorg en welzijn

Ten slotte is het profiel van de zorgmedewerkers in Nederland essentieel voor de simulatie, omdat de gevonden determinanten van geheugen toegepast moeten worden op de juiste gesimuleerde zorgmedewerker. Daarvoor zijn de statistieken van het arbeidsmarktprofiel van zorg en welzijn in 2022 van het CBS (2023) gebruikt. In Tabel 1 zijn alle relevante percentages genoemd.

Wat is de optimale frequentie van cybersecurity trainingen in de zorg om geheugenretentie te bevorderen bij werknemers?

3 Methode

3.1 Procedure

De methodologische uitvoering van dit onderzoek begon met een literatuurstudie, gericht op het verkrijgen van inzicht in de theoretische basis en onderliggende aannames van de simulatie. Op basis van deze theoretische kaders zijn vervolgens de relevante waarden van de parameters geïdentificeerd die de werkelijkheid op een adequate manier representeren.

Na het vaststellen van de parameters is de volledige simulatie geïmplementeerd in Python. Zowel de bijbehorende pseudocode als de code zijn opgenomen in de bijlage. Om het aantal benodigde iteraties te achterhalen is de relatie tussen de foutmarge en de aantal iteraties gevisualiseerd in Figuur 2.

Tabel 1: Arbeidsmarktprofiel van zorg en welzijn (CBS, 2023)

Geslacht	
Man	Vrouw
15,9%	84,1%

Leeftijd		
Jonger dan 35	35 tot 55	55 of ouder
35,2%	40,1%	24,7%

Opleiding			
Laag	Middelbaar	Hoog	onbekend
11,7%	43,3%	43,0%	1,9%

Arbeidsduur (uur per week)				
<12	12 tot 20	20 tot 35	35 tot 41	>40
6,9%	9,9%	56,0%	20,7%	6,5%

Werkdruk (ervaart werkdruk als te hoog)	
Wel	Niet
41,8%	58,2%

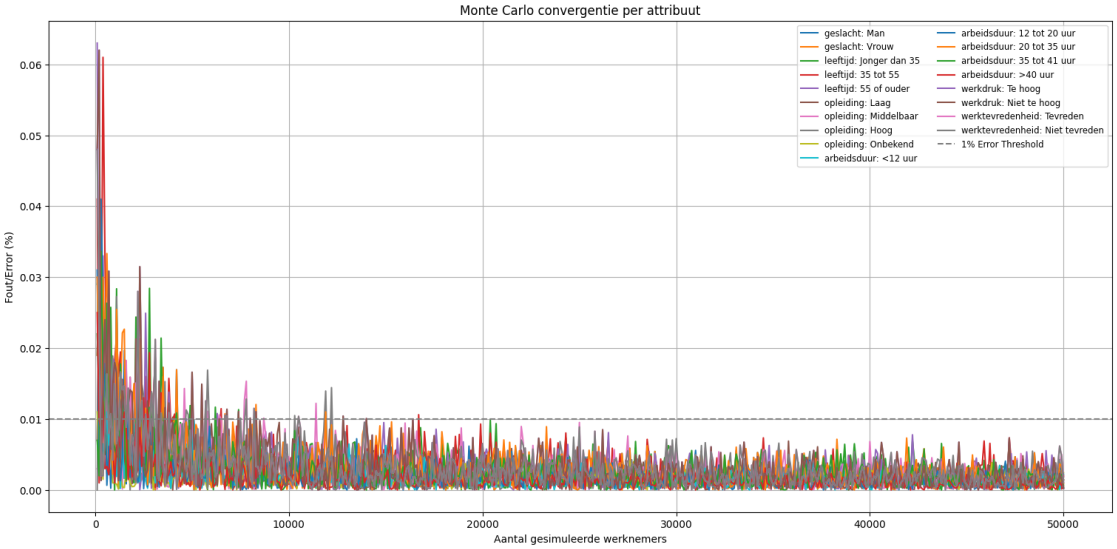
Werktevredenheid	
Tevreden	Niet tevreden
76,2%	23,8%

Met dit aantal iteraties is een initiële testrun uitgevoerd, met als doel de verdelingsvorm van de gegenereerde data te analyseren. Hierbij is vastgesteld of de data normaal verdeeld, scheef verdeeld of volgens een andere verdelingsvorm verspreid is. Op basis van deze analyse is vervolgens bepaald welke uitkomstmaat, het gemiddelde of de mediaan, het meest geschikt is voor de uiteindelijke dataverwerking.

Na deze voorbereidende stappen is de simulatie uitgevoerd voor alle experimentele condities. De gegenereerde data zijn verzameld en vervolgens onderworpen aan een kwantitatieve analyse.

3.2 Parametrische aanpassingen

De parameters spelen een belangrijke rol in de simulatie omtrend de vergeetcurve, waardoor er uitvoerig is gekeken naar de invloed van μ_1, μ_2, a_1 en a_2 in functie 4. Volgens het MCM(Memory Chain Model) zijn alle parameters te koppelen aan bepaalde geheugenfuncties. Deze parametrische definities zijn te vinden in Tabel 2. Voor de initiële waarden van de parameters is gekeken naar het onderzoek van Murre en Dros (2015). Hierin zijn vier soortgelijke onderzoeken bekeken, waarbij de parameters geoptimaliseerd zijn voor de data van die onderzoeken. Het gemiddelde van deze vier waarden is als initiële waarde genomen van de parameters. In Excel is er vervolgens gekeken naar de vorm van de curve afhankelijk van parameteraanpassingen, waarna de parameters van *reviews* werden vastgesteld. Voor de parameterverandering van de *boosters* is er gekeken naar het onderzoek van Chun



FIGUUR 2: Convergentie grafiek van error per iteraties

et al. (2018). De parameters werden zo aangepast dat de curve vergelijkbaar was als de functie die te zien is in Figuur 1. Hieruit bleek dat dit bewerkstelligd werd door mu2, a1 en a2 bij de eerste *booster* te vermenigvuldigen met 0.44. Voor elke volgende *booster* werd dit wederom vermenigvuldigd met 0.66. Voor de aanpassingen aan de parameters vanuit de demografische en werkdruk kenmerken werd naar de uitkomstmaten (afwijkingen in standaarddeviaties) van Bloomberg et al. (2021) gekeken. Vervolgens werden de aanpassingen van de andere kenmerken geschaald naar de b waarden van het regressiemodel van Pliatsikas et al. (2019), zodat de effecten van de kenmerken proportioneel aan elkaar zijn.

TABEL 2: Definities van parameters volgens Murre et al. (2015)

A1	Decay rate in store 1 (hippocampus, mainly short term memory)
A2	Decay rate in store 2 (neocortex, mainly long term memory)
MU1	Initial strength of memory traces in store 1
MU2	rate of consolidating the contents of store 1 to store 2

3.3 Experimentele samenstelling

Voor de samenstelling van de experimenten is de onderzoeksvraag en de bijbehorende deelvragen als uitgangspunt genomen. Hierbij is initieel gekeken naar vooraf bepaalde experimenten met praktische trainingsverdelingen (experiment 1 tot 17). Deze experimenten liepen gedurende een tijdsspanne van vijf jaar waarover vijf trainingen werden gesimuleerd. Op deze manier zijn alle experimenten vergelijkbaar met de huidige NEN7510/NIS2 richtlijnen van één cybersecurity training aan het begin van elk jaar (experiment 1). Tabel 3 omvat alle gesimuleerde experimenten die als mogelijke verbetering werden geacht ten opzichte van één training aan het begin van elk jaar. Specificering en groepering van deze experimenten zijn weergegeven in Tabel 3.

Naast deze zeventien vooraf bepaalde experimenten zijn er nog 31 dynamische experimenten uitgevoerd. Deze experimenten gebruiken een drempelwaarde (20% tot 50%), waarbij er een nieuwe training ingepland wordt wanneer de mediaan van alle medewerkers onder de drempelwaarde daalt. De experimenten zijn uitgevoerd met een stapgrootte van 1%

TABEL 3: Visuele weergave van experiment 1-17 waarbij X het trainingsmoment weergeeft

Exp. / **Maand**	**1**	**2**	**3**	**4**	**5**		**6**	**7**	**8**	**9**	**10**		**11**	**12**	**13**	**14**		**15**	**16**	**17**
1	X	X	X	X	X		X	X	X	X	X		X	X	X	X		X	X	X
2																				X
3																			X	X
4				X						X				X		X		X		X
5																			X	X
6																				
7					X			X			X		X					X	X	
8		X																		
9														X					X	
10							X								X			X		
11																				
12																				
13	X			X	X				X	X	X		X			X		X		
14																				
15								X												
16				X						X				X		X				
17		X													X					
18																				
19					X		X				X		X							
20			X																	
21																				
22															X					
23									X											
24																				
25	X							X					X	X	X	X				
26																				
27																				
28							X													
29																				
30		X																		
31									X											
32																				
33			X																	
34																				
35																				
36																				
37	X						X	X	X	X	X									
38																				
39																				
40																				
41																				
42			X																	
43																				
44																				
45																				
46																				
47																				
48																				
49	X	X	X	X	X															
50...																				

tussen 20% en 50%.

3.4 Dataverzameling

De data is verzameld door middel van de gecodeerde simulatie. Deze simulatie genereerde voor elk experiment een grafiek en een histogram, waar de mediaan en de *Bootstrapped* betrouwbaarheidsinterval (BI) in aangegeven staan. Zodra alle histogrammen en grafieken verzameld waren, zijn de medianen en de BIs verzameld en in Tabel 4 en Tabel 5 weergegeven. De data is verzameld over 13.000 gesimuleerde werknemers voor experiment 1-17 en 5.000 werknemers voor de dynamische experimenten. Dit resulteert respectievelijk in een 1% en een 2% foutmarge, wat de laagst haalbare foutmarge was voor de beschikbare apparatuur.

3.5 Datakenmerken

Uit de gegenereerde histogrammen bleek de data van de AUCs *right-skewed* te zijn. Dit geeft aan dat het meerendeel van de data onder het gemiddelde ligt. Bij de analyse van *skewed* data geeft de mediaan meer duidelijkheid over de data dan het gemiddelde. Daarom is er gekozen voor het vergelijken van de mediaan van alle histogrammen. Om de zekerheid van de mediaan te bepalen, is er gebruik gemaakt van *bootstrapping*. Hierbij is er 10.000 keer een herhaaldelijke steekproef met teruglegging gedaan, waaruit 10.000 medianen zijn gekomen. Over deze medianen kan je een BI bepalen om de zekerheid van de mediaan weer te geven. Er is daarom gestreefd naar een smalle BI.

3.6 Data analyse

De mediaan van de AUC van elk experiment is de uitkomstmaat waarop de experimenten zijn vergeleken. Een hogere mediaan van de AUC geeft aan dat er een hoger niveau van geheugenretentie is bij het experiment. Hierbij wordt het *bootstrapped* BI in acht genomen, aangezien de zekerheid en specificiteit van de mediaan hieruit afgeleid kan worden. Aangezien de experimenten in een gecontroleerde omgeving plaatsvinden waarbij de ruis optimaal beperkt is, kunnen de medianen verder direct met elkaar vergeleken worden en zijn statistische toetsen, die vaak afhankelijk zijn van onafhankelijkheid waar hier geen sprake van is, overbodig.

4 Resultaten

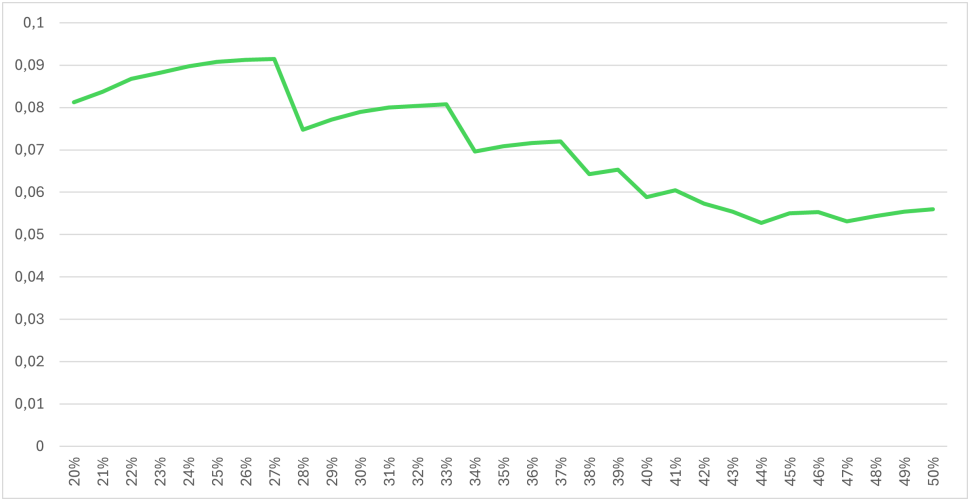
De grafieken en histogrammen van de experimenten zijn weergegeven in bijlage 7.3. De grafieken geven het vlak weer waarin alle lijnen van de gesimuleerde zorgmedewerkers in dat experiment lopen. Zo geeft de grafiek een duidelijke weergave van de spreiding van alle data. Waar in dit vlak de meeste lijnen lopen is af te leiden uit de histogram van de verdeling van de data.

De uitkomsten van de dynamische experimenten met een drempelwaarde van 20% tot en met 50% zijn te zien in Tabel 5. Deze experimenten lopen ook over een periode van vijf jaar, maar fluctueren in aantal trainingen. Een hogere drempelwaarde heeft in sommige gevallen namelijk een hoger aantal trainingen nodig om deze drempelwaarde te behalen.

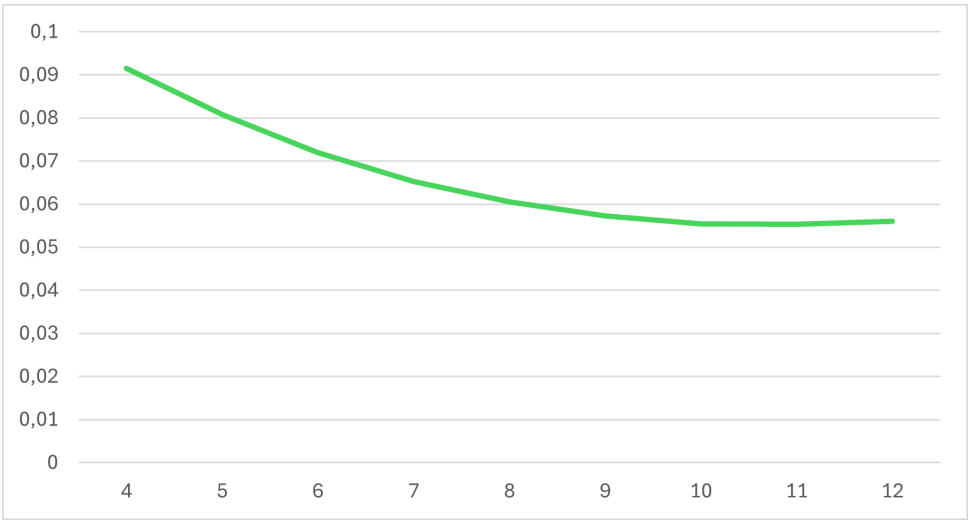
In Tabel 4 is te zien dat er twaalf experimenten zijn die een hogere AUC weergeven dan experiment 1. Experiment 7, 10 en 9 geven de drie beste resultaten weer van de vooropgestelde experimenten. Daarnaast is te zien dat, voor zowel de groep 'Laatste training:

Begin jaar 5' als de groepen 'Laatste training: Begin jaar 4' en 'Laatste training: Begin jaar 3', de afnemende frequentie betere resultaten weergeeft dan de evenredig uitgespreide experimenten. Tegenovergesteld is ook te zien dat een toenemende frequentie bij alle drie groeperingen een slechter resultaat oplevert. Verder lijkt de groepering 'Laatste training: Begin jaar 4' de beste uitkomsten te hebben. Deze groep laat namelijk de hoogste AUC zien vergeleken met soortgelijke experimenten uit elke andere groep.

Wanneer de dynamische experimenten worden vergeleken op efficiëntie door middel van de behaalde AUC gedeeld door het aantal benodigde trainingen, is te zien dat een toenemende trainingsfrequentie resulteert in een afname van efficiëntie, wat stagneert rond de 10 trainingen. Dit is te zien in Figuur 3 en Figuur 4, waar de AUC van de beste experimenten per aantal trainingen is weergegeven.



FIGUUR 3: AUC per training voor verschillende drempelwaarden



FIGUUR 4: Beste AUC per training voor verschillende trainingsaantallen binnen vijf jaar

De histogrammen van de AUCs laten overigens zien dat de data *right-skewed* is. Dit betekent dat het grootste deel van de data onder het gemiddelde ligt. De data analyse leidde ook tot een *bootstrapped* BI voor elk experiment. Deze zijn weergegeven in Tabel 4,

Tabel 4: AUC mediaan per vooraf bepaald experiment, kleurgecodeerd van hoogste score naar laagste score (groen naar rood)

Exp.	Groep	Specificering	Md.	BI	Rang
1	Laatste training: Begin jaar 5	Evenredig uitgespreid (huidige situatie)	0,380	[0.379;0.380]	13
2		Afnemende frequentie	0,395	[0.394;0.395]	5
3		Toenemende frequentie	0,349	[0.348;0.349]	17
4		Dubbele training (maand 1&4) jaar 1&2	0,382	[0.381;0.382]	11
5		Dubbele training (maand 1&7) jaar 1&2	0,387	[0.386;0.387]	9
6	Laatste training: Begin jaar 4	Evenredig uitgespreid	0,394	[0.393;0.394]	6
7		Afnemende frequentie	0,402	[0.401;0.402]	1
8		Toenemende frequentie	0,382	[0.381;0.382]	10
9		Dubbele training (maand 1&4) jaar 1&2	0,399	[0.398;0.399]	3
10		Dubbele training (maand 1&7) jaar 1&2	0,401	[0.400;0.401]	2
11	Laatste training: Begin jaar 3	Evenredig uitgespreid	0,392	[0.391;0.392]	8
12		Afnemende frequentie	0,397	[0.396;0.397]	4
13		Toenemende frequentie	0,381	[0.381;0.381]	12
14		Dubbele training (maand 1&4) jaar 1&2	0,393	[0.392;0.393]	7
15	Trainingsjaar	Uitgespreid trainingsjaar	0,373	[0.372;0.373]	14
16		Gecondenseerd trainingsjaar	0,362	[0.362;0.362]	15
17		Half trainingsjaar	0,350	[0.349;0.350]	16

waarin te zien is dat, voor elk vooraf bepaald experiment, het BI vrijwel op de mediaan ligt. Voor de dynamische experimenten is er bij een hogere drempelwaarde sprake van een lichte toename in de spreiding van het BI.

5 Discussie

De vergelijking tussen de verschillende AUCs geven inzichten in de relatie tussen de trainingsfrequentie en de geheugenretentie. In de discussie zullen de verbanden en verschillen tussen de experimenten en de groeperingen bekeken en geïnterpreteerd worden.

Het doel van de simulatie was om de huidige situatie te vergelijken met mogelijke verbeteringen. In de resultaten is te zien dat de huidige situatie, experiment 1, op rang 13 komt. Dit betekent dat er 12 andere samenstellingen zijn die beter presteren, en soms ook resulteren in een aanzienlijk hogere AUC.

Ook bleek dat de meeste potentie volgens de simulatie lag bij de groep 'Laatste training: Begin jaar 4'. De combinatie met een afnemende frequentie bleek de beste resultaten neer te zetten. Het feit dat deze groep de beste resultaten weergeeft, laat zien dat deze groep in de buurt zit van de optimale timing van trainingen.

Het succes van de afnemende frequentie en het slechte resultaat van de toenemende frequentie sluit volledig aan bij de verwachting volgens de Ebbinghaus curve (1885), aangezien deze theorie draait om het naar voren schuiven van trainingen of *boosters* voor het versterken van het langetermijn effect.

Verder is in Figuur 3 een stapsgewijze grafiek te zien. Dit komt door de stap naar een extra training, wat resulteert in een daling van de AUC per training. Daarom is in Figuur 4 de maxima van deze stappen weergegeven als lijn, waardoor het afnemend dalende effect beter zichtbaar is.

Tabel 5: AUC mediaan per dynamisch experiment

Drempelwaarde	**Aantal benodigde trainingen**	**Mediaan**	**95% BI**
20%	4	0,325	[0,325 ;0,326]
21%	4	0,335	[0.335;0.336]
22%	4	0,347	[0,346;0,347]
23%	4	0,353	[0,353;0,354]
24%	4	0,359	[0,358;0,359]
25%	4	0,363	[0,362;0,363]
26%	4	0,365	[0,365;0,366]
27%	4	0,366	[0,366;0,367]
28%	5	0,374	[0,373;0,374]
29%	5	0,386	[0,385;0,387]
30%	5	0,395	[0,394;0,396]
31%	5	0,4	[0,399;0,400]
32%	5	0,402	[0,401;0,403]
33%	5	0,404	[0,403;0,405]
34%	6	0,418	[0,418;0,420]
35%	6	0,425	[0,424;0,425]
36%	6	0,43	[0,429;0,431]
37%	6	0,432	[0,431;0,433]
38%	7	0,45	[0,448;0,451]
39%	7	0,457	[0,455;0,458]
40%	8	0,471	[0,469;0,473]
41%	8	0,484	[0,482;0,487]
42%	9	0,516	[0,513;0,520]
43%	10	0,554	[0,552;0,557]
44%	11	0,58	[0,576;0,584]
45%	11	0,605	[0,604;0,611]
46%	11	0,609	[0,609;0,614]
47%	12	0,637	[0,636;0,643]
48%	12	0,652	[0,648;0,656]
49%	12	0,665	[0,658; 0,666]
50%	12	0,672	[0,664;0,672]

5.1 Tekortkomingen van het onderzoek

Naast de inzichten die dit onderzoek biedt, zijn er ook enkele tekortkomingen. In Figuur 2 is bijvoorbeeld te zien dat er idealiter 20.000 iteraties plaats zouden vinden per experiment om compleet onder de foutmarge van 1% te blijven, maar dat er slechts 13.000 zijn uitgevoerd. Dit is gedaan met het zicht op het geheugen van de beschikbare apparatuur, aangezien er *crashes* optraden wanneer er 20.000 iteraties werden gedraaid. Idealiter zou er nog steeds 20.000 iteraties worden gedraaid, al wordt verwacht dat dit verschil weinig invloed heeft op de uiteindelijke uitkomsten aangezien er slechts een paar attributen zijn die lichtelijk boven de 1% foutmarge uitsteken.

Daarnaast is de wiskundige aanpassing van parameters ook op aannames gebasseerd. Eén voorbeeld van de tekortkomingen van de huidige transformaties is dat alle aanpassingen vermenigvuldigend zijn. Hierdoor worden de effecten, naarmate de simulatie langer duurt, uitgedoofd door elkaar. Lagere parameters zorgen voor hogere geheugenretentie, maar als een medewerker een factor van 0.9 krijgt van geslacht en een factor van 0.9 van opleiding houdt dat in dat ze niet onafhankelijk handelen, omdat ze elkaar vermenigvuldigen en dus verminderen.

5.2 Vervolgonderzoek

De voornaamste gebieden waar vervolgonderzoek naar gedaan kan worden zijn de kennis over wiskundige verdieping over parametrisatie, een interactieve weergave van de simulatie en experimenteel onderzoek om de simulatie te valideren.

Onderzoek dat dieper ingaat op de wiskundige parametrisatie van de functie is nodig om meer inzichten te krijgen in de juiste en meest realistische aanpassingen van de parameters.

Verder kan dit onderzoek gebruikt worden als een begin voor een praktischere toepassing voor zorginstellingen, zoals een dashboard of een interactieve weergave van de simulatie. Op deze manier kunnen zorginstellingen de simulatie aanpassen naar eigen omstandigheden en een aangepaste trainingsfrequentie bepalen.

Experimenteel onderzoek over langdurige geheugenretentie van cybersecuritytrainingen binnen de zorg is nodig om dit model en onderzoek te valideren. Op deze manier zou de mogelijkheid ontstaan om het model aan te passen aan de experimentele bevindingen en een nog realistischer model neer te zetten.

6 Conclusie

In dit onderzoek is via een grootschalige simulatie onderzocht hoe verschillende frequentie strategieën voor cybersecuritytrainingen de geheugenretentie van zorgmedewerkers beïnvloeden. Op basis van een aangepaste Ebbinghaus-vergeetcurve, rekening houdend met demografische en werkdrukfactoren, zijn twintig trainingsscenario's geëvalueerd. Daarbij is gekeken naar de oppervlakte onder de geheugenretentiecurve (AUC) als maat voor effectiviteit. De resultaten tonen aan dat twaalf scenario's een hogere retentie opleverden dan één jaarlijkse training aan het begin van elk jaar.

6.1 Antwoord op onderzoeksvraag

Om de onderzoeksvraag "Wat is de optimale frequentie van cybersecurity trainingen in de zorg om geheugenretentie te bevorderen bij werknemers?" te beantwoorden zijn de volgende

deelvragen opgesteld:

- *Wat is de optimale trainingsfrequentie voor de huidige hoeveelheid trainingen per jaar?*
- *Welke trainingsfrequentie is nodig om boven bepaalde drempelwaarden van geheugenretentie te blijven?*

De resultaten gaven weer dat de meest efficiënte trainingsfrequentie van alle meegenomen experimenten resulteerde in trainingen in maand 5, 17 en 35 om een drempelwaarde van 27% te behalen door middel van vier trainingen in vijf jaar. Bij meer trainingen is er een afnemende daling van efficiëntie per training.
Daarnaast is de meest optimale trainingsfrequentie van de vooraf bepaalde experimenten met vijf trainingen over vijf jaar experiment 7, waar gebruik werd gemaakt van afnemende frequentie en een toenemend interval tussen trainingen.

6.2 Aanbevelingen voor zorginstellingen

De voornaamste toepassing van dit onderzoek voor zorginstellingen ligt niet in de specifieke maanden waarin trainingen ingedeeld moeten worden, maar voornamelijk in de generieke verbanden die zijn gevonden in de groeperingen. Belangrijker is om het algemene voordeel van een afnemende frequentie en een toenemend interval met dezelfde aantal trainingen toe te passen. Zo word het bijvoobeeld aangeraden aan de hand van dit onderzoek om nieuwe medewerkers twee keer per jaar te trainen, en medewerkers die al langer dan twee jaar werken naar een jaarlijkse training over te zetten. Op deze realistische en toepasbare manier wordt de geheugenretentie gewaarborgd terwijl er efficiënter omgegaan wordt met de tijd van de drukke zorgmedewerkers.

Referentielijst

ANP Producties. (2022). Zorgminister Kuipers: Nederlandse zorg presteert middelmatig. *Noordhollands Dagblad*. https://www.noordhollandsdagblad.nl/binnenland/zorgminister-kuipers-nederlandse-zorg-presteert-middelmatig/11451258.html

Bloomberg, M., Dugravot, A., Dumurgier, J., Kivimaki, M., Fayosse, A., Steptoe, A., Britton, A., Singh-Manoux, A., & Sabia, S. (2021). Sex differences and the role of education in cognitive ageing: analysis of two UK-based prospective cohort studies. *Lancet Public Health*, *6*(2), e106–e115. https://doi.org/10.1016/S2468-2667(20)30258-9

Centraal Bureau Voor De Statistiek. (2023). Arbeidsmarktprofiel van zorg en welzijn in 2022. https://www.cbs.nl/nl-nl/longread/statistische-trends/2023/arbeidsmarktprofiel-van-zorg-en-welzijn-in-2022?onepage=true#c-4--Arbeidsomstandigheden

Cevallos-Torres, L., & Botto-Tobar, M. (2019). Monte Carlo Simulation Method. In *Problem-Based Learning: A Didactic Strategy in the Teaching of System Simulation* (pp. 87–96). https://link.springer.com/book/10.1007/978-3-030-13393-1

Chun, B. A., & Heo, H. J. (2018). The effect of flipped learning on academic performance as an innovative method for overcoming ebbinghaus' forgetting curve. *Proceedings of the 6th International Conference on Information and Education Technology*, 56–60. https://doi.org/10.1145/3178158.3178206

Ebbinghaus, H. (1885). *Über das gedächtnis. Untersuchungen zur experimentellen psychologie.*

Kruse, C. S., Frederick, B., Jacobson, T., & Monticone, D. K. (2017). Cybersecurity in healthcare: A systematic review of modern threats and trends. *Technol Health Care*, *25*(1), 1–10. https://doi.org/10.3233/THC-161263

MarbleScience. (2021). Monte Carlo Simulation. https://www.youtube.com/watch?v=7ESK5SaP-bc

Murre, J. M., & Dros, J. (2015). Replication and Analysis of Ebbinghaus' Forgetting Curve. *PLoS One*, *10*(7), e0120644. https://doi.org/10.1371/journal.pone.0120644

Nationaal Coördinator Terrorismebestrijding en Veiligheid. (2024). *Cybersecuritybeeld Nederland 2024* (tech. rap.). Ministerie van Justitie en Veiligheid. https://www.nctv.nl/onderwerpen/cybersecuritybeeld-nederland/documenten/publicaties/2024/10/28/cybersecuritybeeld-nederland-2024

Pliatsikas, C., Verissimo, J., Babcock, L., Pullman, M. Y., Glei, D. A., Weinstein, M., Goldman, N., & Ullman, M. T. (2019). Working memory in older adults declines with age, but is modulated by sex and education. *Q J Exp Psychol (Hove)*, *72*(6), 1308–1327. https://doi.org/10.1177/1747021818791994

Rijksinstituut voor Volksgezondheid en Milieu. (2022). Gebruik digitale zorg neemt toe maar is nog lang niet altijd effectief. https://www.rivm.nl/nieuws/gebruik-digitale-zorg-neemt-toe-maar-is-nog-lang-niet-altijd-effectief

Schoofs, D., Preuss, D., & Wolf, O. T. (2008). Psychosocial stress induces working memory impairments in an n-back paradigm. *Psychoneuroendocrinology*, *33*(5), 643–53. https://doi.org/10.1016/j.psyneuen.2008.02.004

Virtanen, M., Singh-Manoux, A., Ferrie, J. E., Gimeno, D., Marmot, M. G., Elovainio, M., Jokela, M., Vahtera, J., & Kivimaki, M. (2009). Long working hours and cognitive function: the Whitehall II Study. *Am J Epidemiol*, *169*(5), 596–605. https://doi.org/10.1093/aje/kwn382

Zorgbord. (2023). De toekomst van de gezondheidszorg: digitalisering in de zorg. https://www.zorgbord.nl/digitalisering-in-de-zorg/

7 Bijlage

7.1 Pseudocode

7.1.1 Medewerker genererende functie

def *generate_healthcare_worker*

r = willekeurig getal tussen de 0 en 100
Geslacht van medewerker is man als r groter of gelijk is aan 15.9,
anders is geslacht van medewerker vrouw

r = nieuw willekeurig getal tussen de 0 en 100
leeftijd van medewerker is jonger dan 35 als r kleiner of gelijk is aan 35.1,
anders is leeftijd van medewerker 35 tot 55 als r kleiner of gelijk is aan 75.2,
anders is leeftijd van medewerker 55 of ouder

r = nieuw willekeurig getal tussen de 0 en 100
opleiding van medewerker is laag als r kleiner of gelijk is aan 11.6,
anders is opleiding van medewerker middelbaar als r kleiner of gelijk is aan 54.9,
anders is opleiding van medewerker hoog als r kleiner of gelijk is aan 97.9,
anders is opleiding van medewerker onbekend

r = nieuw willekeurig getal tussen de 0 en 100
arbeidsduur van medewerker is <12 uur als r kleiner of gelijk is aan 6.8,
anders is arbeidsduur van medewerker 12 tot 20 uur als r kleiner of gelijk is aan 16.8,
anders is arbeidsduur van medewerker 20 tot 35 uur als r kleiner of gelijk is aan 72.8,
anders is arbeidsduur van medewerker 35 tot 41 uur als r kleiner of gelijk is aan 93.5,
anders is arbeidsduur van medewerker >40 uur

r = nieuw willekeurig getal tussen de 0 en 100
werkdruk van medewerker is Te hoog als r kleiner of gelijk is aan 41.7,
anders is werkdruk van medewerker Niet te hoog

r = nieuw willekeurig getal tussen de 0 en 100
Werktevredenheid van medewerker is Tevreden als r kleiner of gelijk is aan 76.1,
anders is werktevredenheid van medewerker Niet tevreden

Geheugenretentie van medewerker is 1.0
Laatste trainingsdag van medewerker is 0

Geef medewerker

7.1.2 Geheugenretentie functie

```
def Q(t, mu1, mu2, a1, a2):
t_minuten = t * 60
term1 = mu1 * exp(-a1 * t_minuten)
term2_teller = mu1 * mu2 * (exp(-a2 * t_minuten) - exp(-a1 * t_minuten))
term2_noemer = a1 - a2
term2 = term2_teller / term2_noemer
geef term1 + term2
```

7.1.3 Parameter functie

def get_retention_parameters(worker, training_number = 1)
mu1 = 0.619
mu2 = 0.000174
a1 = 0.00035625
a2 = 0.00000232275

als de leeftijd van de medewerker gelijk is aan "jonger dan 35", dan:
mu2 = mu2 * 0.9
a1 = a1 * 0.9
a2 = a2 * 0.9
als de leeftijd van de medewerker gelijk is aan "35 tot 55", dan:
mu2 = mu2 * 1.0
a1 = a1 * 1.0
a2 = a2 * 1.0
als de leeftijd van de medewerker gelijk is aan "55 of ouder", dan:
mu2 = mu2 * 1.1
a1 = a1 * 1.1
a2 = a2 * 1.1

als het geslacht van de medewerker gelijk is aan "Vrouw", dan:
mu2 = mu2 * 1.0
a1 = a1 * 1.1
a2 = a2 * 0.8
als het geslacht van de medewerker gelijk is aan "Man", dan:
mu2 = mu2 * 1.0
a1 = a1 * 1.0
a2 = a2 * 1.0

als de opleiding van de medewerker gelijk is aan "Hoog", dan:
mu2 = mu2 * 0.85
a1 = a1 * 0.85
a2 = a2 * 0.85
als de opleiding van de medewerker gelijk is aan "Middelbaar", dan:
mu2 = mu2 * 1.0
a1 = a1 * 1.0
a2 = a2 * 1.0
als de opleiding van de medewerker gelijk is aan "Laag", dan:
mu2 = mu2 * 1.15
a1 = a1 * 1.15
a2 = a2 * 1.15
als de opleiding van de medewerker gelijk is aan "Onbekend", dan:
mu2 = mu2 * 1.0
a1 = a1 * 1.0
a2 = a2 * 1.0

als de arbeidsduur van de medewerker gelijk is aan "<12 uur", dan:
mu2 = mu2 * 0.9
a1 = a1 * 0.9
a2 = a2 * 1.0

als de arbeidsduur van de medewerker gelijk is aan "12 tot 20 uur", dan:
mu2 = mu2 * 0.95
a1 = a1 * 0.95
a2 = a2 * 1.0
als de arbeidsduur van de medewerker gelijk is aan "20 tot 35 uur", dan:
mu2 = mu2 * 1.0
a1 = a1 * 1.0
a2 = a2 * 1.0
als de arbeidsduur van de medewerker gelijk is aan "35 tot 41 uur", dan:
mu2 = mu2 * 1.05
a1 = a1 * 1.05
a2 = a2 * 1.0
als de arbeidsduur van de medewerker gelijk is aan ">40 uur", dan:
mu2 = mu2 * 1.1
a1 = a1 * 1.1
a2 = a2 * 1.0

als de leeftijd van de medewerker gelijk is aan "55 of ouder", en geslacht aan "Man", dan:
mu2 = mu2 * 1.05
a1 = a1 * 1.05
a2 = a2 * 1.05
als de opleiding van de medewerker gelijk is aan "Hoog", en geslacht aan "Vrouw", dan:
mu2 = mu2 * 0.8
a1 = a1 * 0.8
a2 = a2 * 0.7

7.1.4 Vooraf bepaalde trainingen plotten

Thesis_main.py

Zet een *"random seed"*

maak een lijst met 13.000 medewerkers door middel van de generate_healthcare_worker functie

zet totale_maanden op 60
geef trainingsmaanden aan (verschilt per experiment)

definieer functie compute_retention_curve:
Slaat de begin en eindmomenten op van alle trainingscurves in t_waarden

bepaal mu1, mu2, a1 en a2 d.m.v. get_retention_parameters
bepaal het segment(individuele curve) d.m.v. functie Q() met alle nieuwe parameters

definieer num_workers als aantal medewerkers (in dit geval 13.000)

Loop door elke medewerker in je lijst en bepaalt de bijbehorende curve d.m.v. compute_retention_curve en voeg de curve toe aan all_curves

bepaal de minimale waarde van alle curves op elke waarde van t
bepaal de maximale waarde van alle curves op elke waarde vna t

plot een grafiek over 5 jaar met een opgevuld vlak tussen de maximale en minimale curve

Loop door elke curve in all_curves en bereken de AUC. Sla elke AUC op in de lijst aucs

Standaardiseer de AUC door te delen door de totale periode waarover gesimuleerd wordt (5 jaar)

Bootstrap de mediaan met 10.000 iteraties

Plot een histogram met de mediaan van alle AUCs en de bootstrapped betrouwbaarheids-interval

als het aantal gevolgde trainingen van de medewerker gelijk is aan 1, dan:
afname_factor = 1
als het aantal gevolgde trainingen van de medewerker gelijk is aan 2, dan:
afname_factor = 0.44
als het aantal gevolgde trainingen van de medewerker gelijk is aan 3 of meer, dan:
afname_factor = 0.44 * (0.66 ** (aantal trainingen - 2)

7.1.5 Dynamische trainingen plotten

Thesis_dynamic_schedule.py

Bepaal totaal aantal maanden (60)
Bepaal drempelwaarde (verschilt per experiment)
Bepaal aantal medewerkers (5000)

definieer functie compute_retention_curve:
Slaat de begin en eindmomenten op van alle trainingscurves in t_waarden

bepaal mu1, mu2, a1 en a2 d.m.v. get_retention_parameters
bepaal het segment(individuele curve) d.m.v. functie Q() met alle nieuwe parameters

bepaal training_months als lijst met alleen de eerste training op t=0

Zolang er niet wordt onderbroken:
Loop door elke medewerker in je lijst en bepaalt de bijbehorende curve d.m.v. compute_retention_curve

Loop door alle dagen heen
als de mediaan onder de drempelwaarde valt wordt de volgende maand toegevoegd aan de lijst met trainingen
breek uit *loop*

als er geen nieuwe training wordt gevonden voor het einde van de periode:
breek uit de *loop*

bepaal de minimale waarde van alle curves op elke waarde van t
bepaal de maximale waarde van alle curves op elke waarde vna t

plot een grafiek over 5 jaar met een opgevuld vlak tussen de maximale en minimale curve

Loop door elke curve in all_curves en bereken de AUC. Sla elke AUC op in de lijst aucs

Standaardiseer de AUC door te delen door de totale periode waarover gesimuleerd wordt (5 jaar)

Bootstrap de mediaan met 10.000 iteraties

Plot een histogram met de mediaan van alle AUCs en de bootstrapped betrouwbaarheids-interval

7.2 Python code

7.2.1 Medewerker genererende functie

```python
import random

def generate_healthcare_worker():
    worker = {}

    # Geslacht (Gender)
    r = random.uniform(0, 100)
    worker['geslacht'] = "Man" if r <= 15.9 else "Vrouw"

    # Leeftijd (Age)
    r = random.uniform(0, 100)
    if r <= 35.1:
        worker['leeftijd'] = "Jonger dan 35"
    elif r <= 75.2:
        worker['leeftijd'] = "35 tot 55"
    else:
        worker['leeftijd'] = "55 of ouder"

    # Opleiding (Education)
    r = random.uniform(0, 100)
    if r <= 11.6:
        worker['opleiding'] = "Laag"
    elif r <= 54.9:
        worker['opleiding'] = "Middelbaar"
    elif r <= 97.9:
        worker['opleiding'] = "Hoog"
    else:
        worker['opleiding'] = "Onbekend"

    # Arbeidsduur (Work hours)
    r = random.uniform(0, 100)
    if r <= 6.8:
        worker['arbeidsduur'] = "<12 uur"
    elif r <= 16.8:
        worker['arbeidsduur'] = "12 tot 20 uur"
    elif r <= 72.8:
        worker['arbeidsduur'] = "20 tot 35 uur"
    elif r <= 93.5:
        worker['arbeidsduur'] = "35 tot 41 uur"
    else:
        worker['arbeidsduur'] = ">40 uur"

    # Werkdruk (Workload)
    r = random.uniform(0, 100)
    worker['werkdruk'] = "Te hoog" if r <= 41.7 else "Niet te hoog"

    # Werktevredenheid (Job satisfaction)
    r = random.uniform(0, 100)
    worker['werktevredenheid'] = "Tevreden" if r <= 76.1 else "Niet
    tevreden"

    worker['MemoryRetention'] = 1.0
    worker['LastTrainingDay'] = 0

```

```
    return worker
```

7.2.2 Geheugenretentie functie

```
import numpy as np

def Q(t, mu1, mu2, a1, a2):
    t_minutes = t * 60  # Convert hours to minutes
    term1 = mu1 * np.exp(-a1 * t_minutes)
    term2_numerator = mu1 * mu2 * (np.exp(-a2 * t_minutes) - np.exp(-a1 *
    t_minutes))
    term2_denominator = a1 - a2
    term2 = term2_numerator / term2_denominator
    return term1 + term2
```

7.2.3 Parameter functie

```
import numpy as np

def get_retention_parameters(worker, training_number = 1):
    mu1 = 0.619
    mu2 = 0.000174
    a1 = 0.00035625
    a2 = 0.00000232275

    # Adjust based on age
    if worker["leeftijd"] == "Jonger dan 35":
        mu2 *= 0.9 #rate of STM -> LTM
        a1 *= 0.9 #decay of STM
        a2 *= 0.95 #decay of LTM
    elif worker["leeftijd"] == "35 tot 55":
        mu2 *= 1.0
        a1 *= 1.0
        a2 *= 1.0
    elif worker["leeftijd"] == "55 of ouder":
        mu2 *= 1.1
        a1 *= 1.1
        a2 *= 1.05

    # Adjust based on gender
    if worker["geslacht"] == "Vrouw":
        mu2 *= 1
        a1 *= 1.1
        a2 *= 0.95
    elif worker["geslacht"] == "Man":
        mu2 *= 1
        a1 *= 1
        a2 *= 1

    # Adjust based on education
    if worker["opleiding"] == "Hoog":
        mu2 *= 0.85
        a1 *= 0.85
        a2 *= 0.95
    elif worker["opleiding"] == "Middelbaar":
        mu2 *= 1.0
        a1 *= 1.0
        a2 *= 1.0
    elif worker["opleiding"] == "Laag":
        mu2 *= 1.15
```

```
        a1 *= 1.15
        a2 *= 1.05
    elif worker["opleiding"] == "Onbekend":
        mu2 *= 1
        a1 *= 1
        a2 *= 1

    # Adjust based on work hours
    if worker["arbeidsduur"] == "<12 uur":
        mu2*= 0.9
        a1 *= 0.9
        a2 *= 1
    elif worker["arbeidsduur"] == "12 tot 20 uur":
        mu2 *= 0.95
        a1 *= 0.95
        a2 *= 1
    elif worker["arbeidsduur"] == "20 tot 35 uur":
        mu2 *= 1
        a1 *= 1
        a2 *= 1
    elif worker["arbeidsduur"] == "35 tot 41 uur":
        mu2 *= 1.05
        a1 *= 1.05
        a2 *= 1
    elif worker["arbeidsduur"] == ">40 uur":
        mu2 *= 1.1
        a1 *= 1.1
        a2 *= 1

    # Interaction Effects
    if worker["leeftijd"] == "55 of ouder" and worker["geslacht"] == "Man":
        mu2 *= 1.2
        a1 *= 1.2
        a2 *= 1.05
    elif worker["opleiding"] == "Hoog" and worker["geslacht"] == "Vrouw":
        mu2 *= 0.8
        a1 *= 0.8
        a2 *= 0.95

    # Adjust based on training number
    if training_number == 1:
        decay_factor = 1.0  # initial training
    elif training_number == 2:
        decay_factor = 0.44  # first booster
    else:
        decay_factor = 0.44 * (0.66 ** (training_number - 2))
    mu2 *= decay_factor
    a1 *= decay_factor
    a2 *= decay_factor
    return mu1, mu2, a1, a2
```

7.2.4 Script voor vooraf bepaalde trainingen

```
import math
import os
import numpy as np
import random
from collections import defaultdict
import matplotlib.pyplot as plt
from Thesis_func_ZMWprofiel import generate_healthcare_worker
```

```python
from Thesis_UpdateMemory import Q
from Thesis_UpdateMemory import get_retention_parameters

# --- Simulation ---

# Set random seed for reproducibility
np.random.seed(23)
random.seed(23)

output_dir = r"C:\Users\maxpo\Desktop\Thesis Python\figuren"

healthcare_workers = {}  # Initialize empty dictionary

for i in range(1, 13001):
    healthcare_workers[i] = generate_healthcare_worker()

##### Plot setup #####
total_months = 60  # 5 years
training_months = [0, 1, 2, 3, 4]
retention_curve = []

def compute_retention_curve(worker, training_months, total_months):
    retention_curve = []

    for i in range(len(training_months)):
        start_month = training_months[i]
        end_month = training_months[i + 1] if i < len(training_months) - 1 else total_months
        duration_months = end_month - start_month
        t_values = np.arange(duration_months * (365/12) * 24)

        mu1, mu2, a1, a2 = get_retention_parameters(worker, i + 1)
        segment = Q(t_values, mu1, mu2, a1, a2)
        segment /= segment[0]  # Normalize to 1

        retention_curve.extend(segment)

    return np.array(retention_curve)

num_workers = 13000
all_curves = []

for j in range(1, num_workers+1):
    worker = healthcare_workers[j]
    curve = compute_retention_curve(worker, training_months, total_months)
    all_curves.append(curve)



# Convert to NumPy array
curves_array = np.array(all_curves)

# X-axis: convert from hours to years
x_years = np.arange(curves_array.shape[1]) / (24 * 365)

# Compute min and max at each time step
min_curve = np.min(curves_array, axis=0)
max_curve = np.max(curves_array, axis=0)
median_curve = np.median(curves_array, axis=0)

# X-axis
```

```
x = np.arange(curves_array.shape[1]) / (24 * 365)  # Converts to years

# Plot
plt.figure(figsize=(12, 6))
plt.fill_between(x, min_curve, max_curve, color='skyblue', alpha=0.4, label
    ="Min-Max Range")
plt.plot(x_years, median_curve, color='black', linewidth=2, label='Mediaan'
    )
plt.title("Geheugenretentie van 13.000 Zorgwerkers over 5 Jaar")
plt.xlabel("Jaren")
plt.ylabel("Geheugenretentie (%)")
plt.ylim(0, 1)
plt.grid(True)
plt.tight_layout()
plt.savefig(os.path.join(output_dir, "grafiek_exp_17.png"))
plt.show()

# Calculate AUC
aucs = []
standardized_aucs = []

for curve in all_curves:
    if len(curve) == 0:
        continue
    auc = np.trapz(curve)
    aucs.append(auc)

    max_auc = len(curve)  # Maximum AUC is the length of the curve
    standardized_auc = auc / max_auc
    standardized_aucs.append(standardized_auc)
    #print(max_auc, auc, standardized_auc)


q1 = np.percentile(standardized_aucs, 25)
q3 = np.percentile(standardized_aucs, 75)

# Histogram
plt.figure(figsize=(10, 6))
plt.hist(standardized_aucs, bins=8, color='skyblue', edgecolor='black',
    alpha=0.7)

# Median line
median_auc = np.median(standardized_aucs)
plt.axvline(median_auc, color='red', linestyle='--', linewidth=2, label=f'
    Median = {median_auc:.3f}')

# Bootstrap for 95% CI of the median
bootstrap_samples = 10000
boot_medians = [
    np.median(np.random.choice(standardized_aucs, size=len(
    standardized_aucs), replace=True))
    for _ in range(bootstrap_samples)
]
ci_lower = np.percentile(boot_medians, 2.5)
ci_upper = np.percentile(boot_medians, 97.5)

# CI lines
plt.axvline(ci_lower, color='green', linestyle=':', linewidth=2, label=f'
    95% BI = [{ci_lower:.3f}, {ci_upper:.3f}]')
plt.axvline(ci_upper, color='green', linestyle=':', linewidth=2)

```

```
plt.axvline(q1, color='orange', linestyle=':', linewidth=2, label=f'Q1 (25%) = {q1:.3f}')
plt.axvline(q3, color='orange', linestyle=':', linewidth=2, label=f'Q3 (75%) = {q3:.3f}')

# Aesthetics
plt.title("Histogram van gestandaardiseerde AUCs met 95% Bootstrapped BI (Mediaan)")
plt.xlabel("Gestandaardiseerde AUC")
plt.ylabel("Frequentie")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig(os.path.join(output_dir, "histogram_exp_17.png"))
plt.show()
```

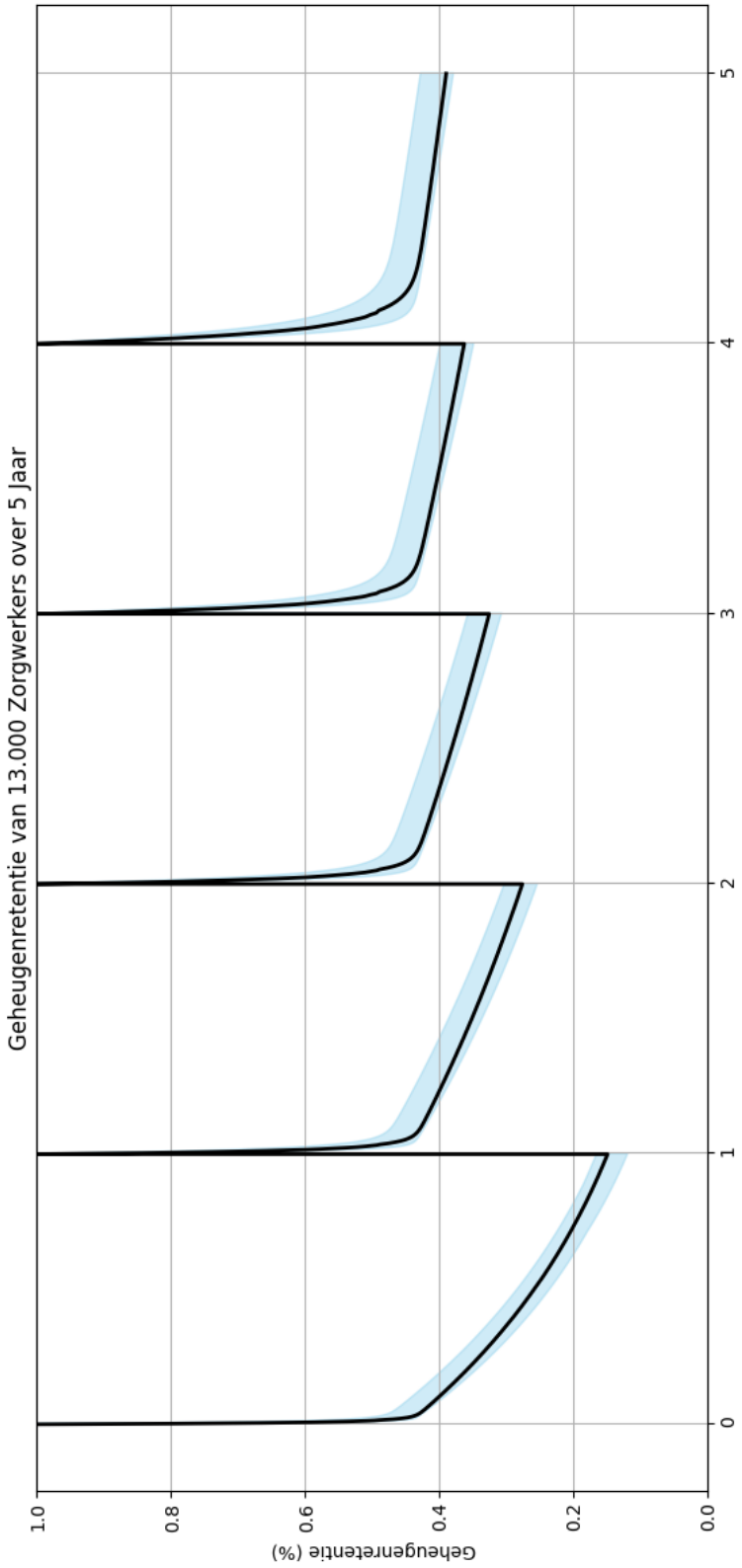
7.2.5 Script voor dynamische trainingen

```
import numpy as np
import matplotlib.pyplot as plt
import random
from Thesis_UpdateMemory import Q, get_retention_parameters
from Thesis_func_ZMWprofiel import generate_healthcare_worker
import os
from pathlib import Path

# Seed for reproducibility
np.random.seed(23)
random.seed(23)

output_dir = r"C:\Users\maxpo\Desktop\Thesis Python\figuren"

# Generate worker population
healthcare_workers = {}
for i in range(1, 5001):
    healthcare_workers[i] = generate_healthcare_worker()

# Parameters
total_months = 60  # 5 years
threshold = 0.35
days_per_month = 365 * 24 / 12
total_days = int(total_months * days_per_month)
num_workers = 5000

# Functions
def compute_retention_curve(worker, training_months, total_months):
    retention_curve = []

    for i in range(len(training_months)):
        start_month = training_months[i]
        end_month = training_months[i + 1] if i < len(training_months) - 1 else total_months
        duration_days = int((end_month - start_month) * days_per_month)
        t_values = np.arange(duration_days)  # in days

        mu1, mu2, a1, a2 = get_retention_parameters(worker, i + 1)
        segment = Q(t_values, mu1, mu2, a1, a2)
        segment /= segment[0]  # Normalize

        retention_curve.extend(segment)
```

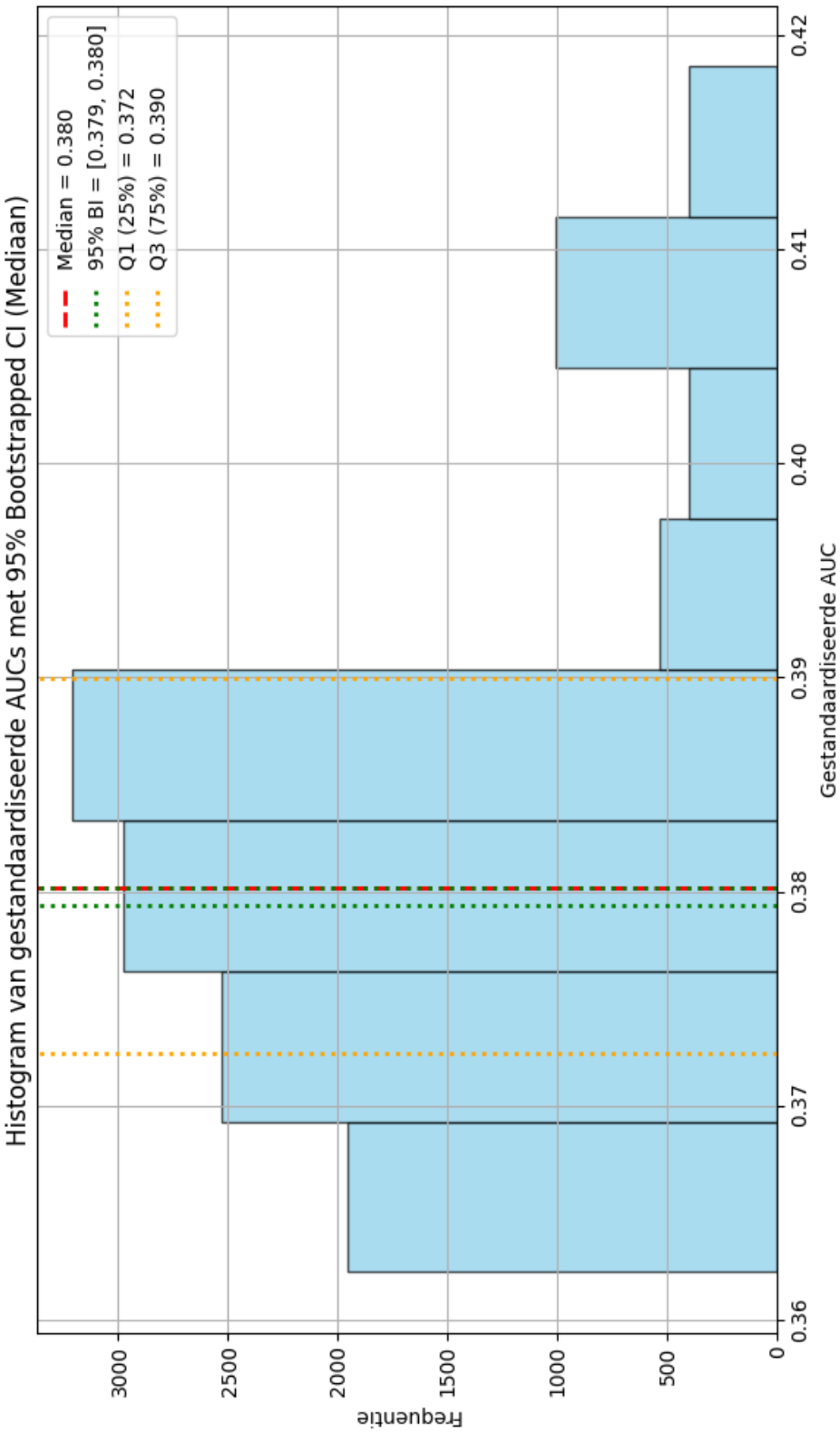
```
42
43      return np.array(retention_curve)
44
45  # Simulation loop
46  training_months = [0]
47  curves_array = None
48  next_month = None
49
50  while True:
51      # Recompute curves with current training schedule
52      all_curves = []
53      for j in range(1, num_workers + 1):
54          worker = healthcare_workers[j]
55          curve = compute_retention_curve(worker, training_months,
        total_months)
56          all_curves.append(curve)
57
58      # Convert to array
59      curves_array = np.array(all_curves)
60      current_days = curves_array.shape[1]
61      current_month = training_months[-1]
62
63      # Track if we found a new training month
64      found_new_training = False
65
66      # Scan for next threshold drop
67      for day in range(current_days):
68          month = day / days_per_month
69          if month <= current_month:
70              continue
71
72          median_ret = np.median(curves_array[:, day])
73          if median_ret < threshold:
74              next_month = int(month)
75              if next_month > current_month and next_month < total_months:
76                  training_months.append(next_month)
77                  print(f"Training scheduled at month {next_month} (day {day
        })")
78                  found_new_training = True
79                  break
80
81      if not found_new_training:
82          print("No new training needed, exiting loop.")
83          break
84
85  # Final plotting
86  x_years = np.arange(curves_array.shape[1]) / 365 / 24  # X-axis in years
87  min_curve = np.min(curves_array, axis=0)
88  max_curve = np.max(curves_array, axis=0)
89  median_curve = np.median(curves_array, axis=0)
90
91  plt.figure(figsize=(12, 6))
92  plt.fill_between(x_years, min_curve, max_curve, color='skyblue', alpha=0.4,
        label="Min-Max Range")
93  plt.plot(x_years, median_curve, color='black', linewidth=2, label='Mediaan'
        )
94
95  plt.title("Geheugenretentie van 5.000 Zorgwerkers over 5 Jaar (Met drempel-
        gebaseerde trainingen)")
96  plt.xlabel("Jaren")
97  plt.ylabel("Geheugenretentie")
```

```
plt.grid(True)
plt.tight_layout()
plt.ylim(0, 1)
plt.savefig(os.path.join(output_dir, "grafiek_dyn_35.png"))
plt.show()

# AUCs
aucs = []
standardized_aucs = []

for curve in all_curves:
    auc = np.trapz(curve)
    aucs.append(auc)
    standardized_auc = auc / len(curve)
    standardized_aucs.append(standardized_auc)

# Histogram
plt.figure(figsize=(10, 6))
plt.hist(standardized_aucs, bins=8, color='skyblue', edgecolor='black',
    alpha=0.7)

median_auc = np.median(standardized_aucs)
q1 = np.percentile(standardized_aucs, 25)
q3 = np.percentile(standardized_aucs, 75)

plt.axvline(median_auc, color='red', linestyle='--', linewidth=2, label=f'
    Mediaan = {median_auc:.3f}')

# Bootstrapping
boot_medians = [
    np.median(np.random.choice(standardized_aucs, size=len(
    standardized_aucs), replace=True))
    for _ in range(10000)
]
ci_lower = np.percentile(boot_medians, 2.5)
ci_upper = np.percentile(boot_medians, 97.5)
plt.axvline(ci_lower, color='green', linestyle=':', linewidth=2, label=f'
    95% BI = [{ci_lower:.3f}, {ci_upper:.3f}]')
plt.axvline(ci_upper, color='green', linestyle=':', linewidth=2)

plt.axvline(q1, color='orange', linestyle=':', linewidth=2, label=f'Q1
    (25%) = {q1:.3f}')
plt.axvline(q3, color='orange', linestyle=':', linewidth=2, label=f'Q3
    (75%) = {q3:.3f}')

plt.title("Histogram van gestandaardiseerde AUCs met 95% Bootstrapped BI")
plt.xlabel("Gestandaardiseerde AUC")
plt.ylabel("Frequentie")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig(os.path.join(output_dir, "histogram_dyn_35.png"))
plt.show()

# Summary
print(f"Trainings ingepland op maanden: {training_months}")
print(f"Gestandaardiseerde mediaan AUC: {median_auc:.3f}")
print(f"95% BI: [{ci_lower:.3f}, {ci_upper:.3f}]")
print("Eerste 5 gestandaardiseerde AUCs:", standardized_aucs[:5])
print("Eerste 5 bootstrap-medians:", boot_medians[:5])
```

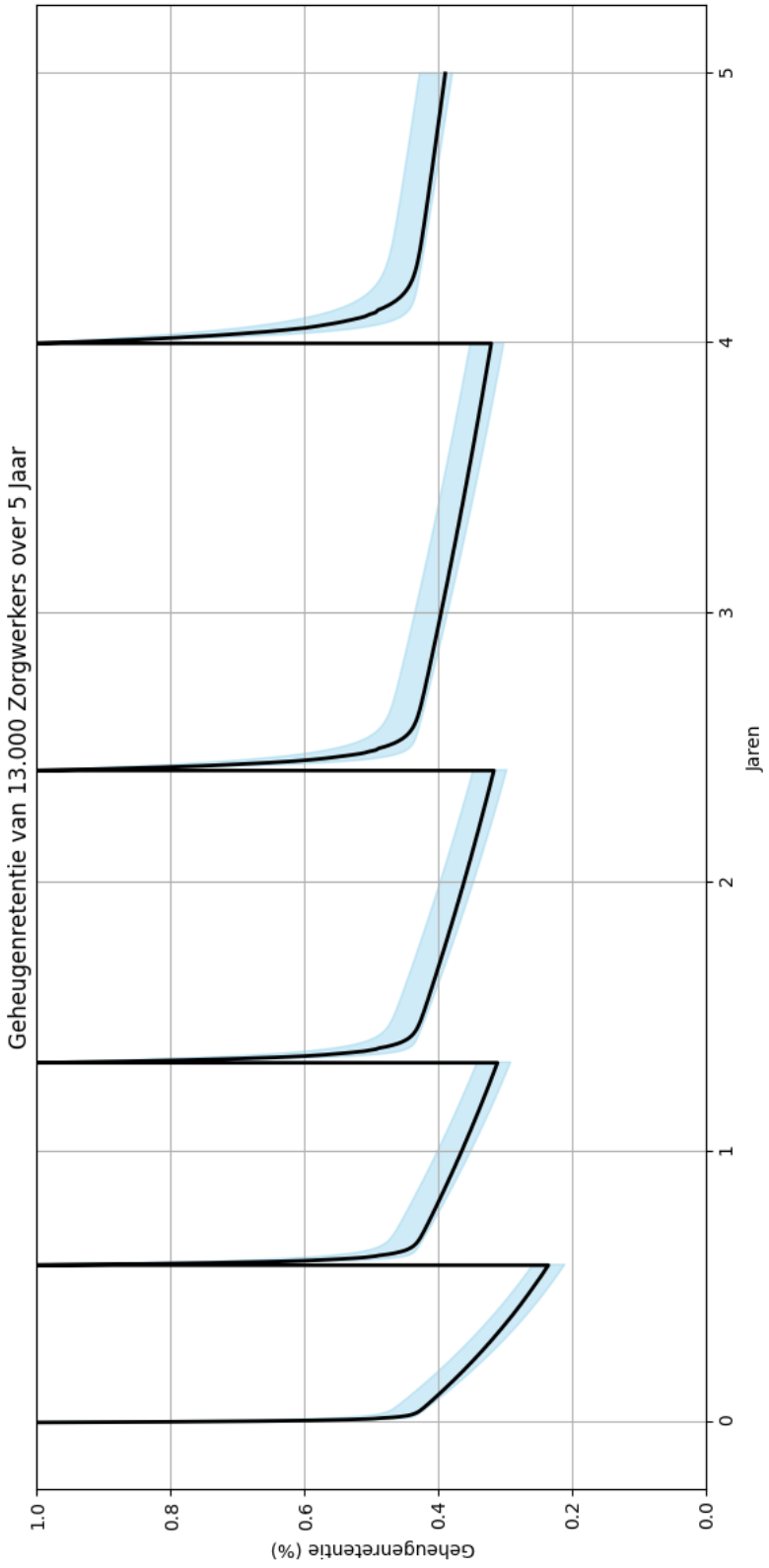
7.3 Grafieken en histogrammen van de experimenten



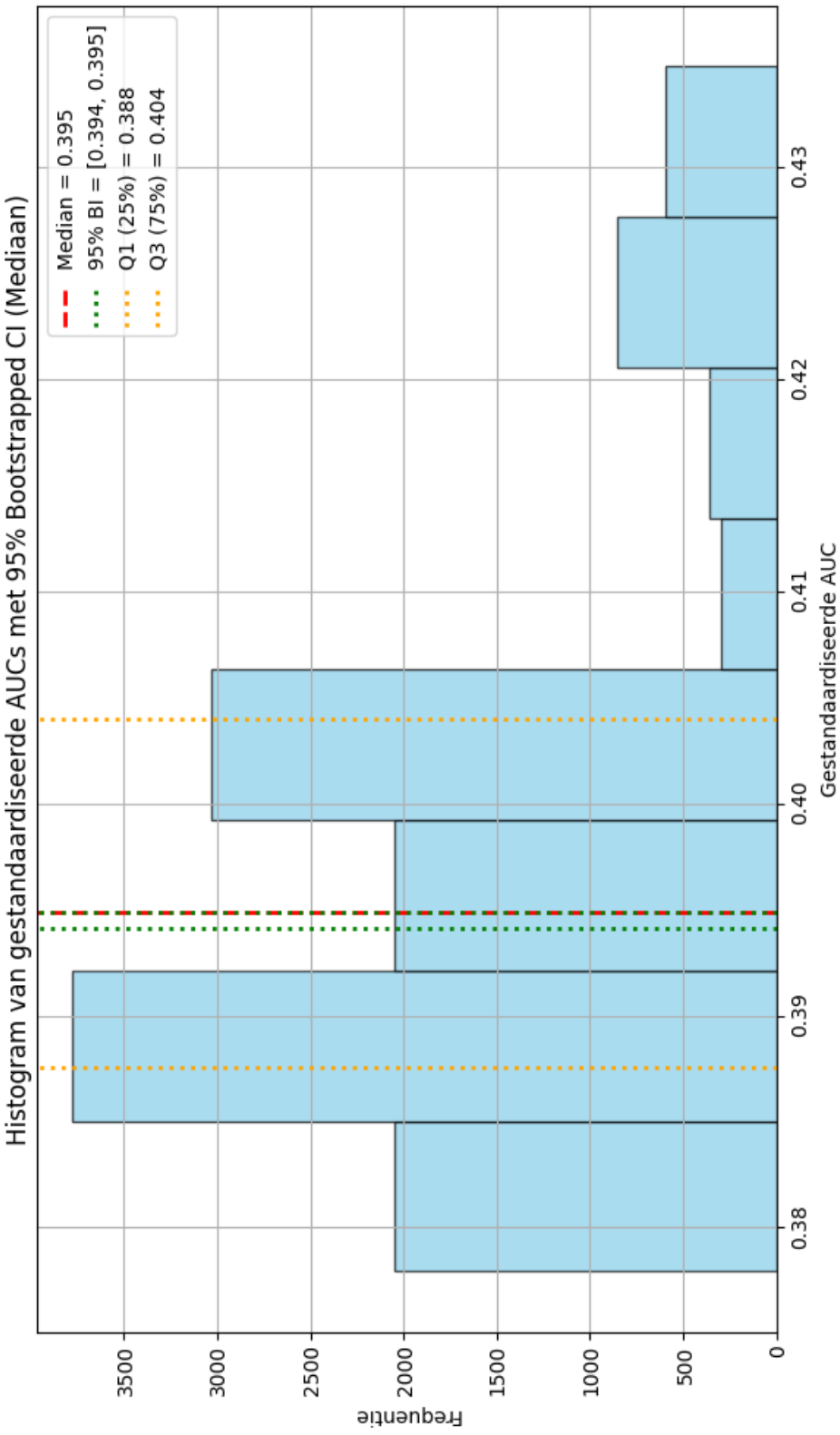
FIGUUR 5: Grafiek van experiment 1



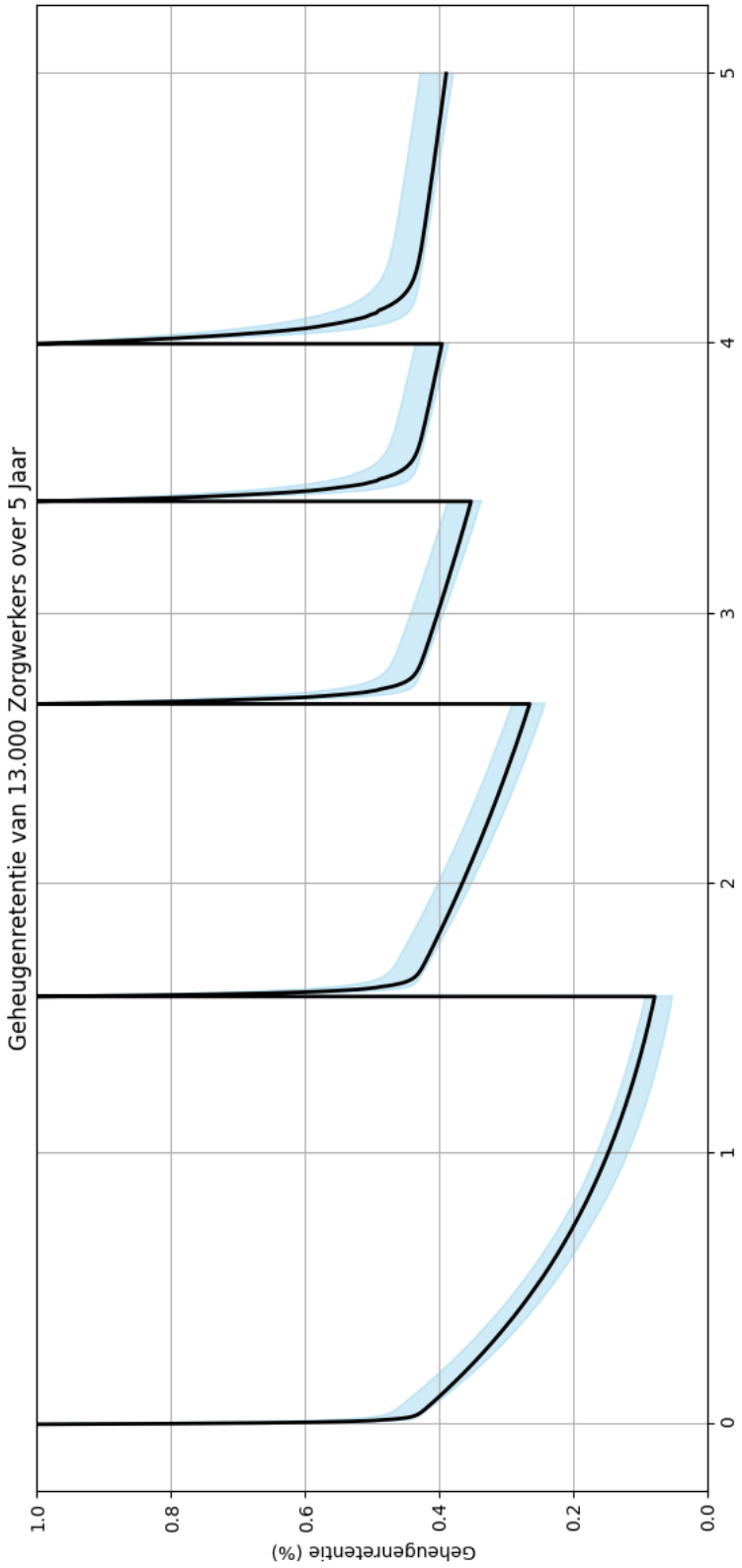
FIGUUR 6: Histogram van experiment 1



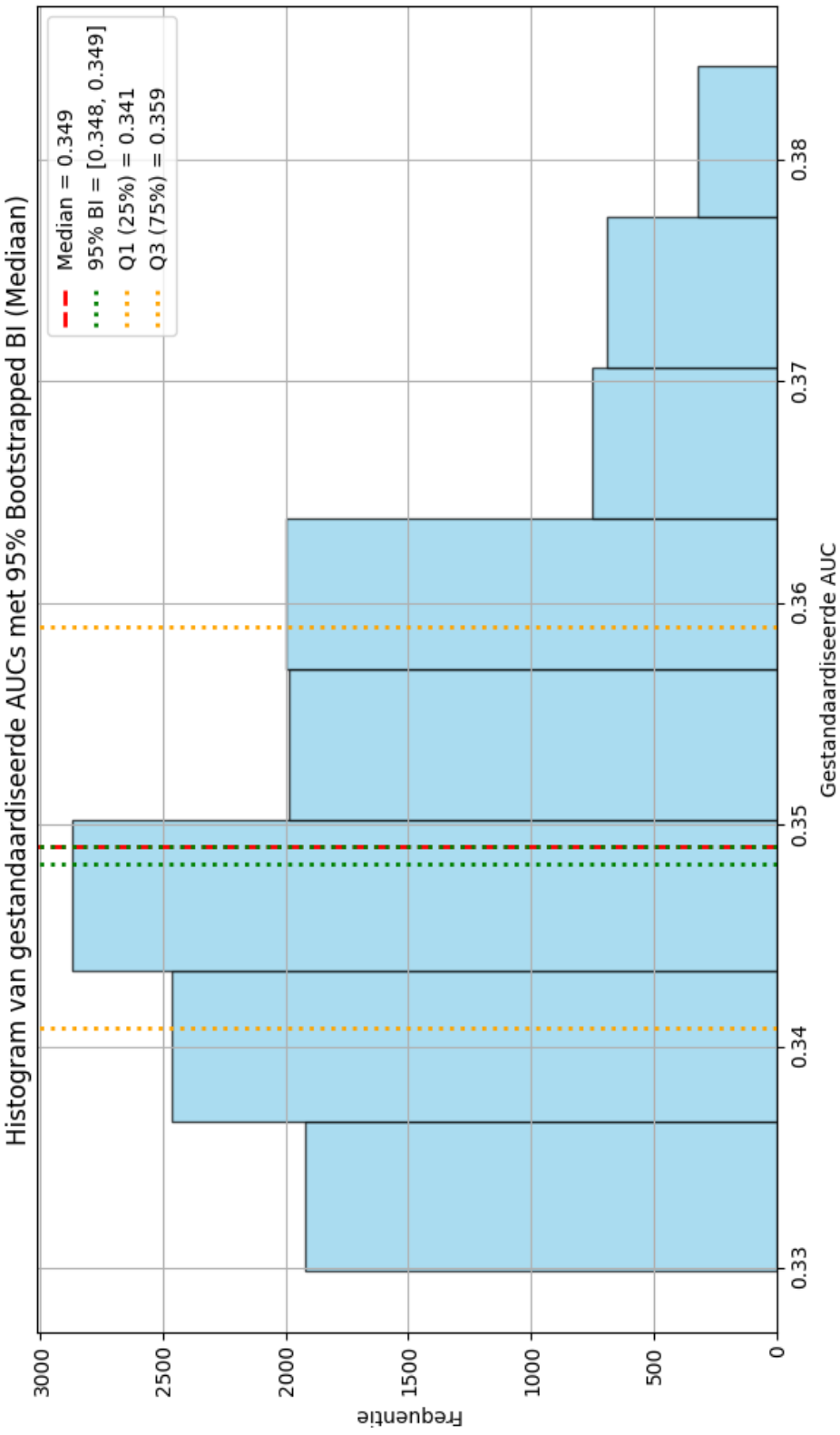
FIGUUR 7: Grafiek van experiment 2



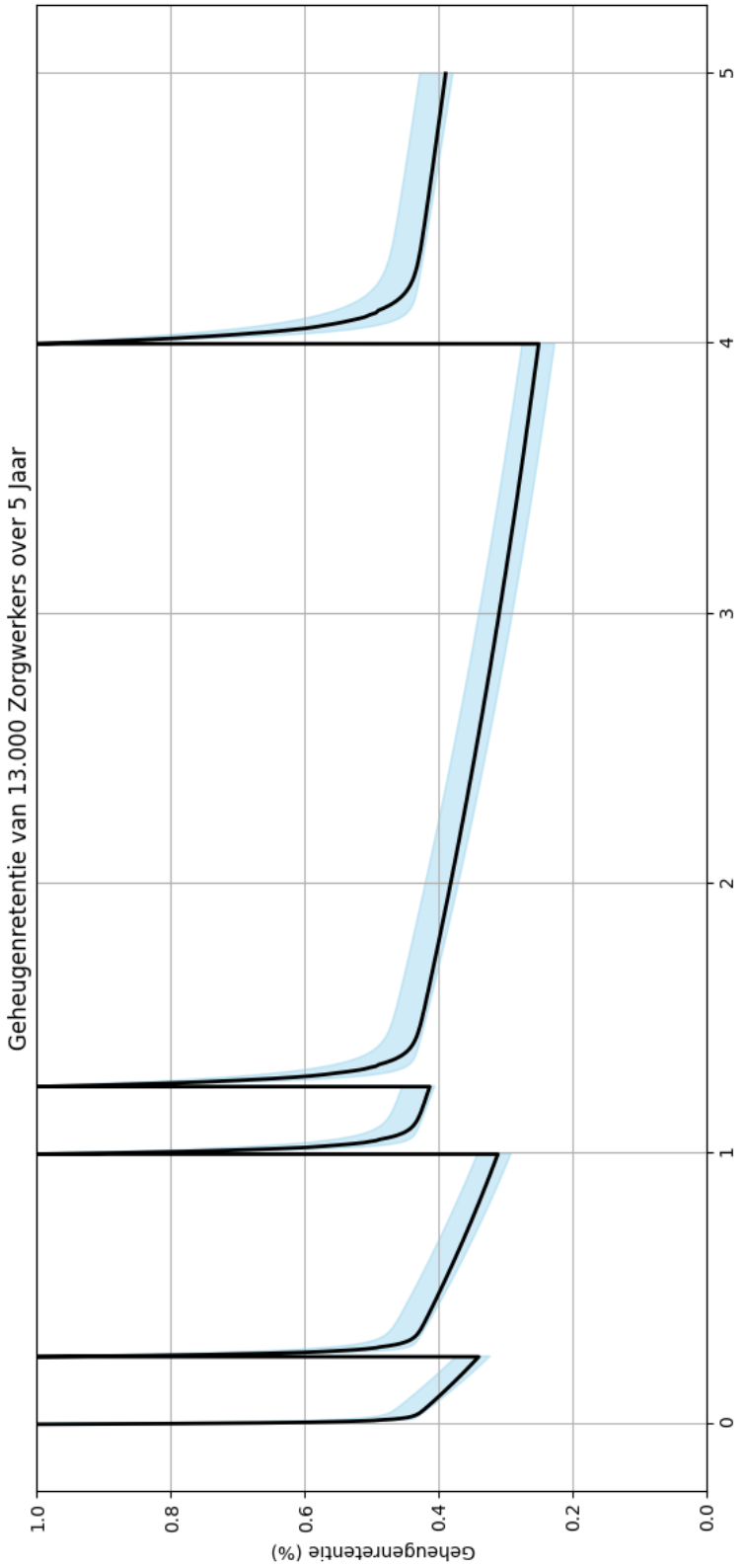
FIGUUR 8: Histogram van experiment 2



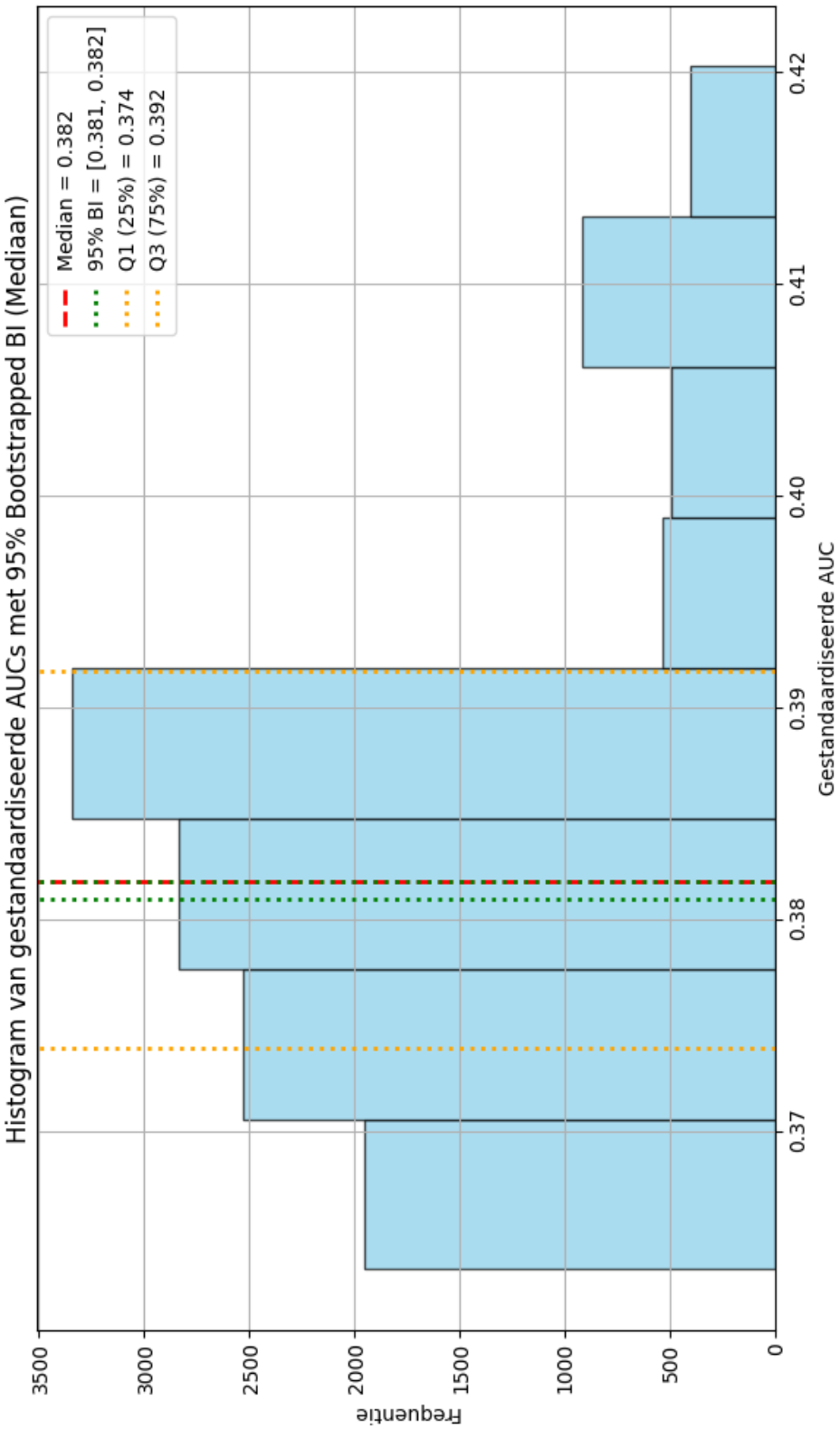
FIGUUR 9: Grafiek van experiment 3



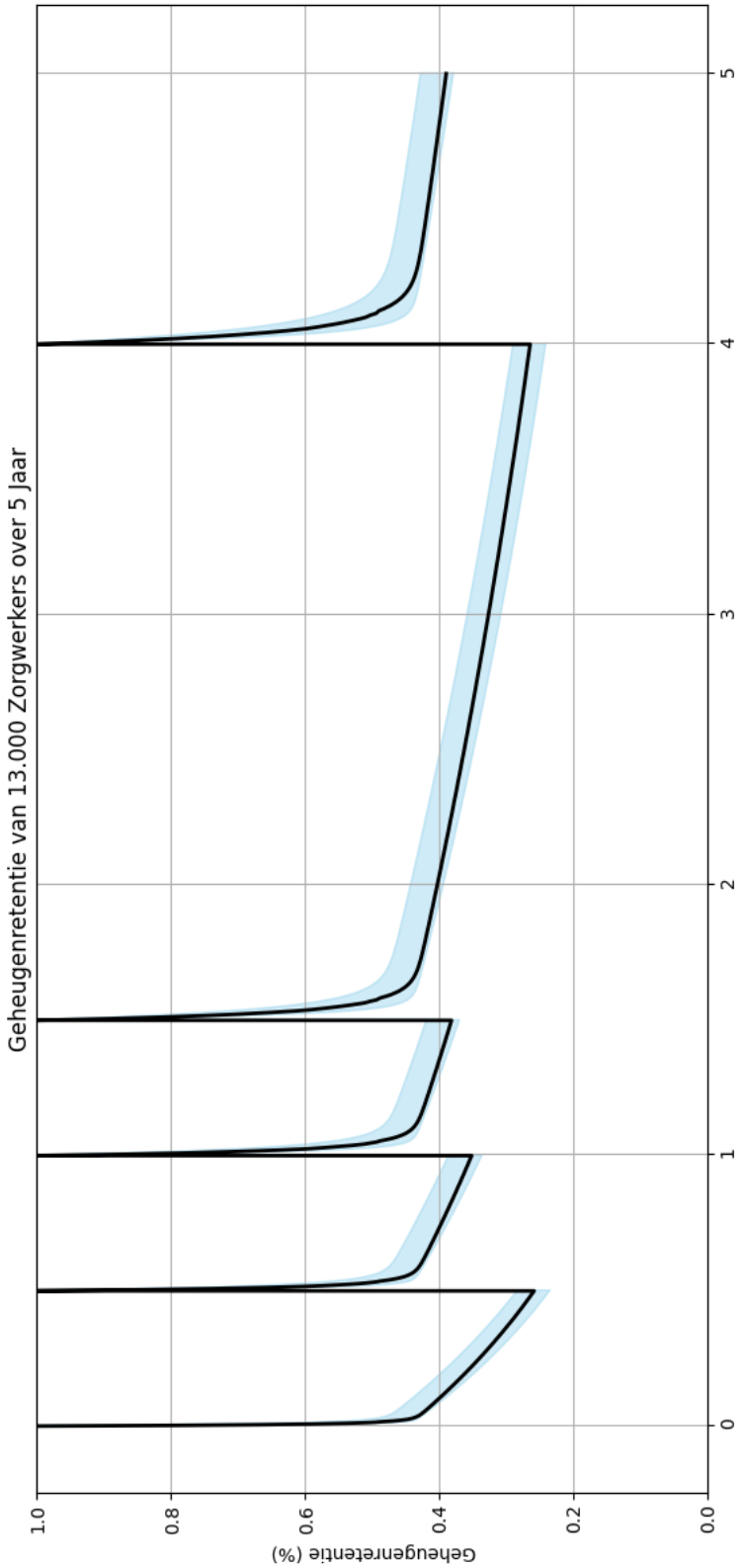
FIGUUR 10: Histogram van experiment 3



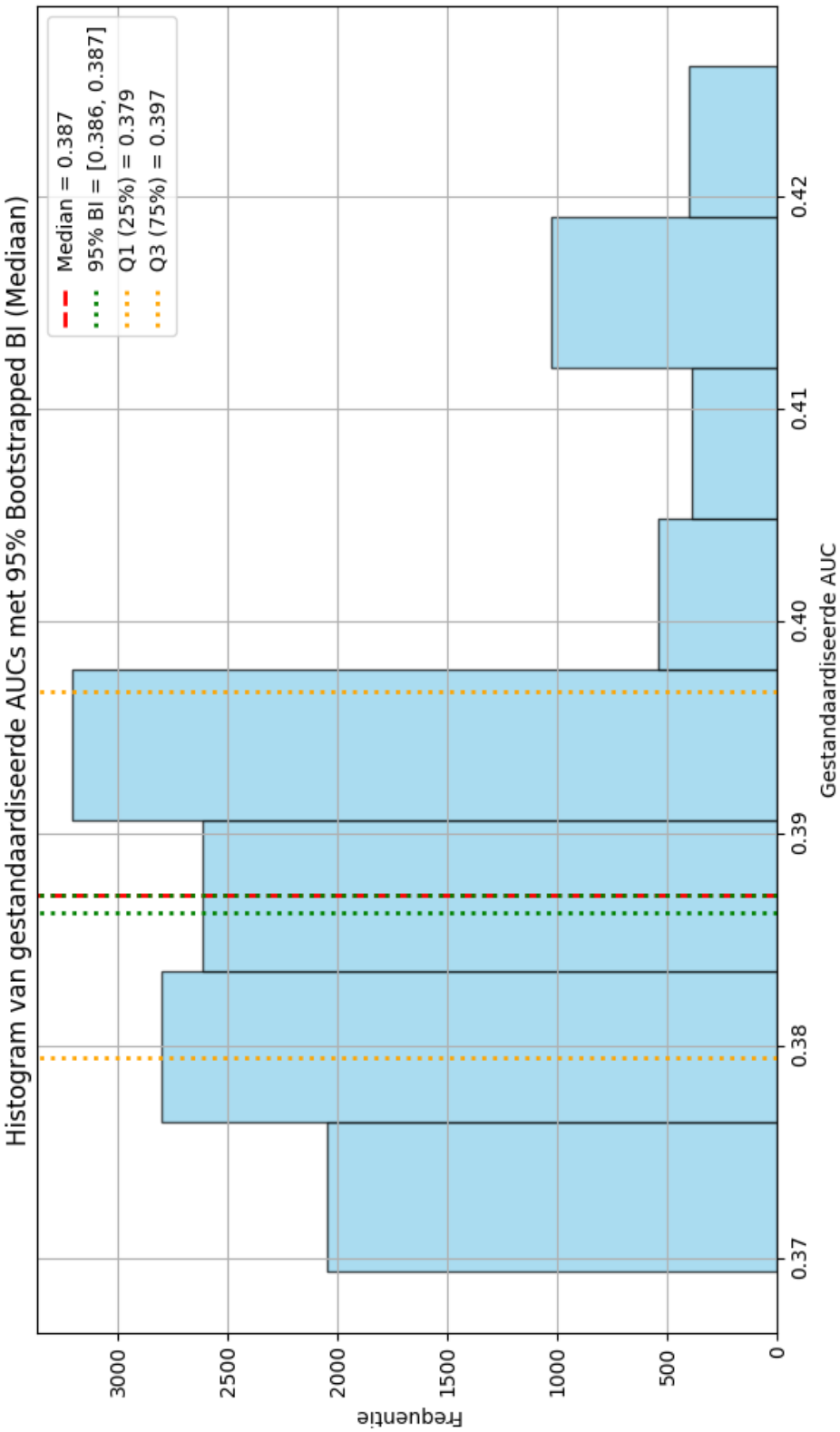
FIGUUR 11: Grafiek van experiment 4



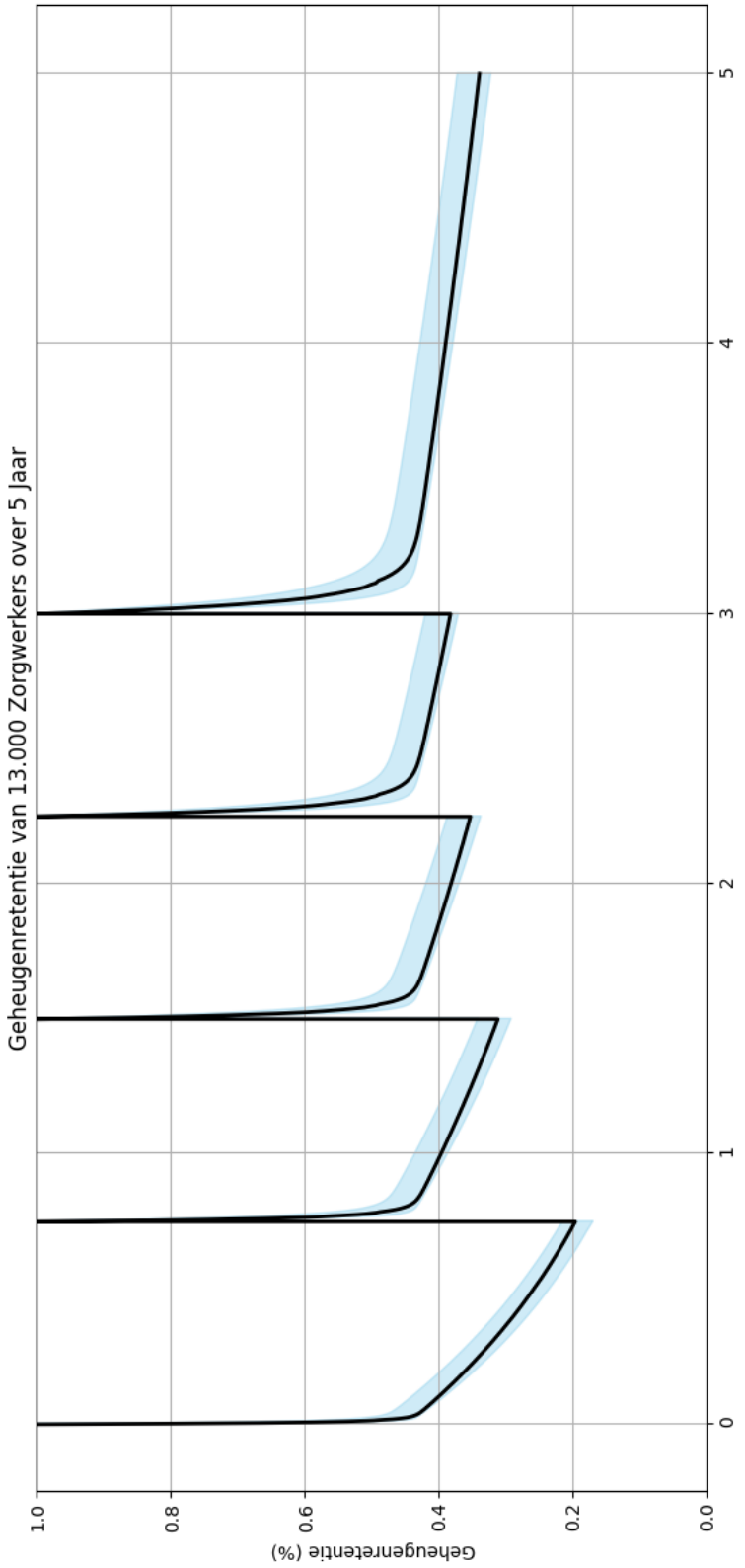
FIGUUR 12: Histogram van experiment 4



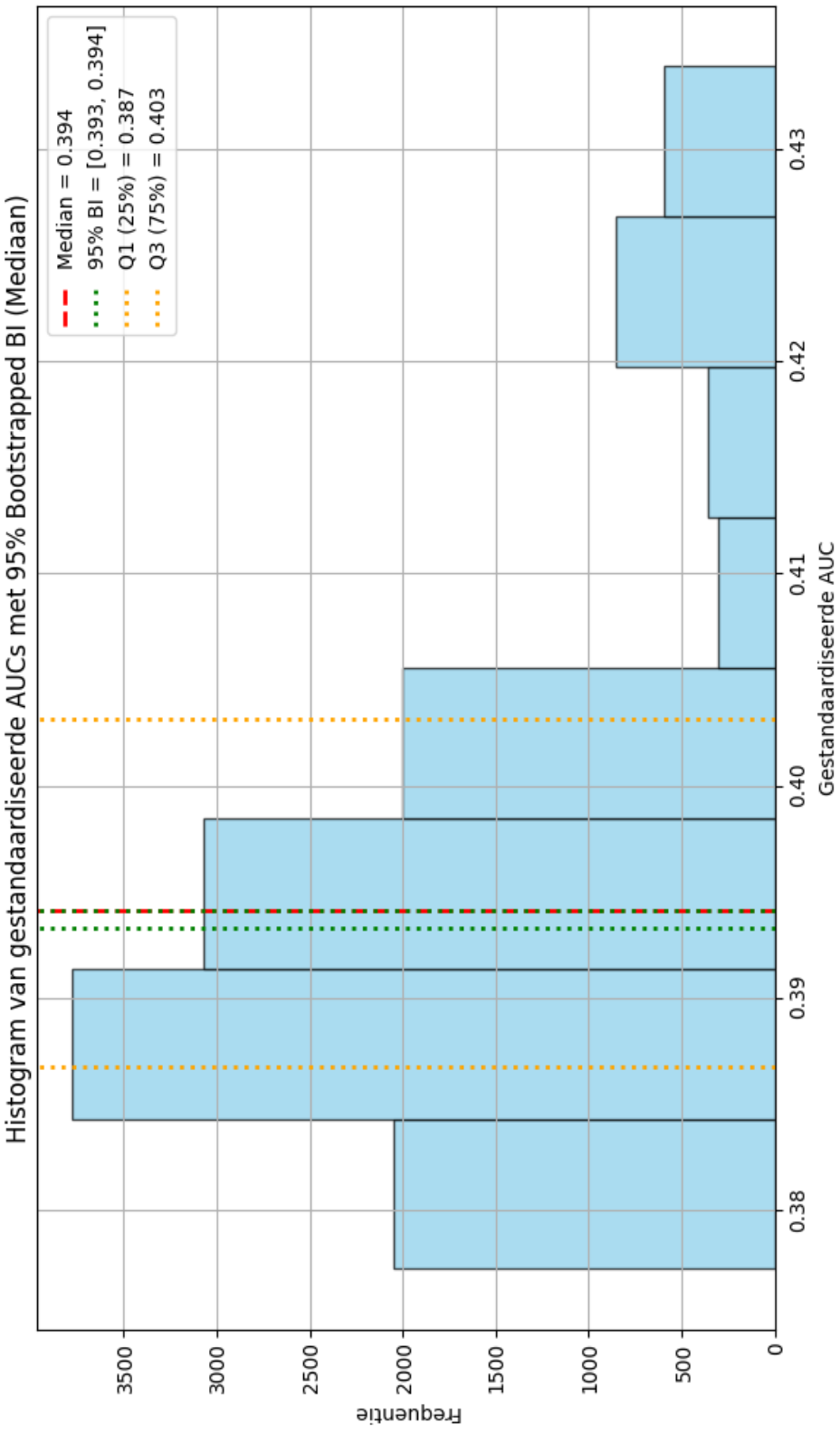
FIGUUR 13: Grafiek van experiment 5



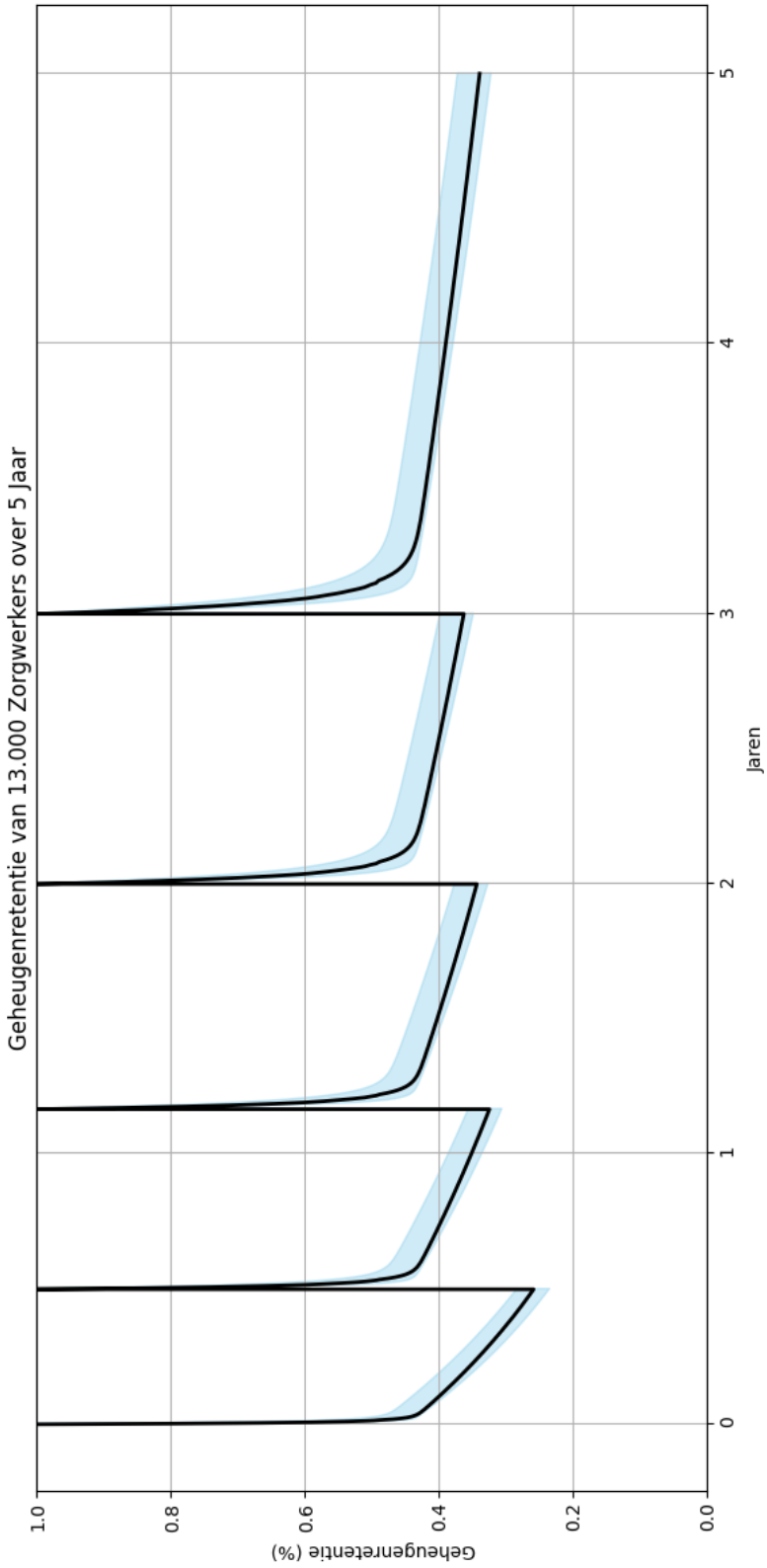
FIGUUR 14: Histogram van experiment 5



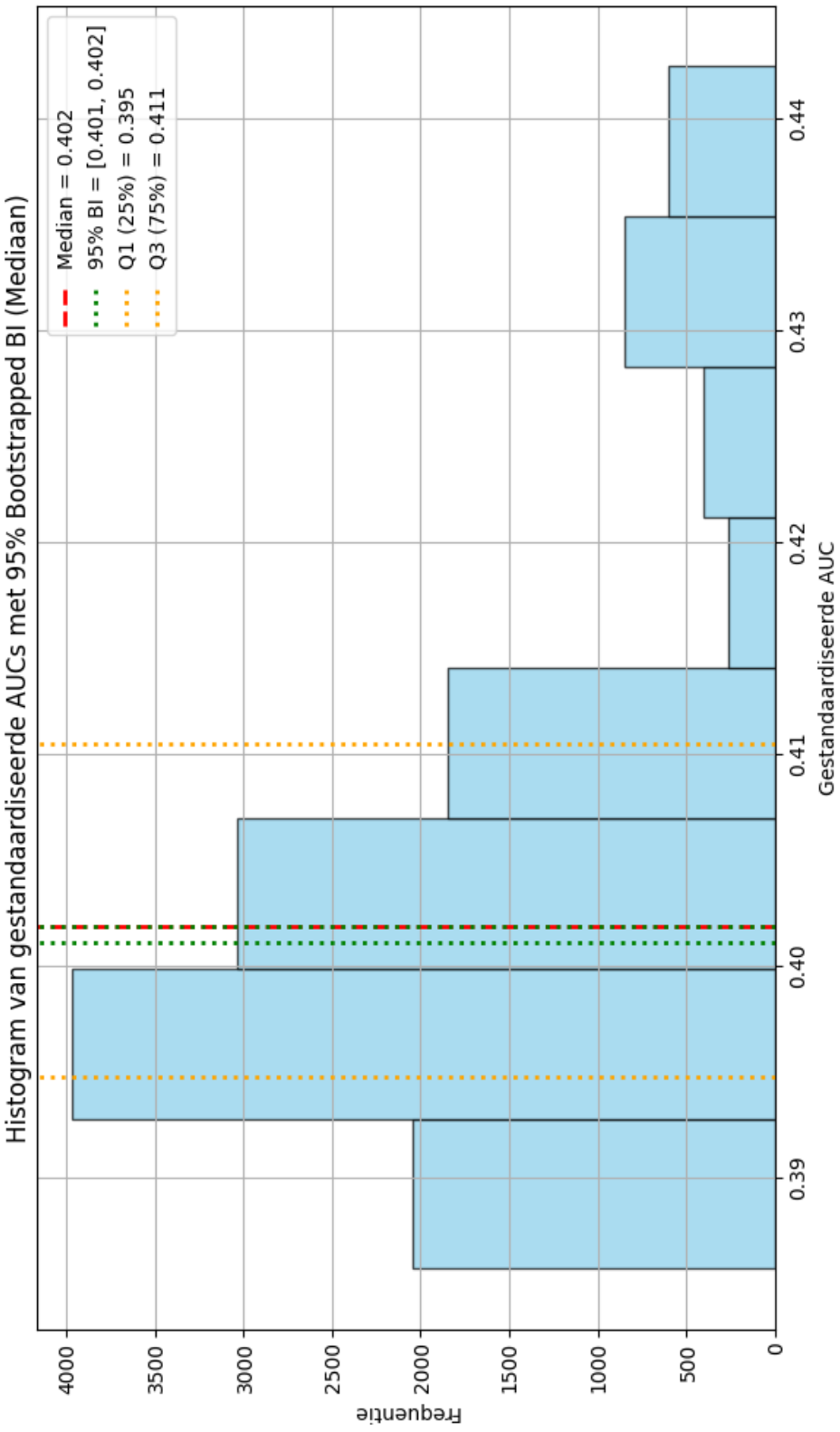
FIGUUR 15: Grafiek van experiment 6



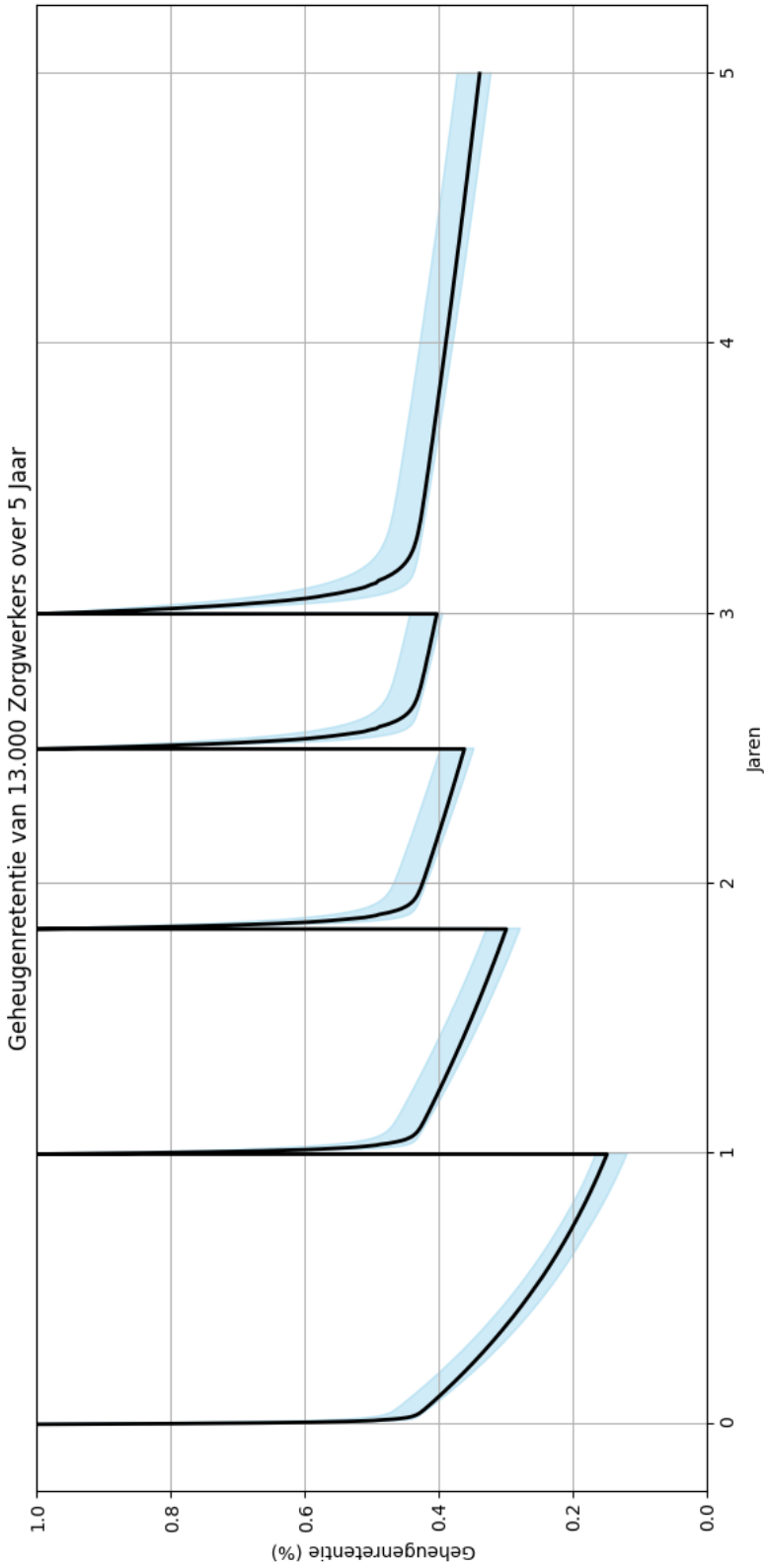
FIGUUR 16: Histogram van experiment 6



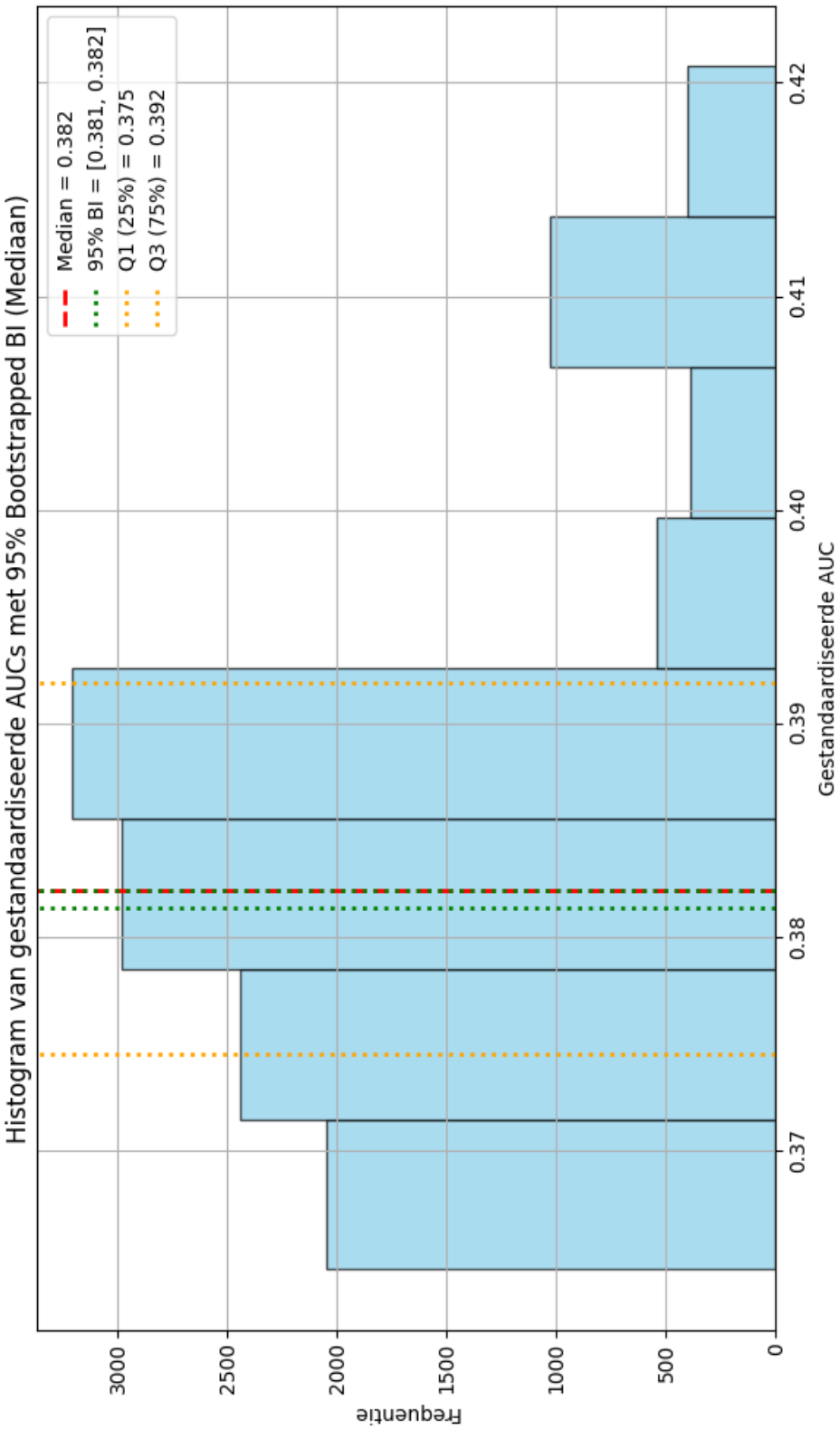
FIGUUR 17: Grafiek van experiment 7



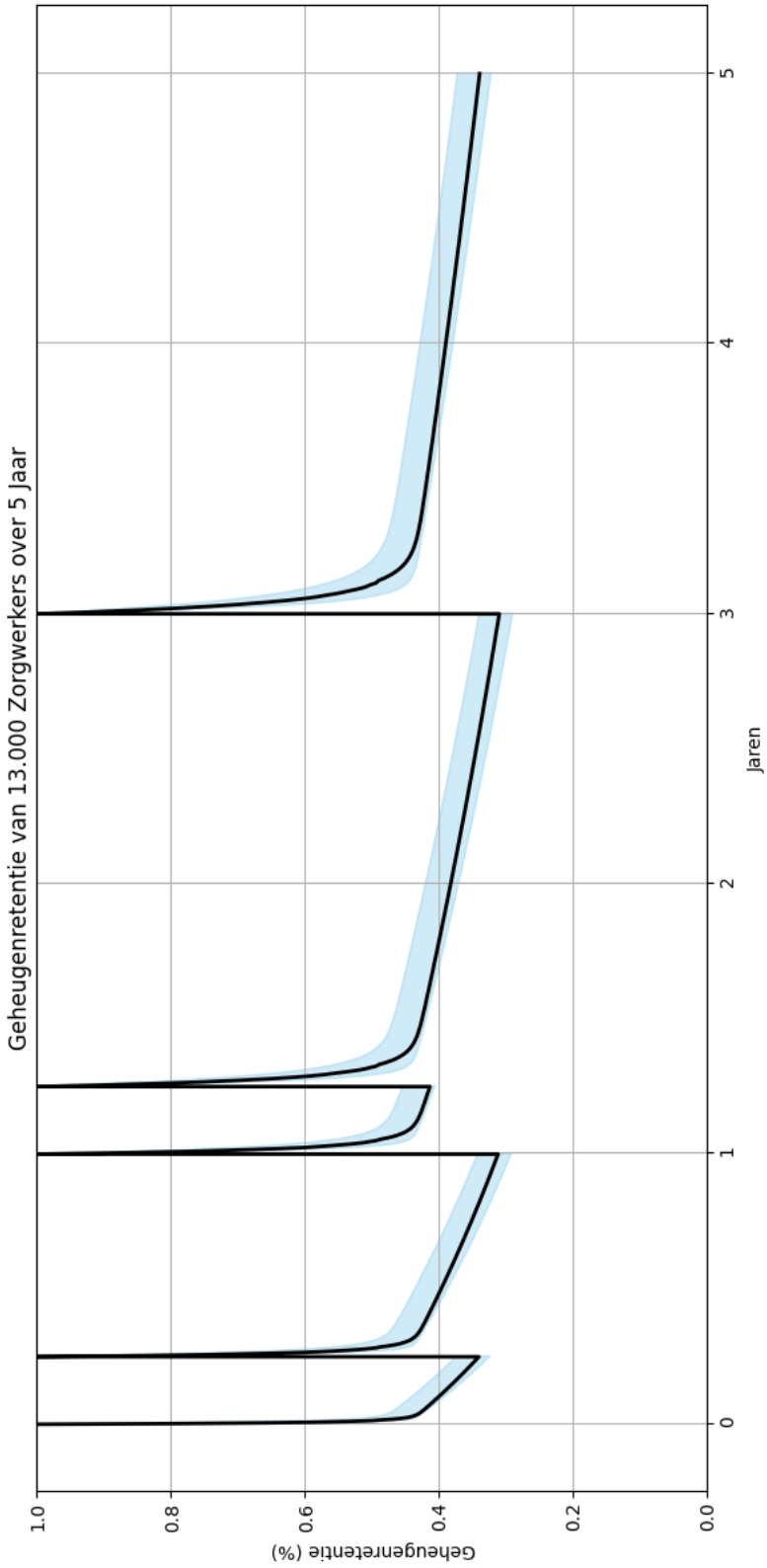
FIGUUR 18: Histogram van experiment 7



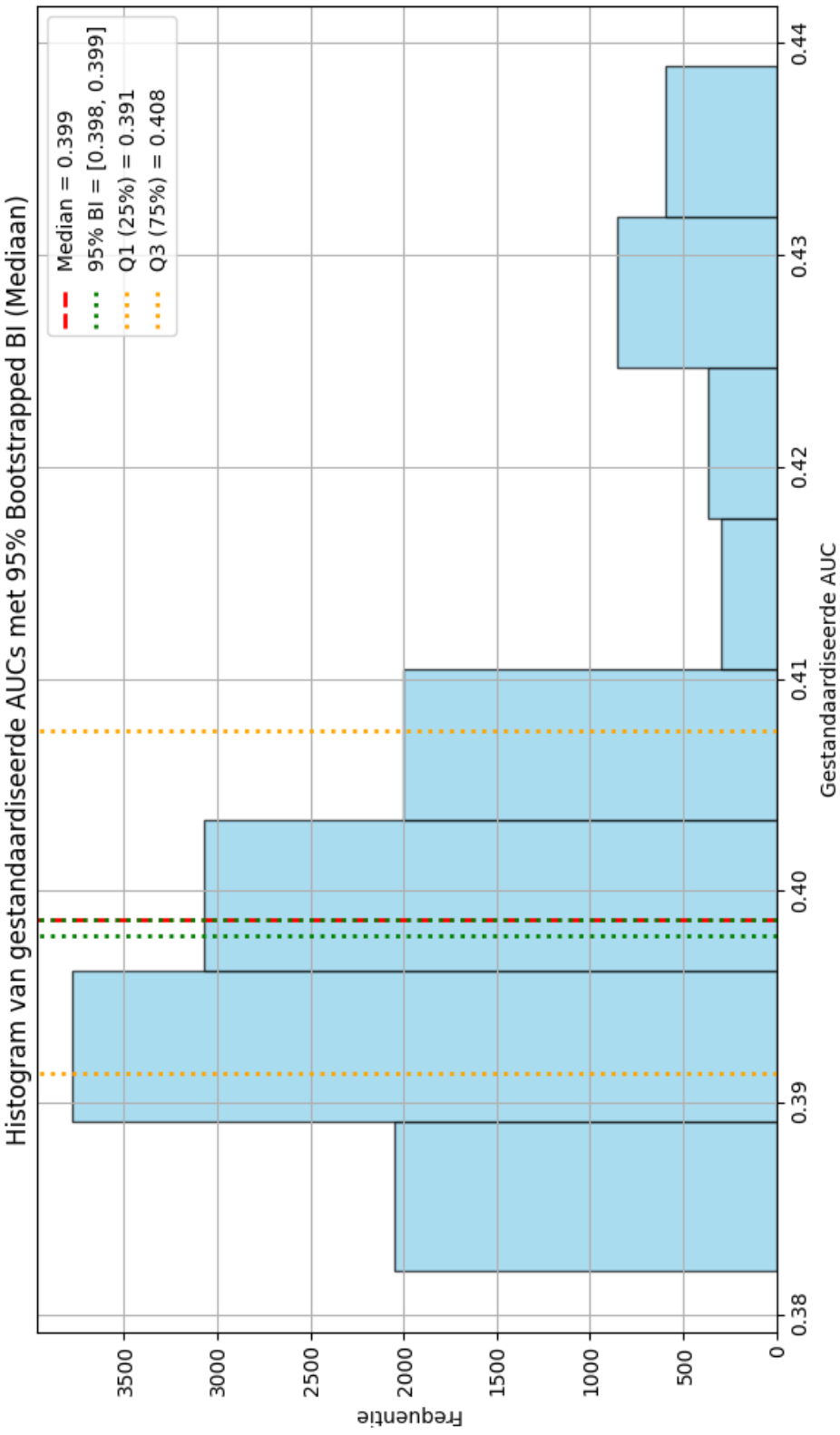
FIGUUR 19: Grafiek van experiment 8



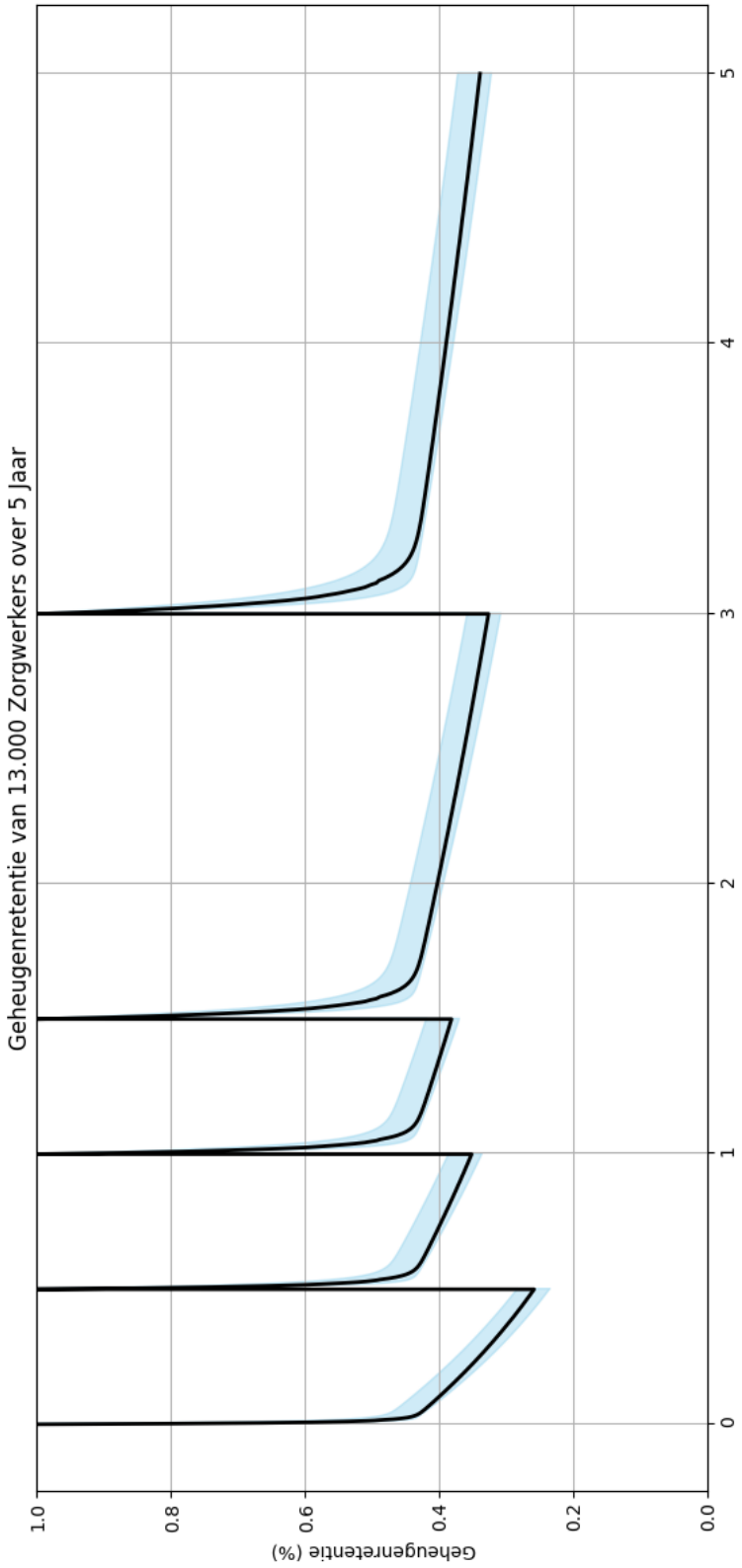
FIGUUR 20: Histogram van experiment 8



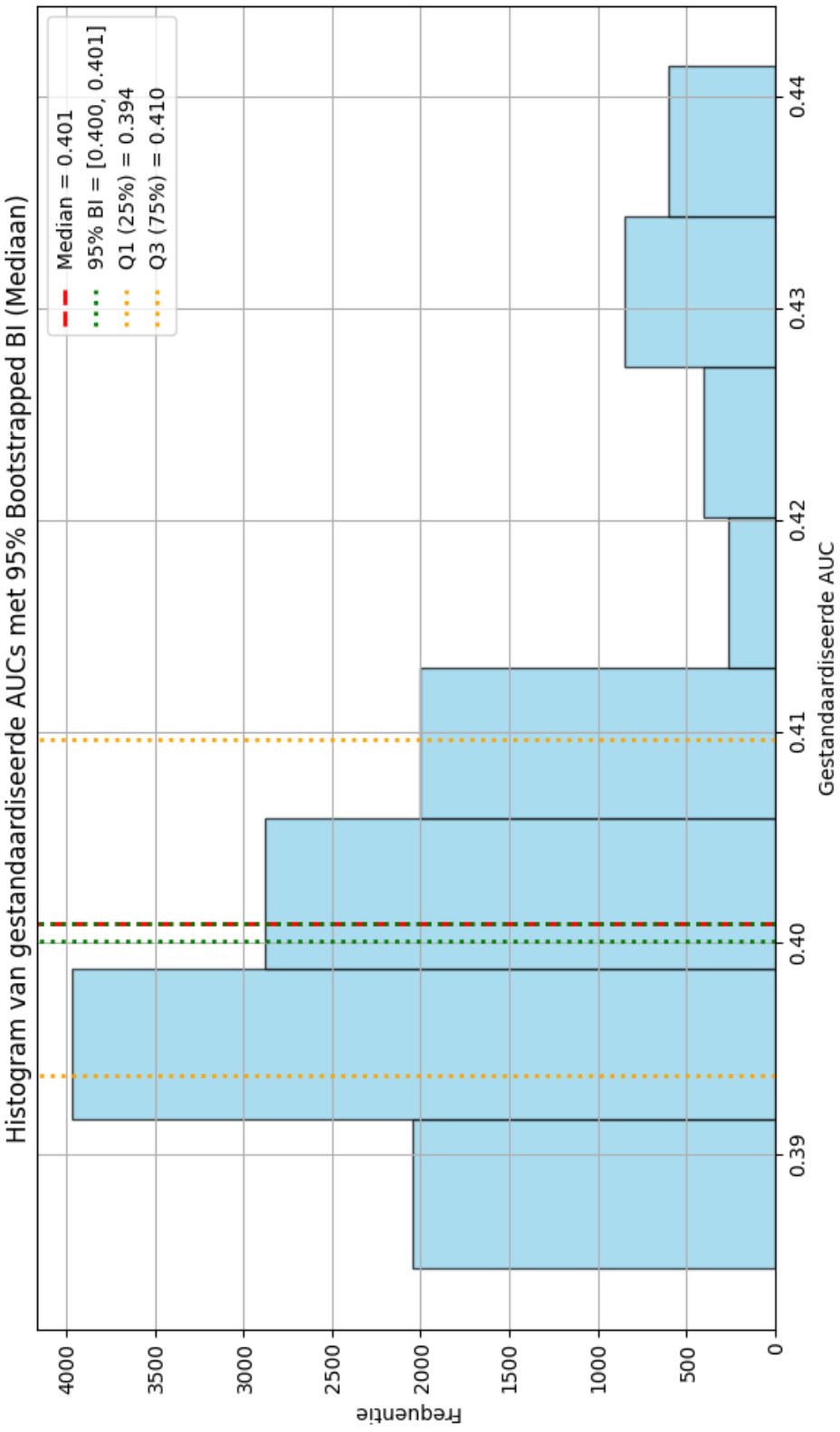
FIGUUR 21: Grafiek van experiment 9



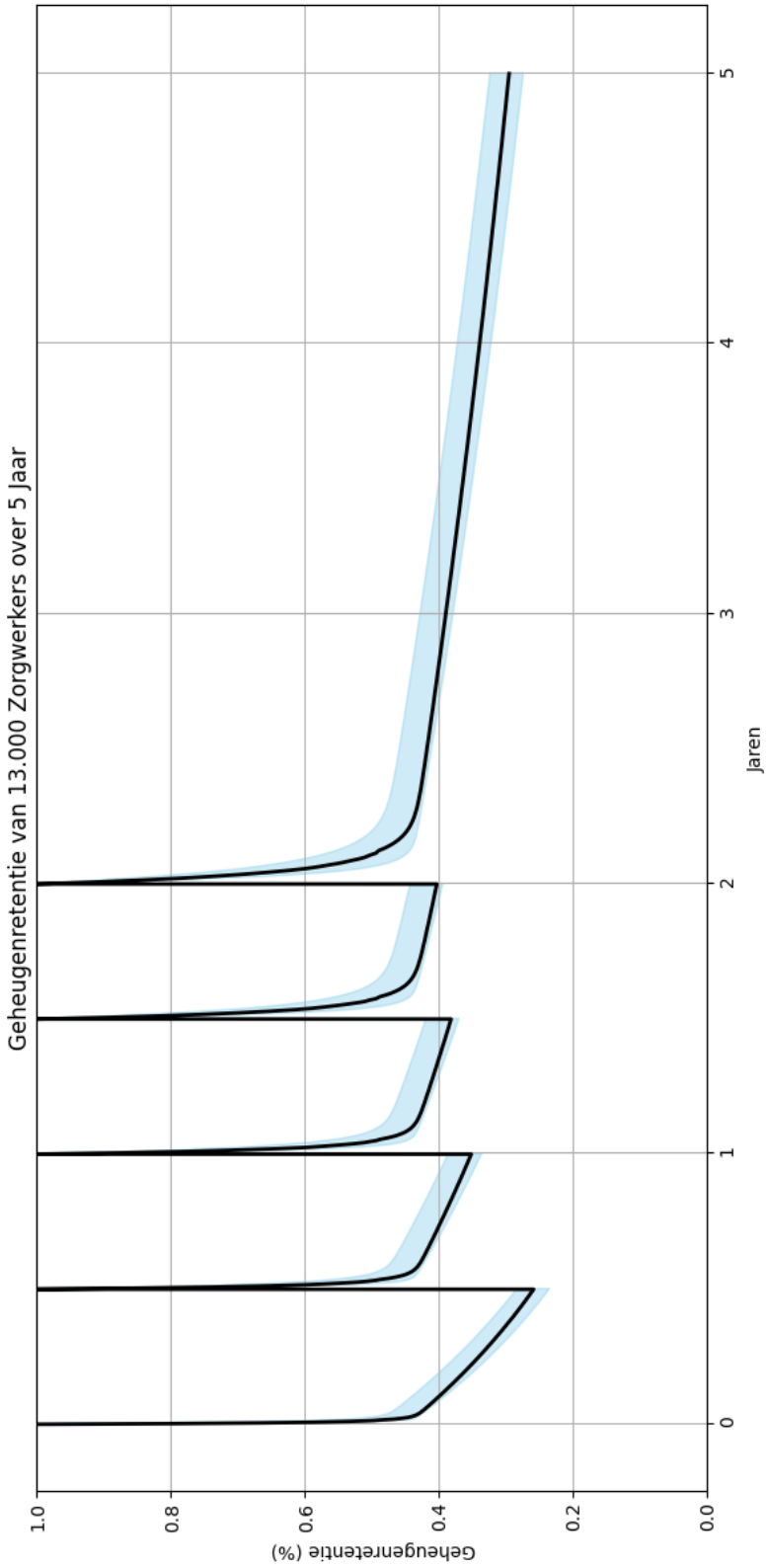
FIGUUR 22: Histogram van experiment 9



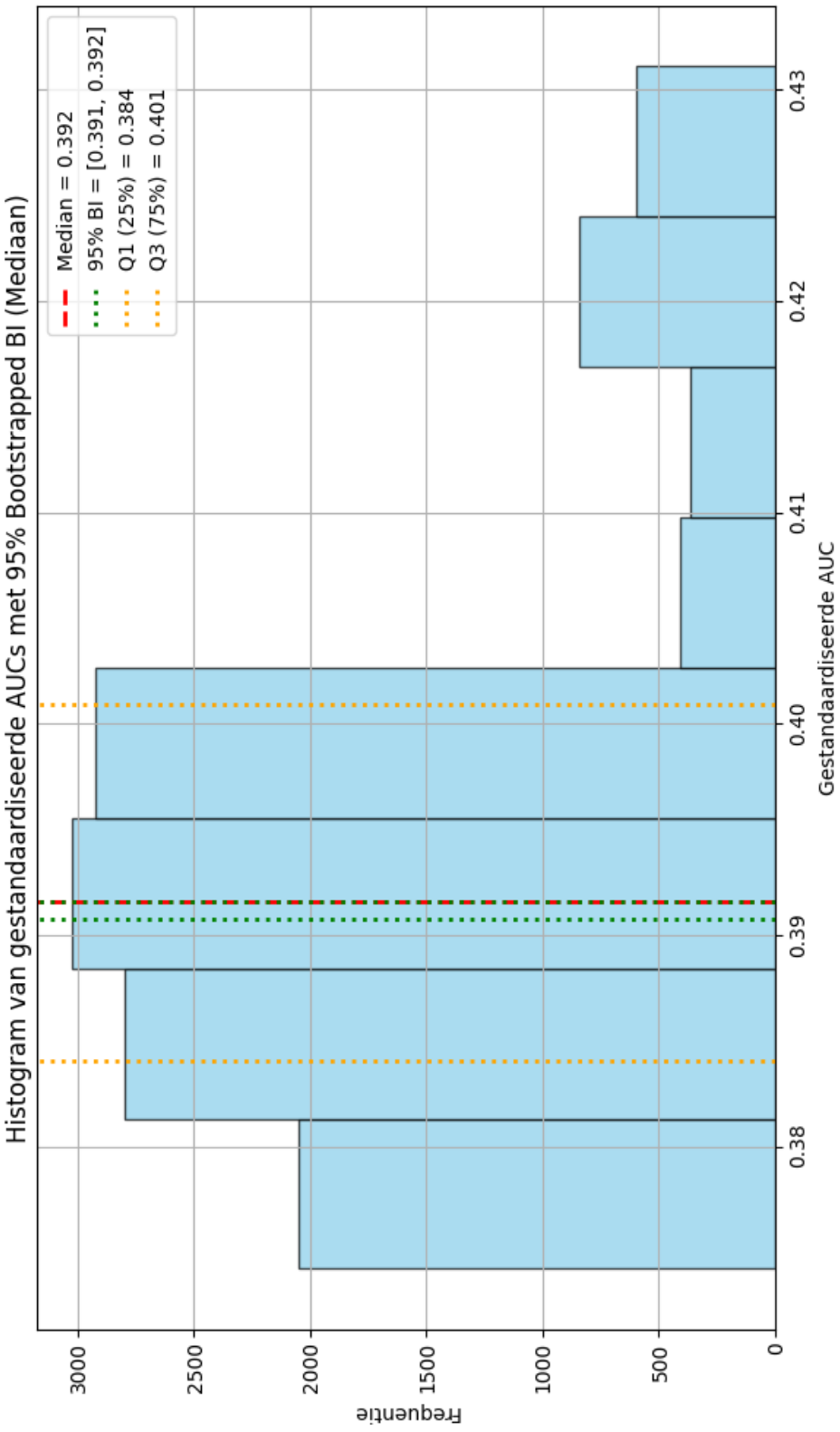
FIGUUR 23: Grafiek van experiment 10



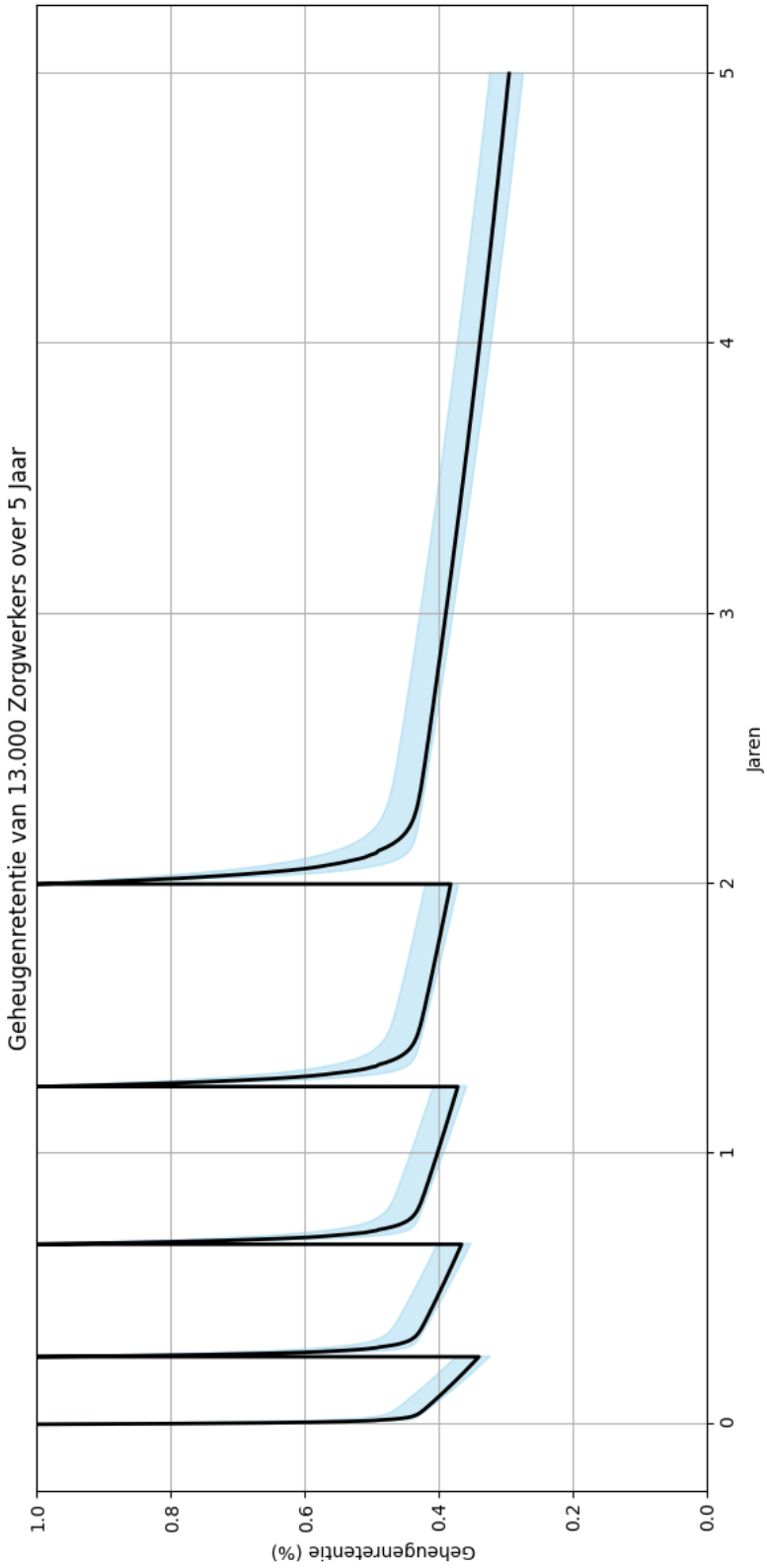
FIGUUR 24: Histogram van experiment 10



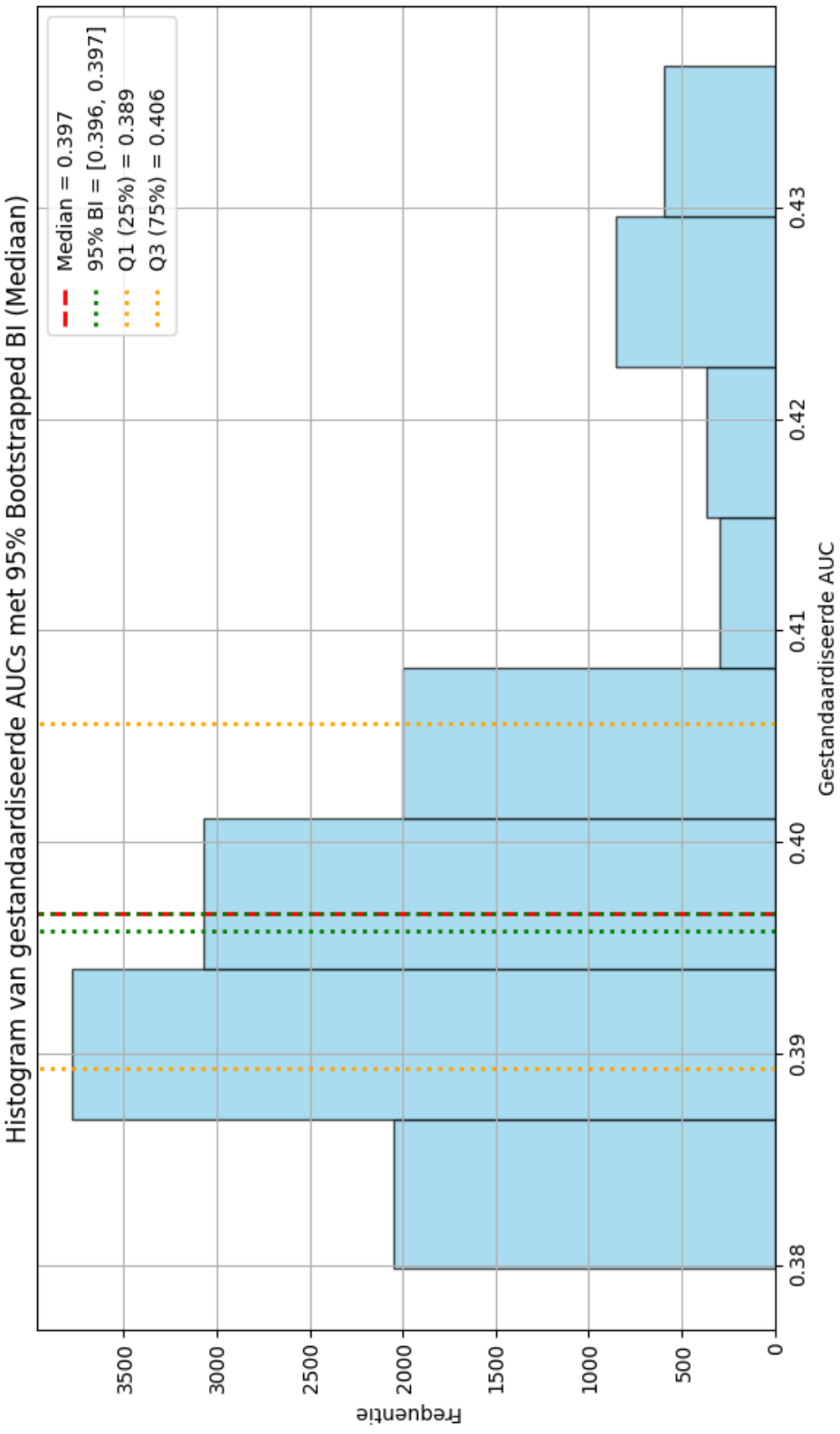
FIGUUR 25: Grafiek van experiment 11



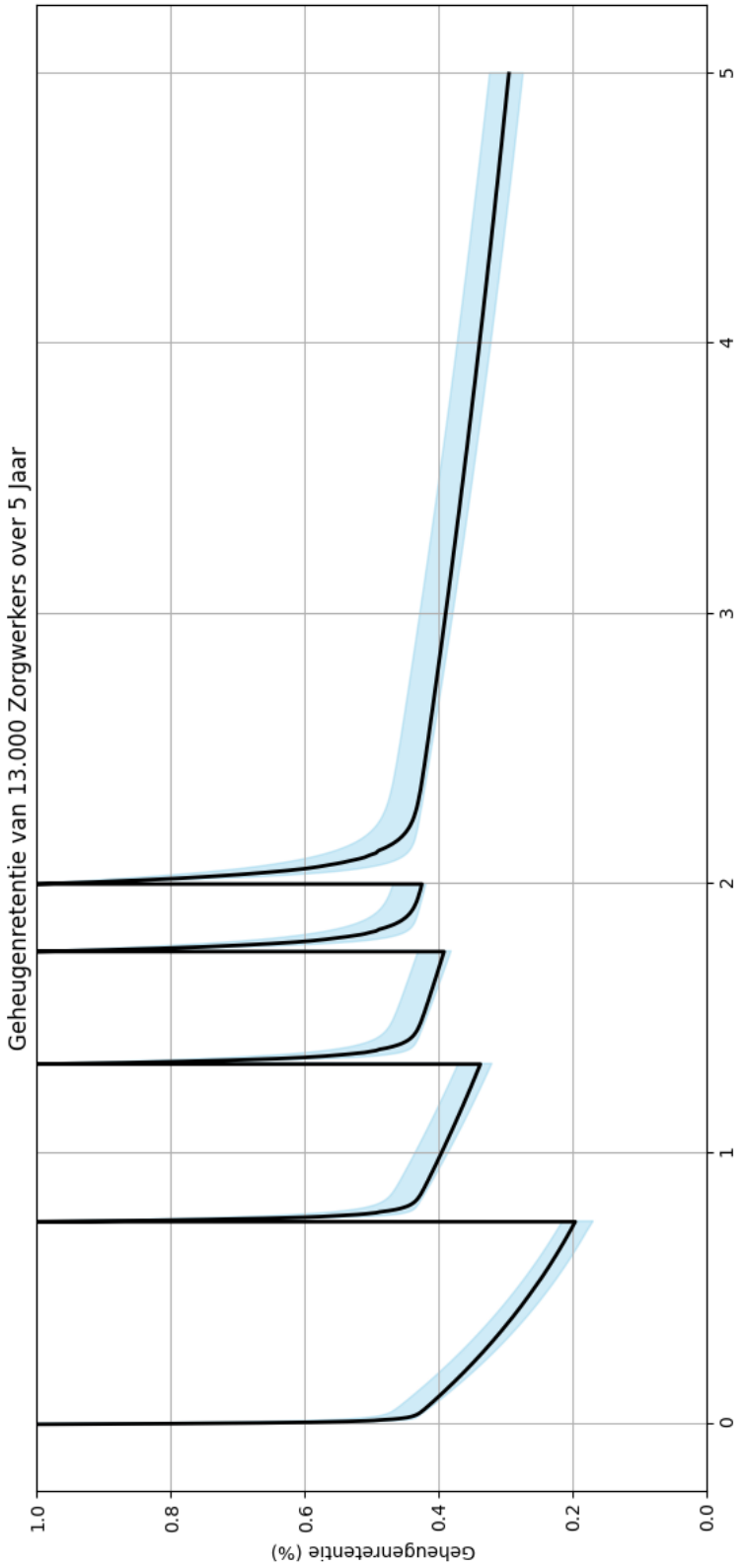
FIGUUR 26: Histogram van experiment 11



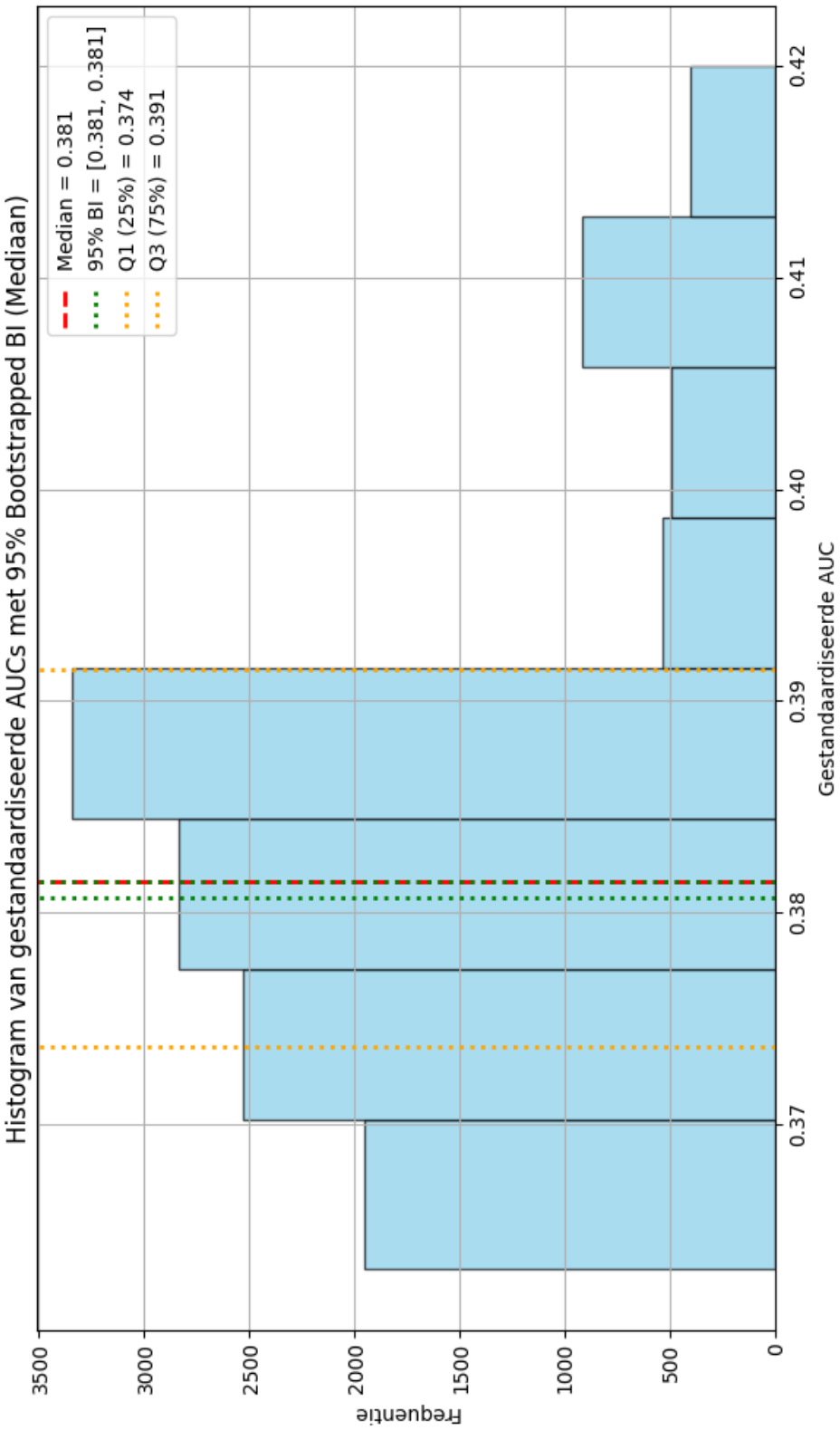
FIGUUR 27: Grafiek van experiment 12



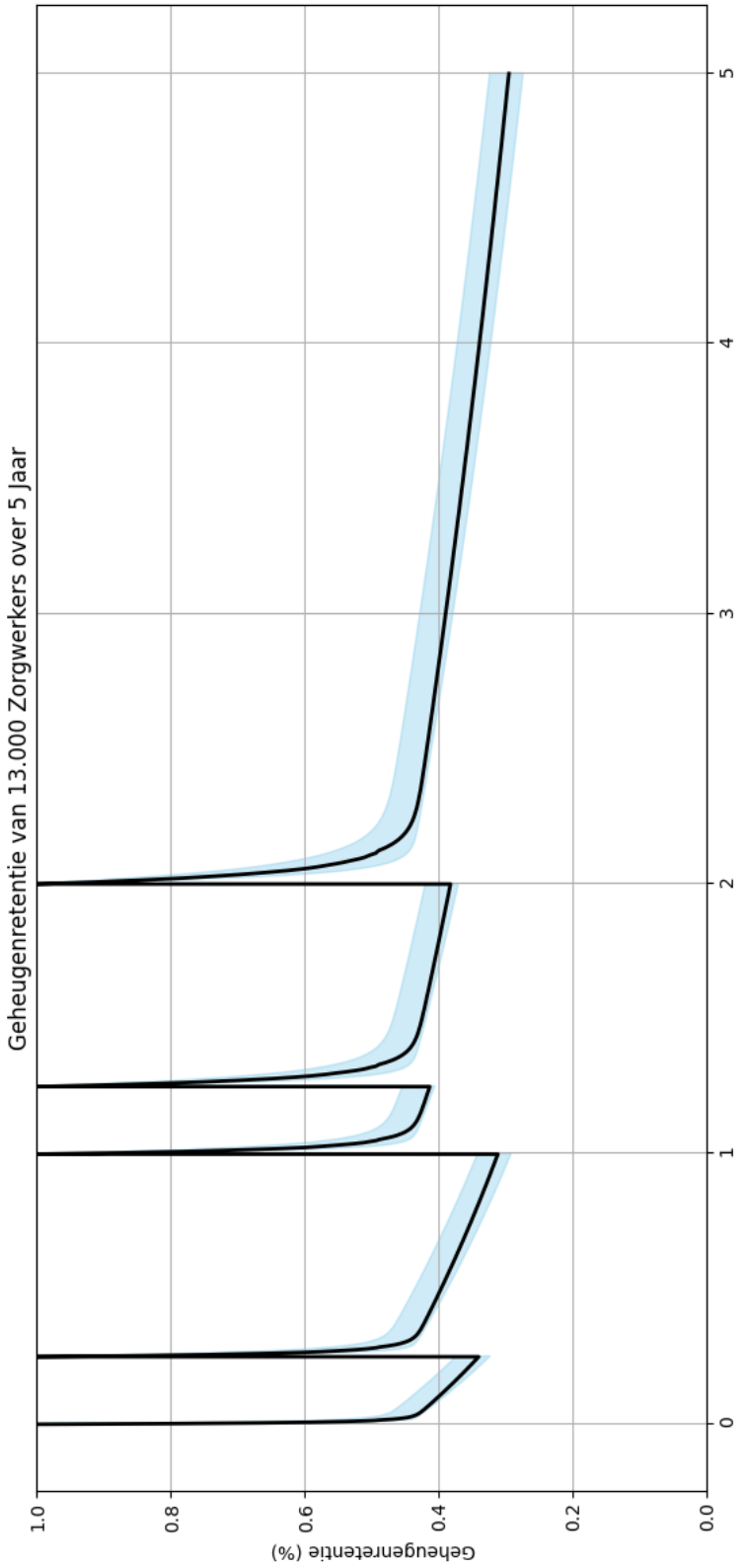
FIGUUR 28: Histogram van experiment 12



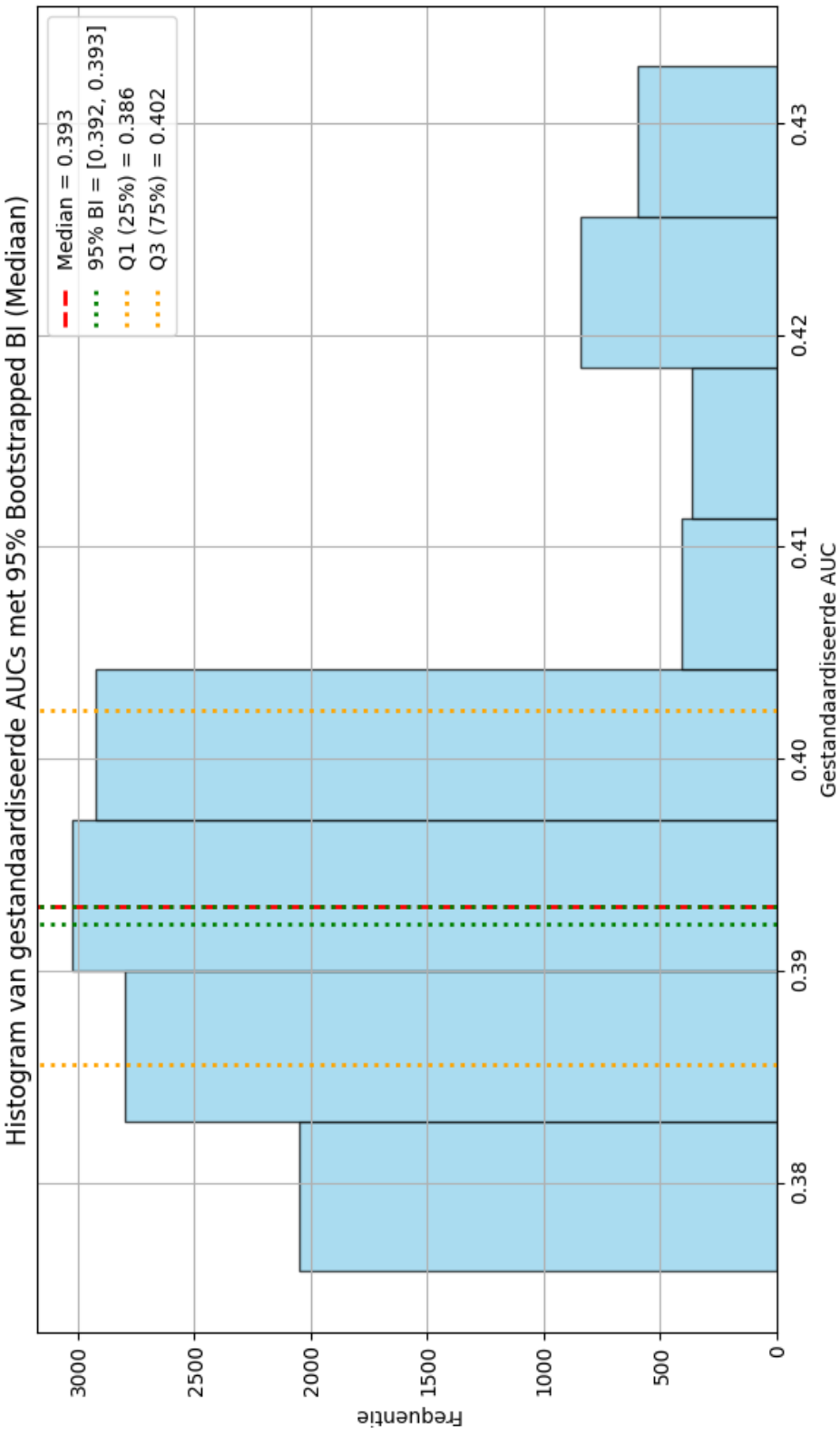
FIGUUR 29: Grafiek van experiment 13



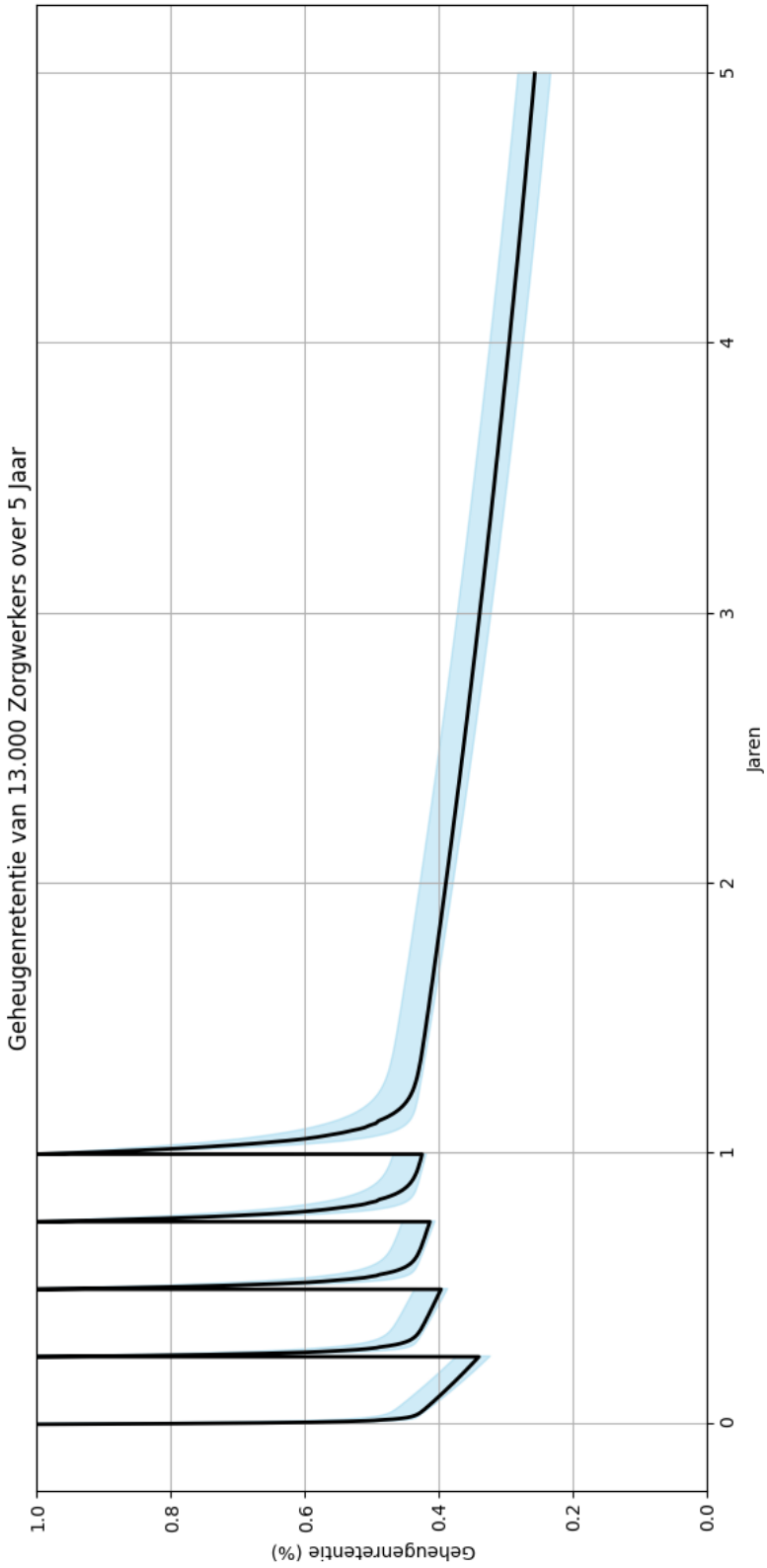
FIGUUR 30: Histogram van experiment 13



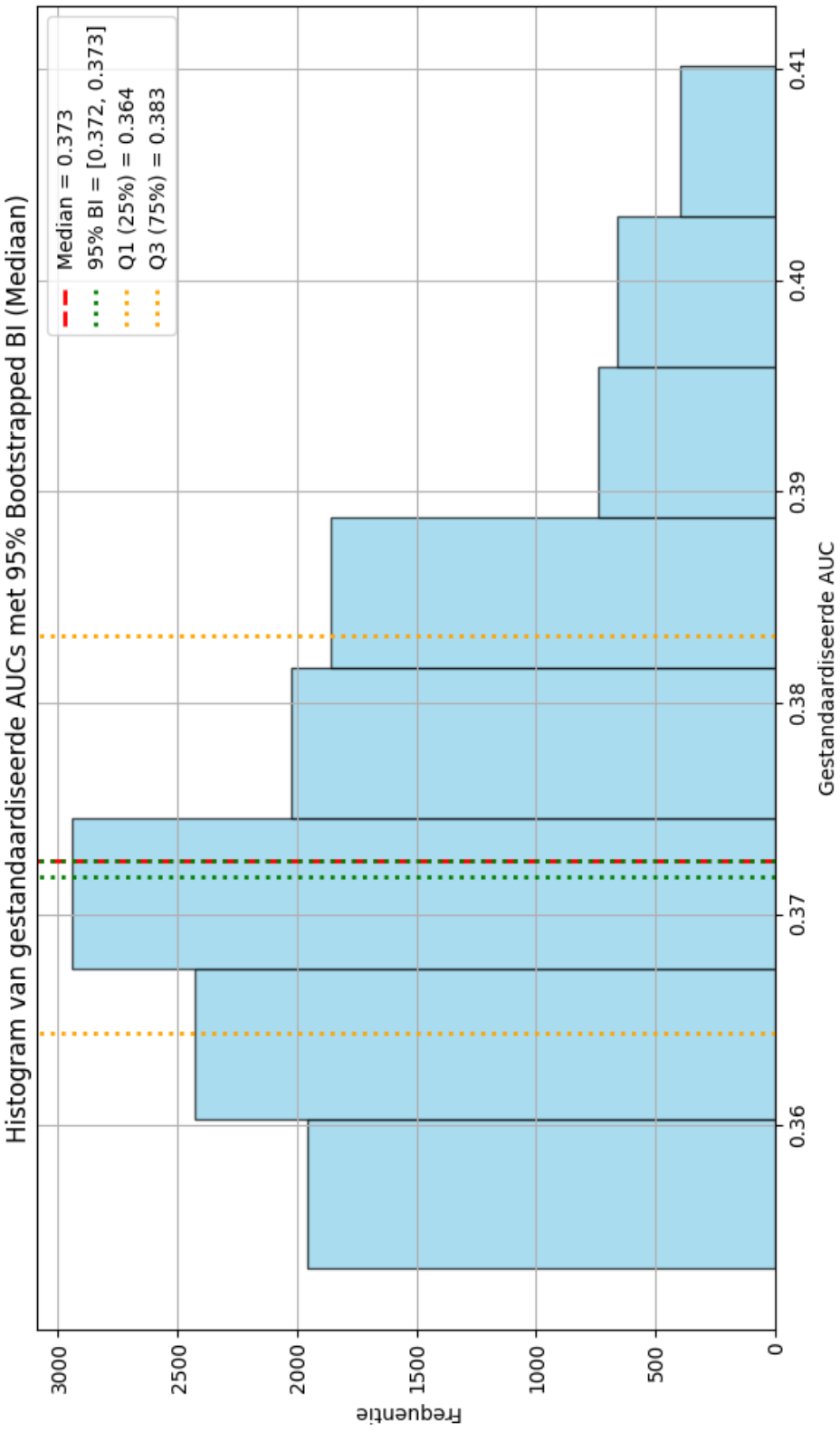
FIGUUR 31: Grafiek van experiment 14



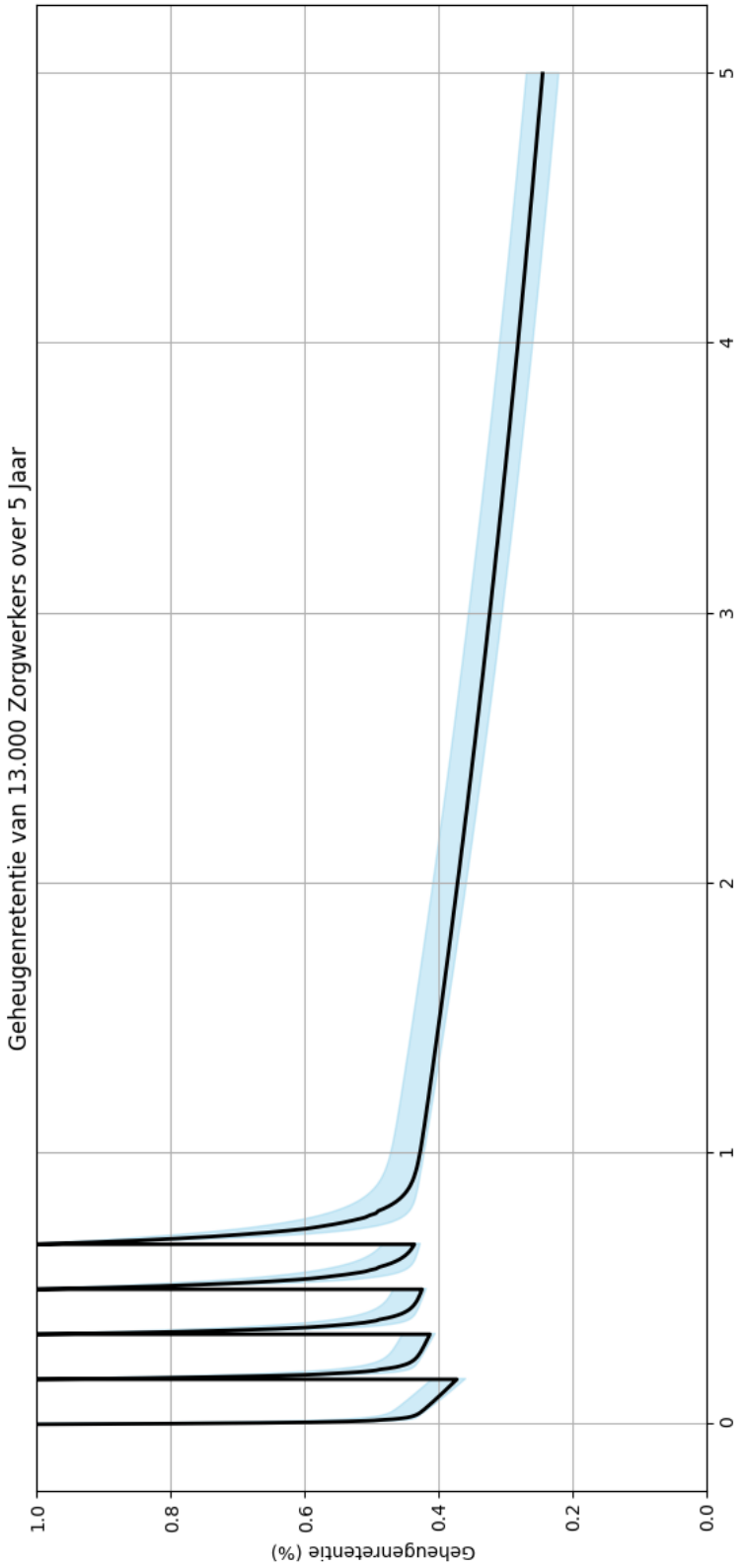
FIGUUR 32: Histogram van experiment 14



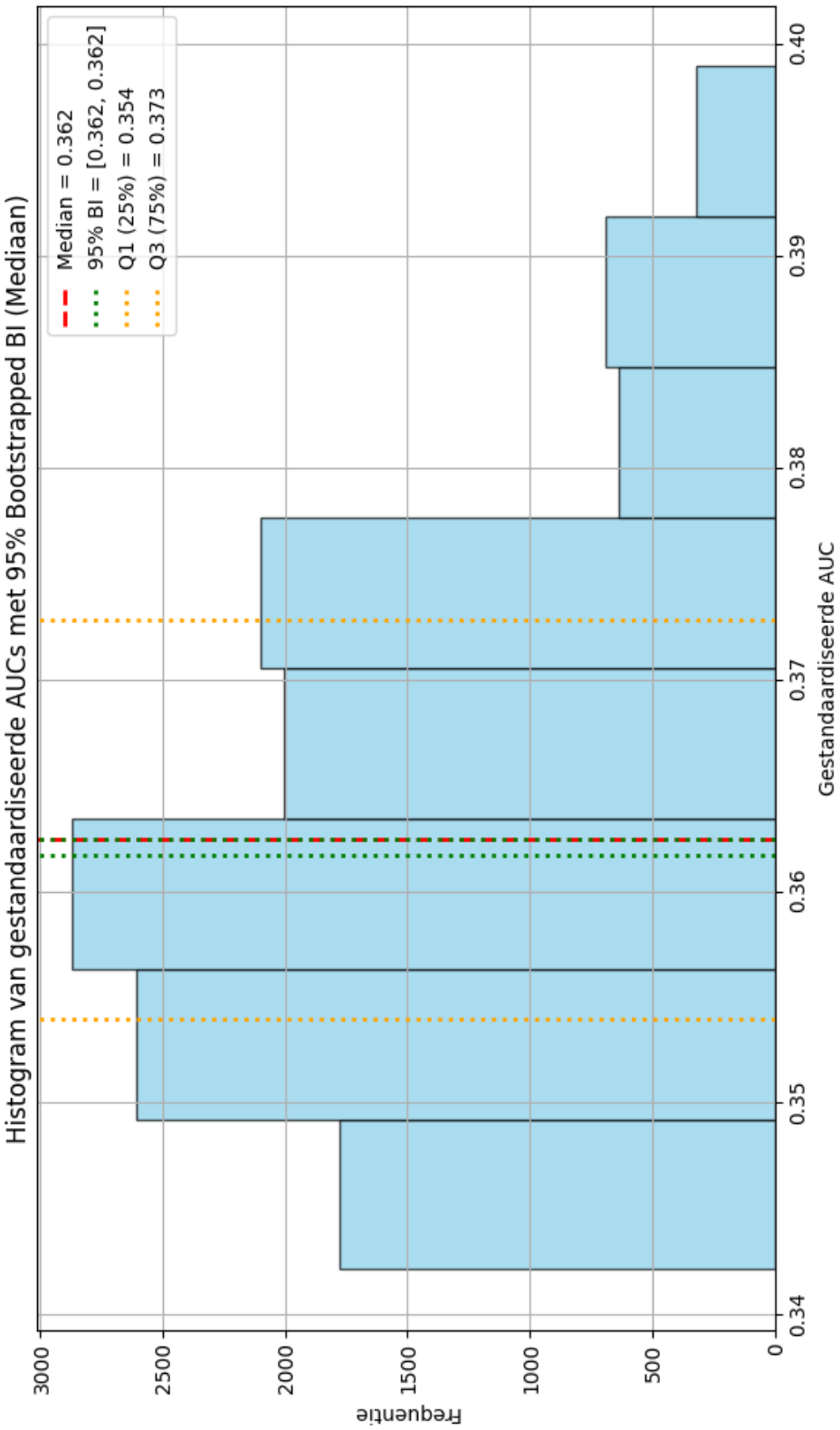
FIGUUR 33: Grafiek van experiment 15



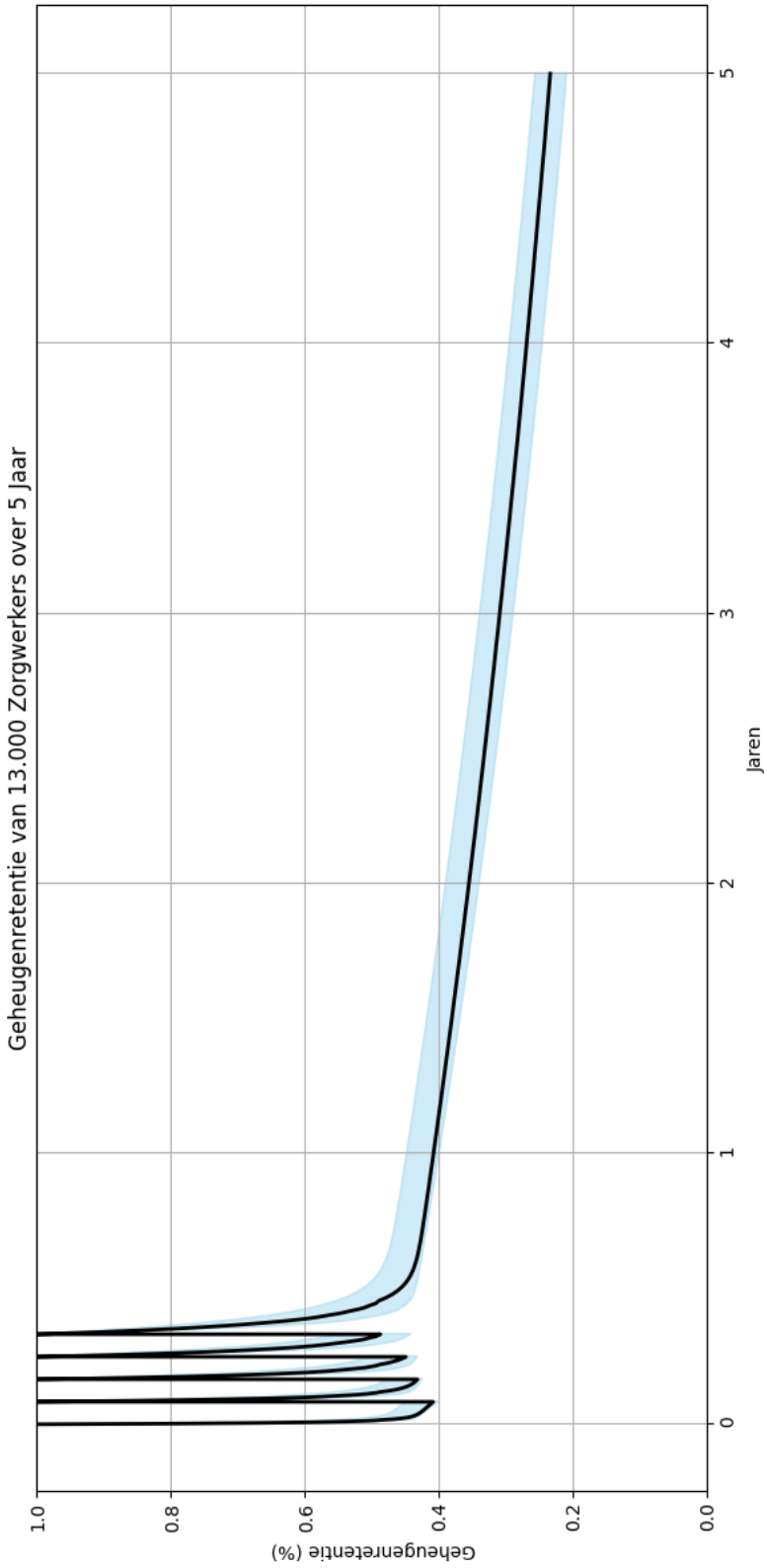
FIGUUR 34: Histogram van experiment 15



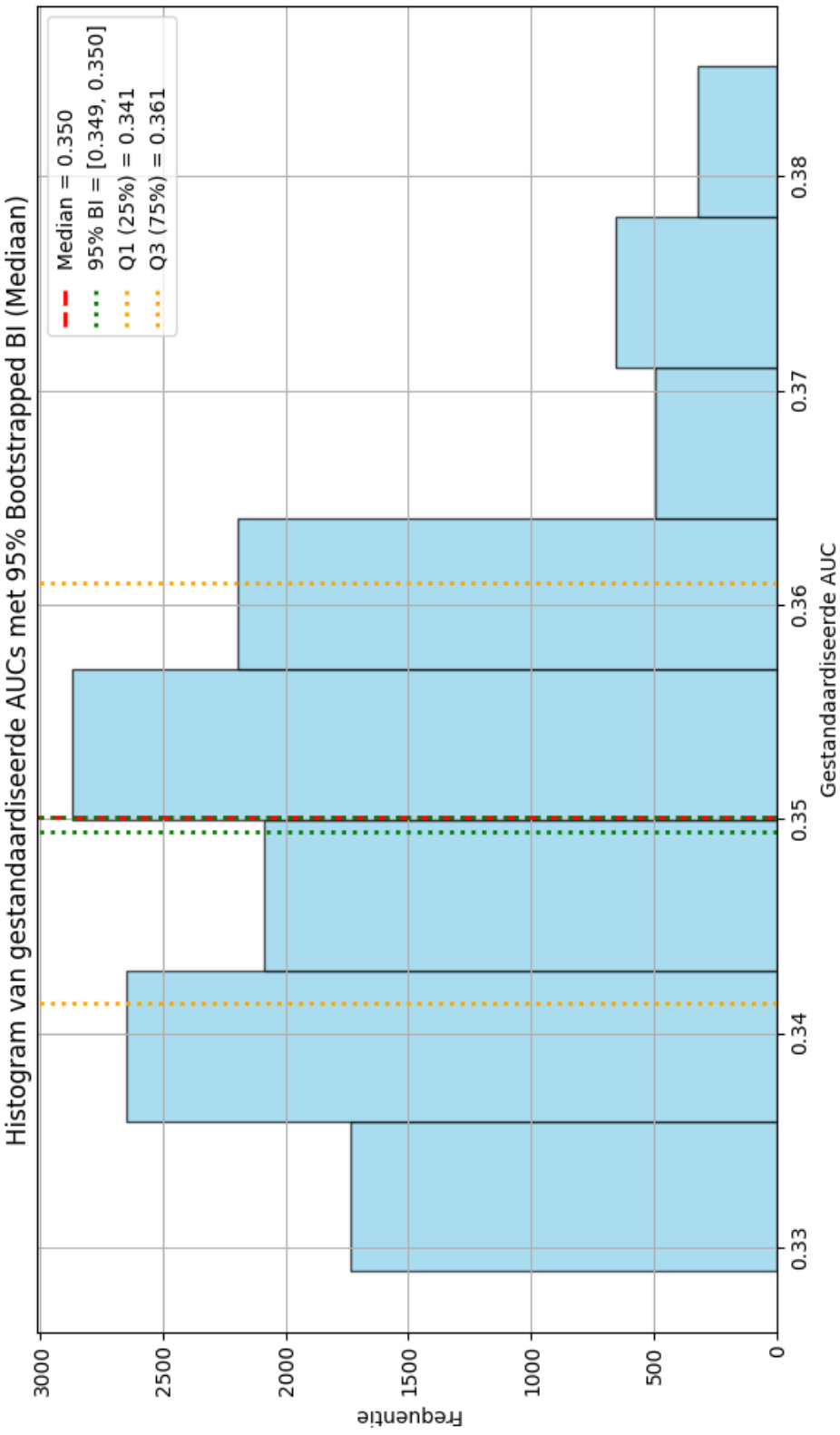
FIGUUR 35: Grafiek van experiment 16



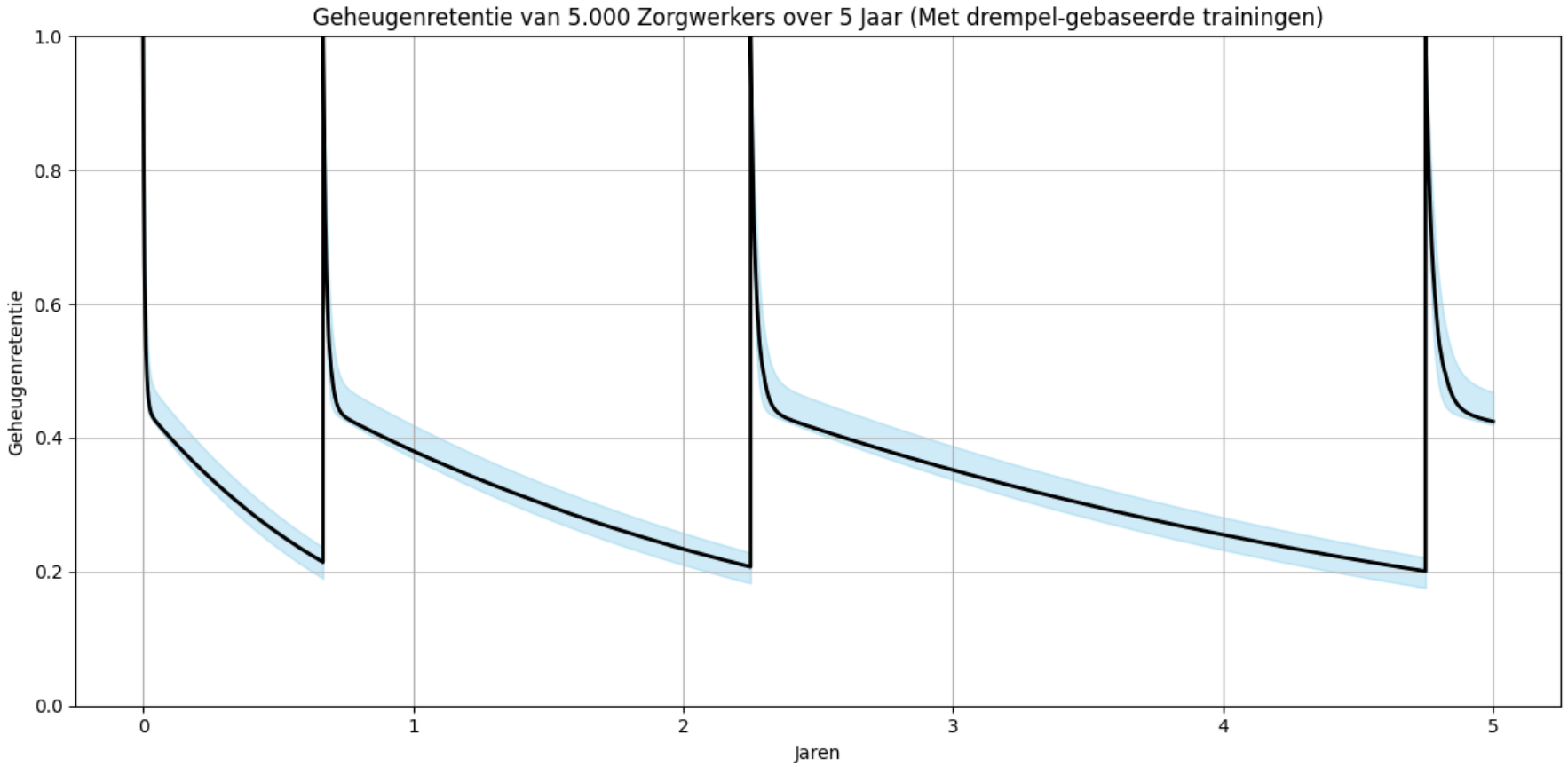
FIGUUR 36: Histogram van experiment 16



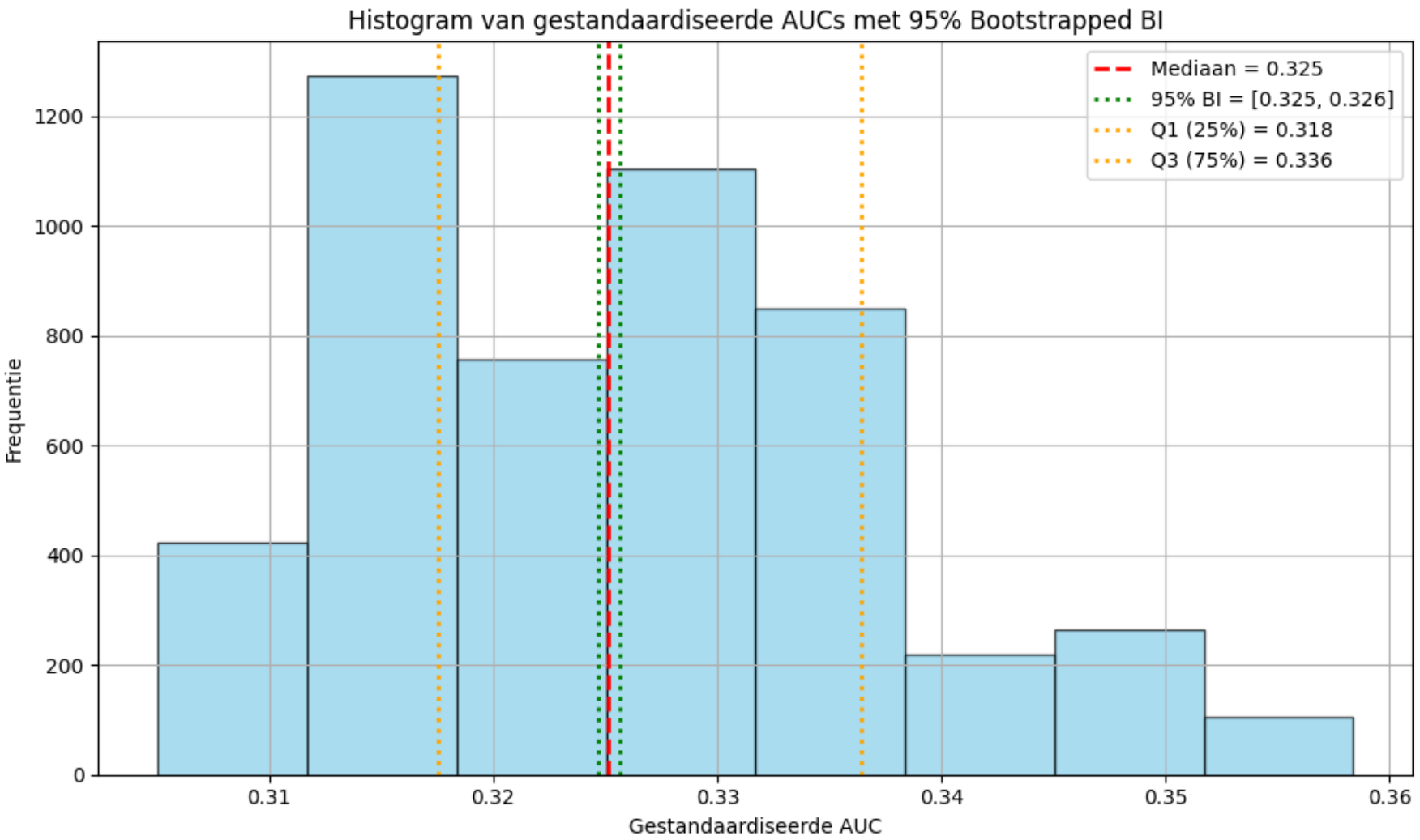
FIGUUR 37: Grafiek van experiment 17



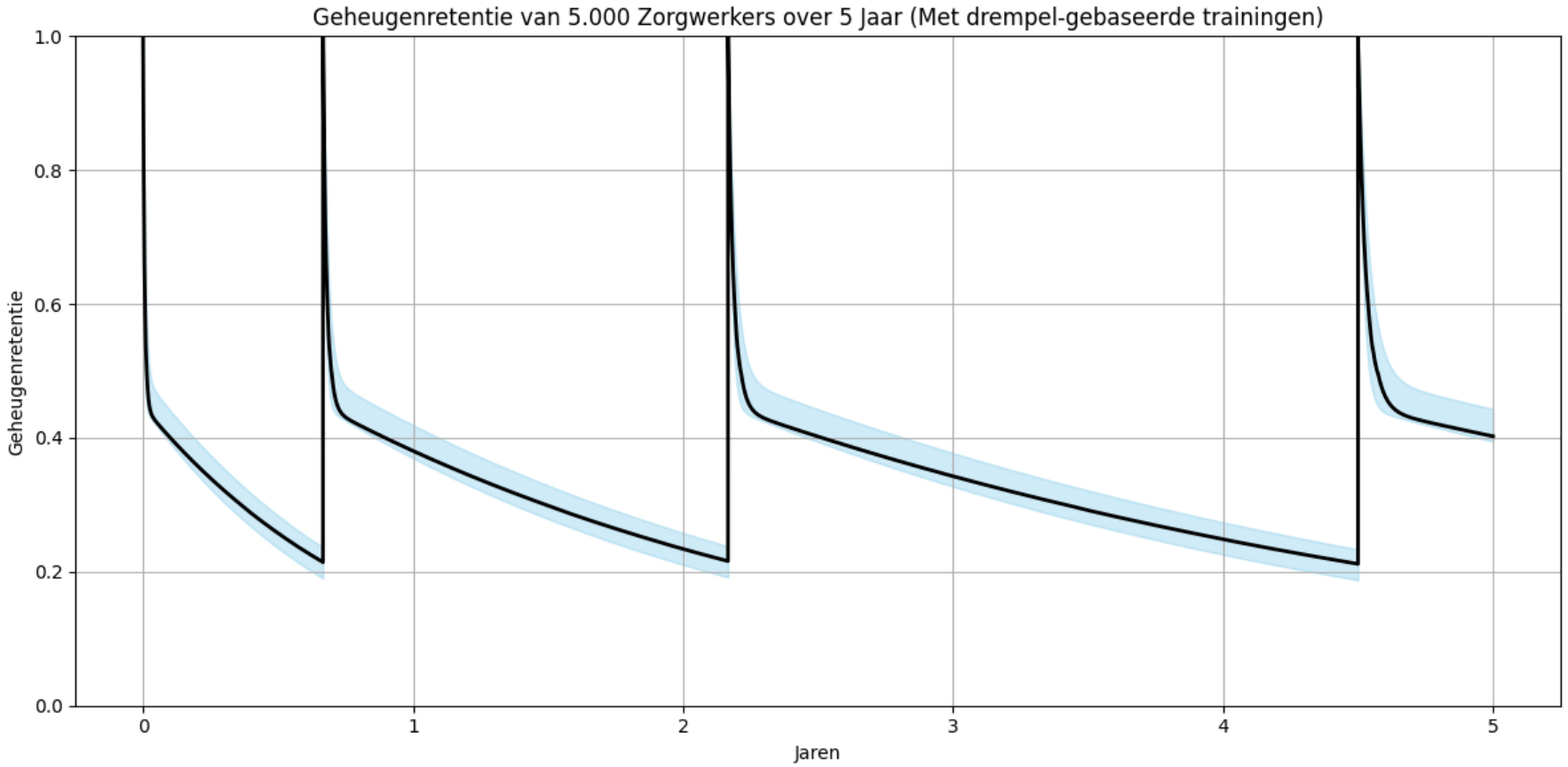
FIGUUR 38: Histogram van experiment 17



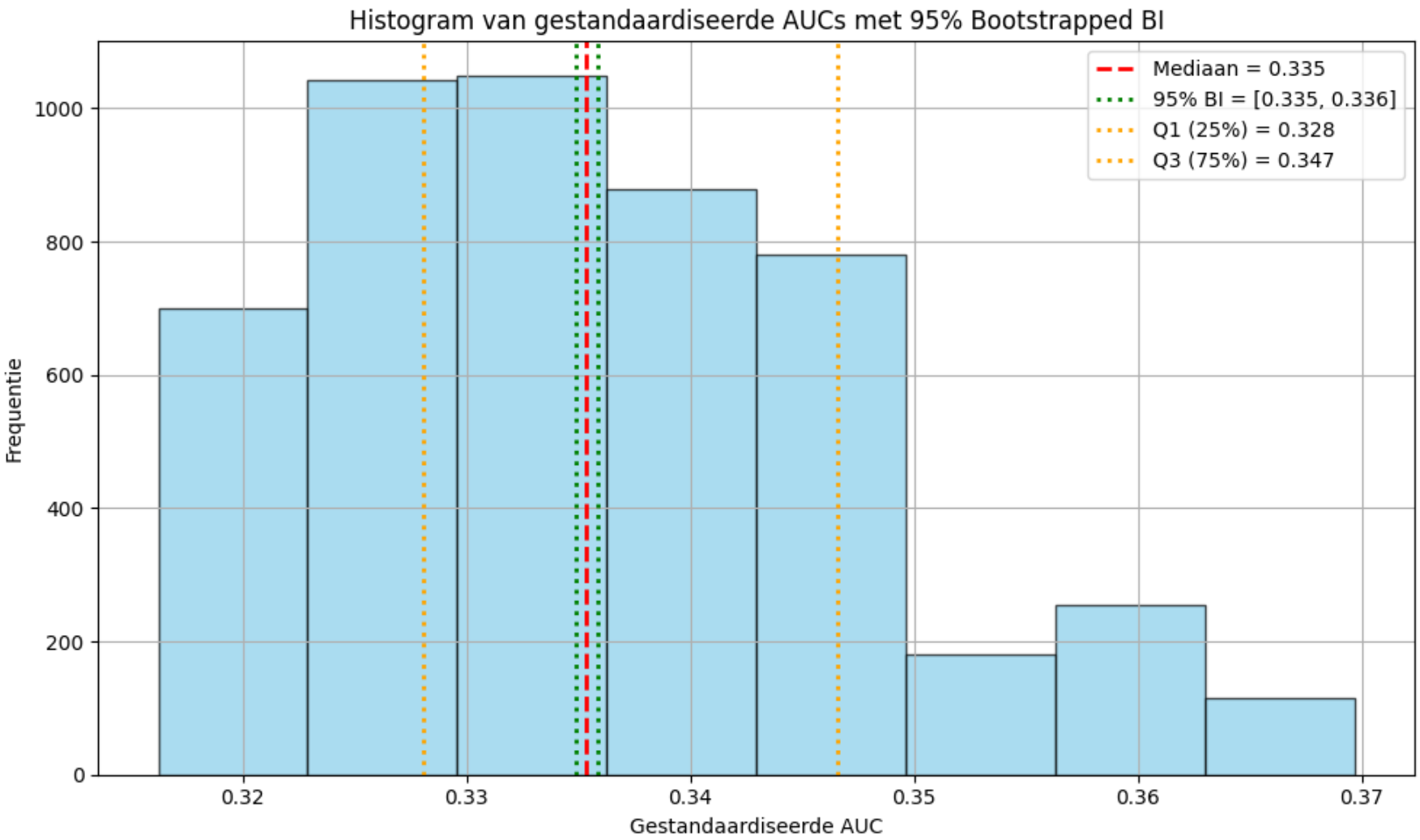
FIGUUR 39: Grafiek van 20% drempelwaarde



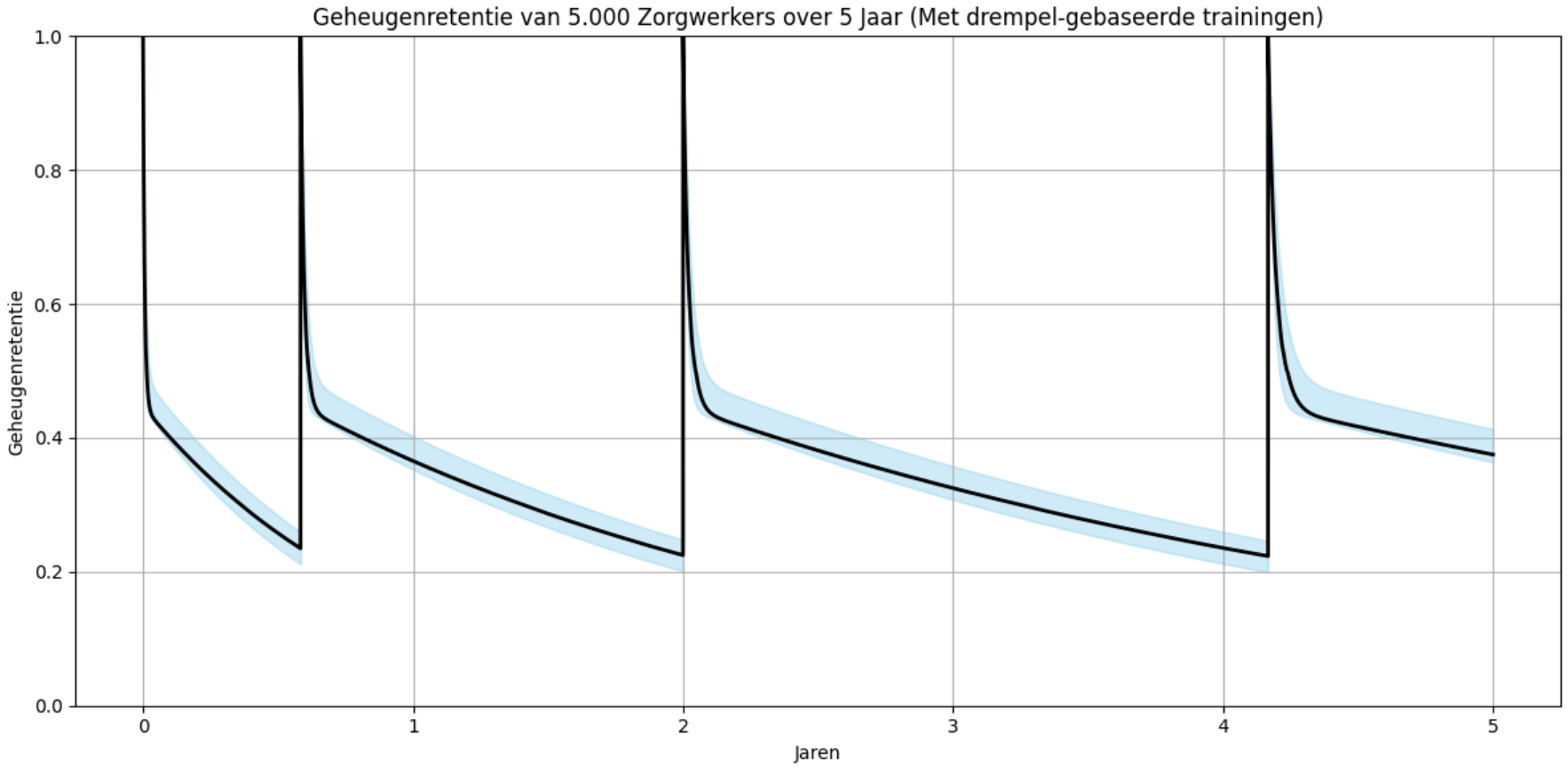
FIGUUR 40: Histogram van 20% drempelwaarde



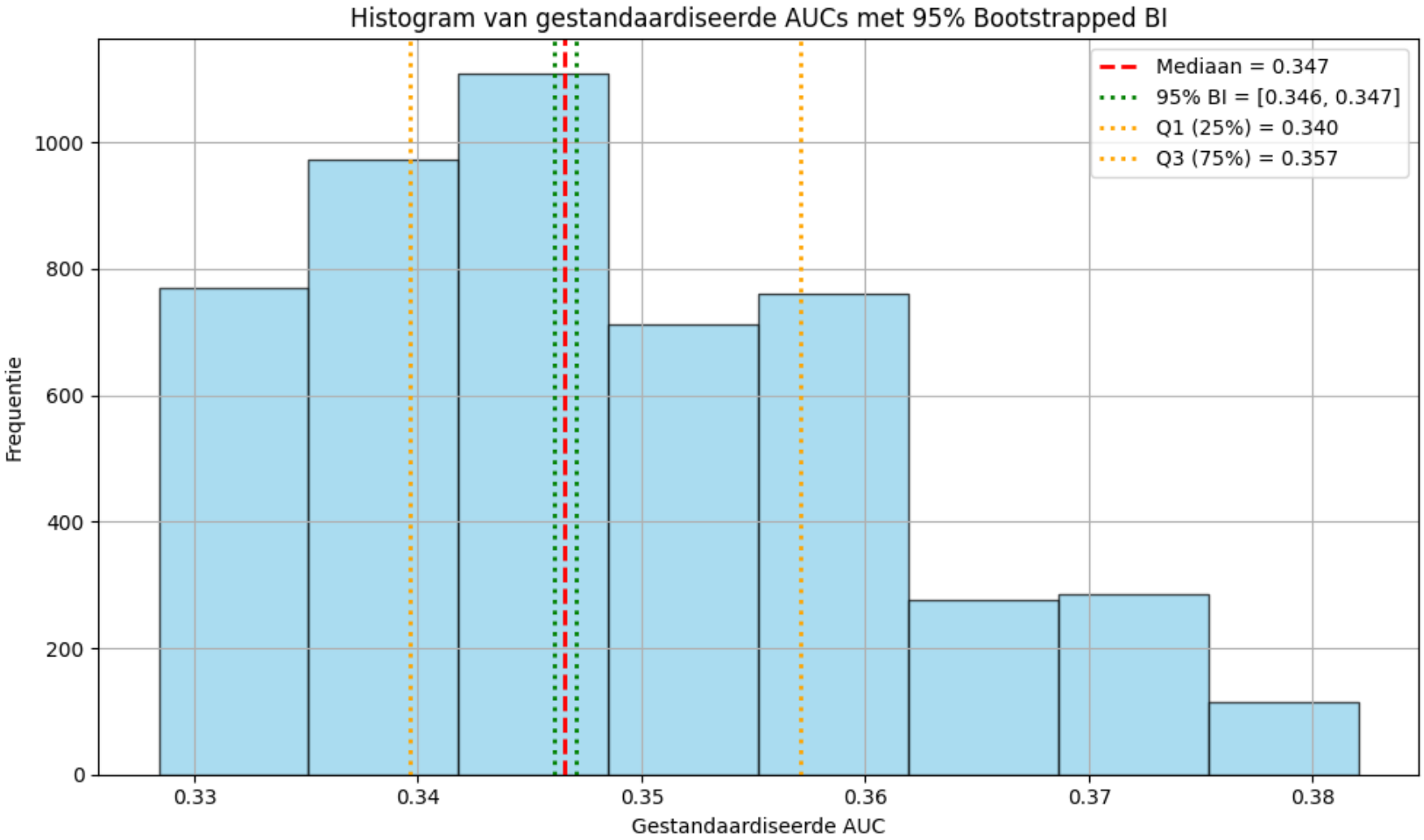
FIGUUR 41: Grafiek van 21% drempelwaarde



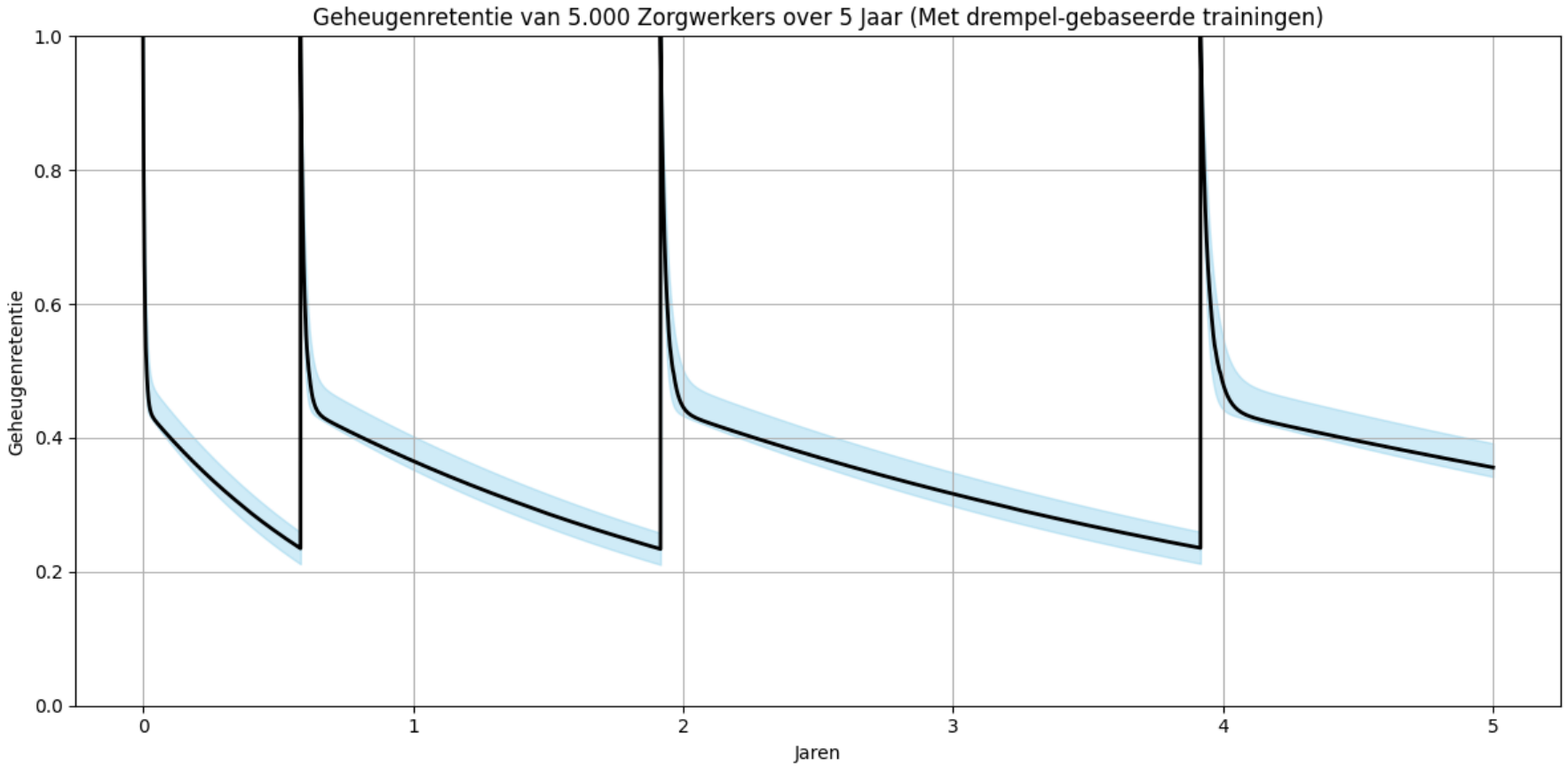
FIGUUR 42: Histogram van 21% drempelwaarde



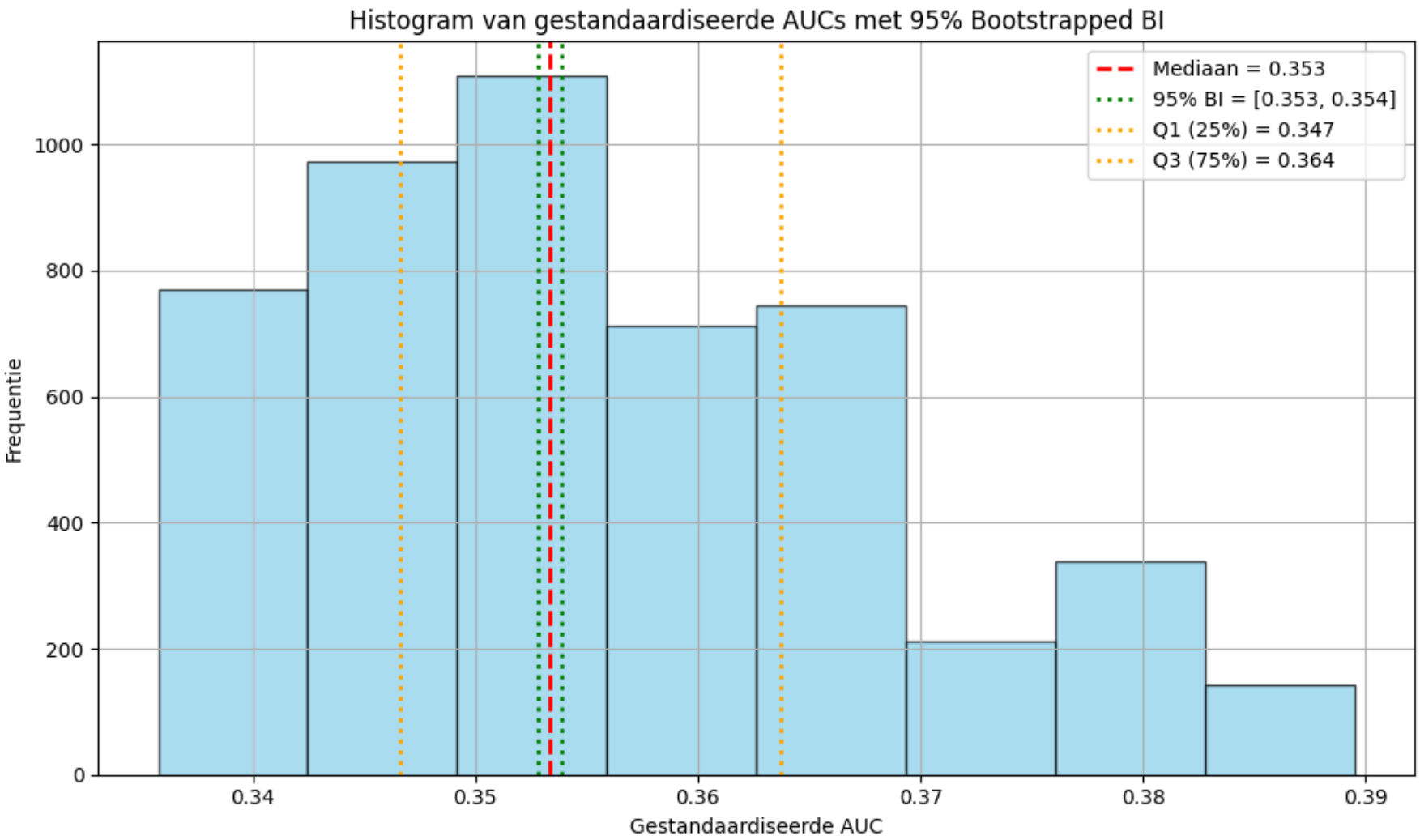
FIGUUR 43: Grafiek van 22% drempelwaarde



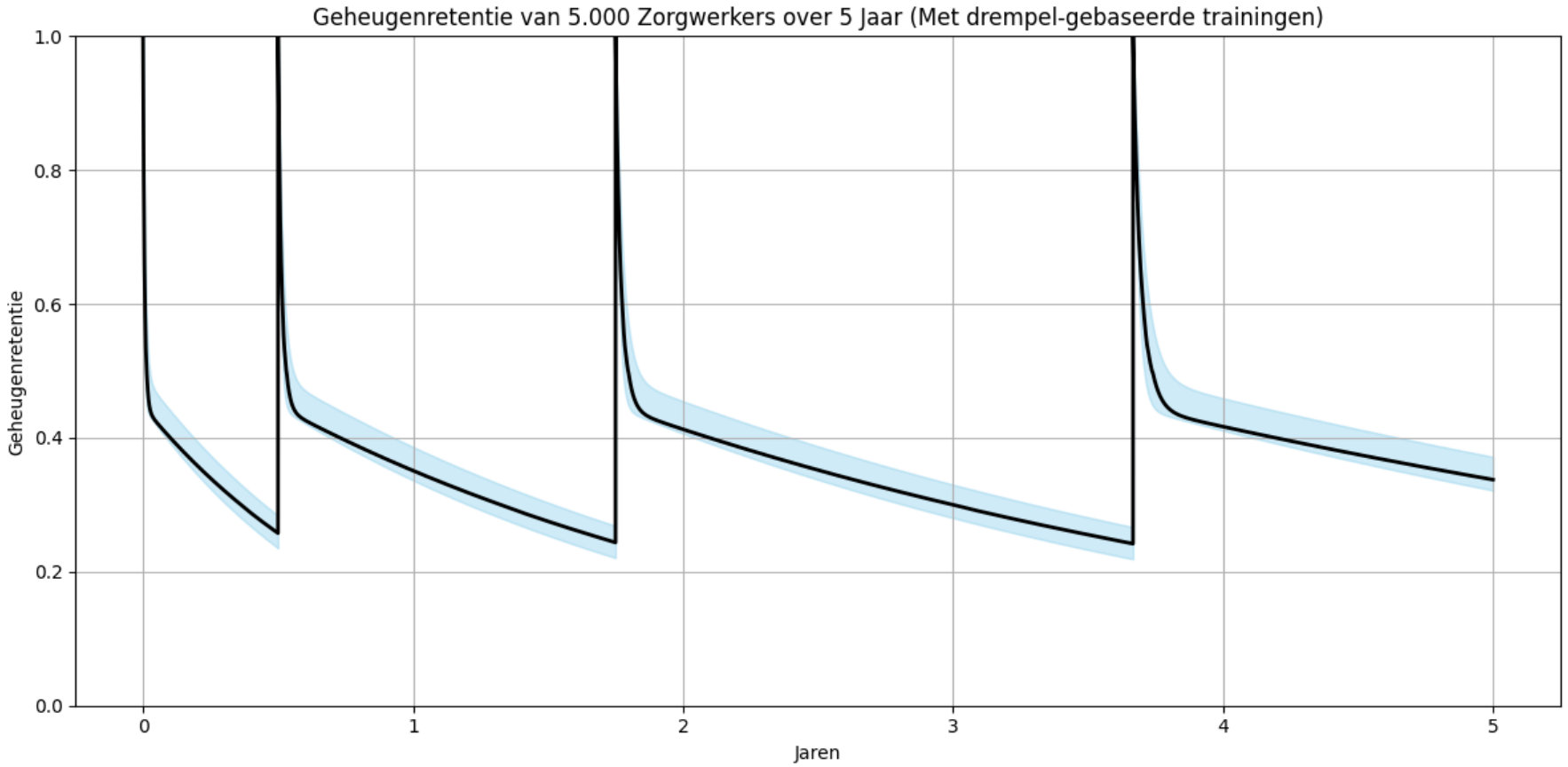
FIGUUR 44: Histogram van 22% drempelwaarde



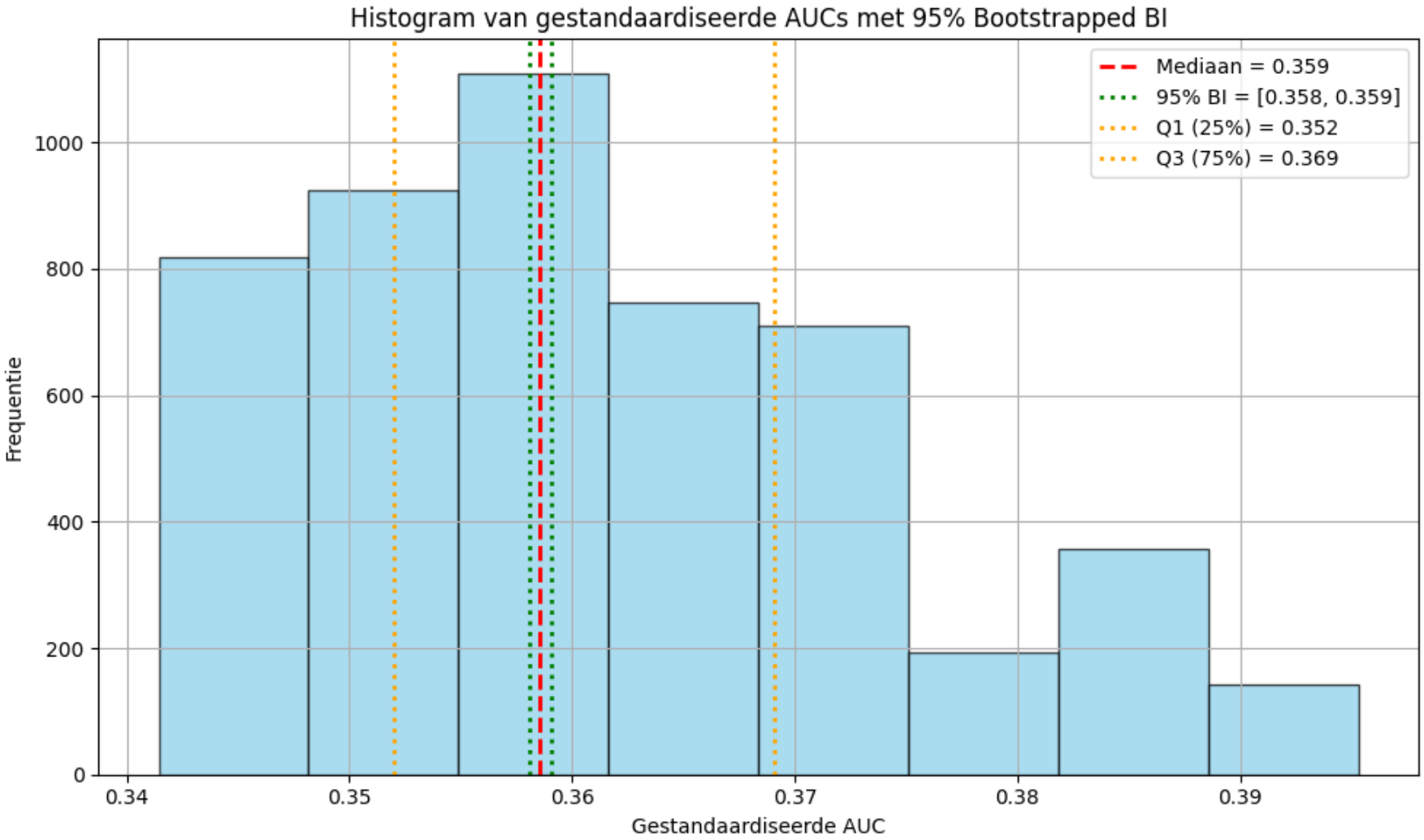
FIGUUR 45: Grafiek van 23% drempelwaarde



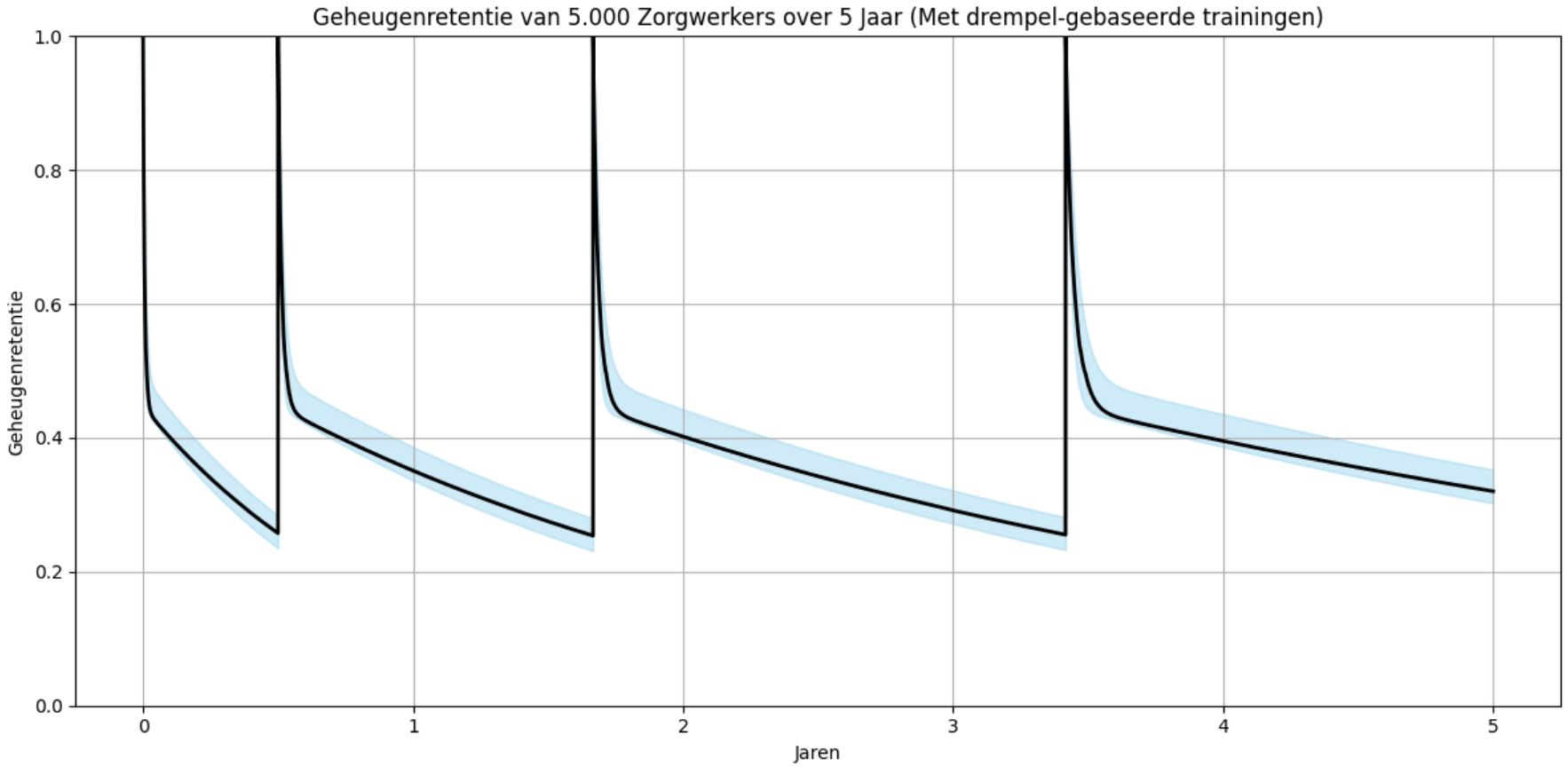
FIGUUR 46: Histogram van 23% drempelwaarde



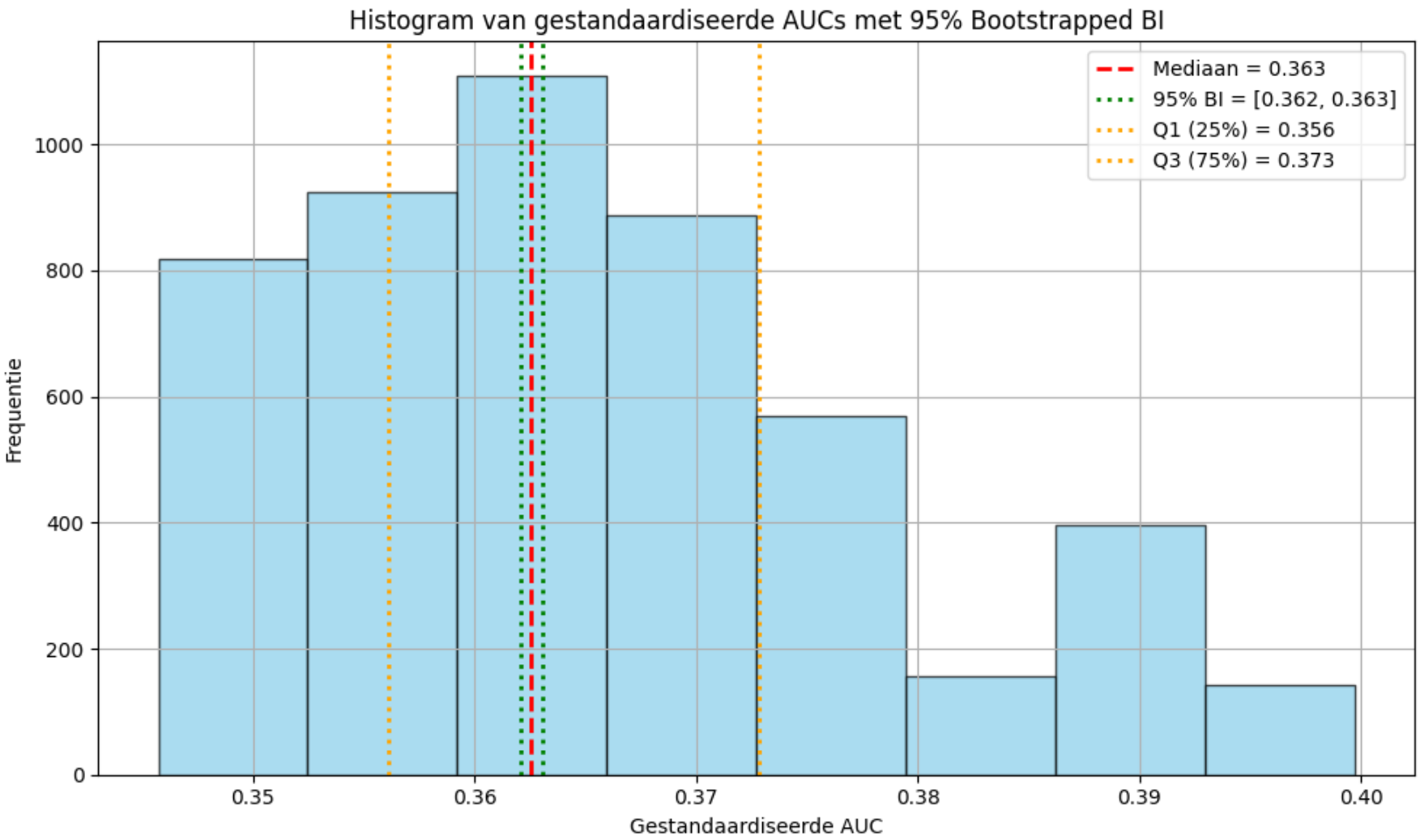
FIGUUR 47: Grafiek van 24% drempelwaarde



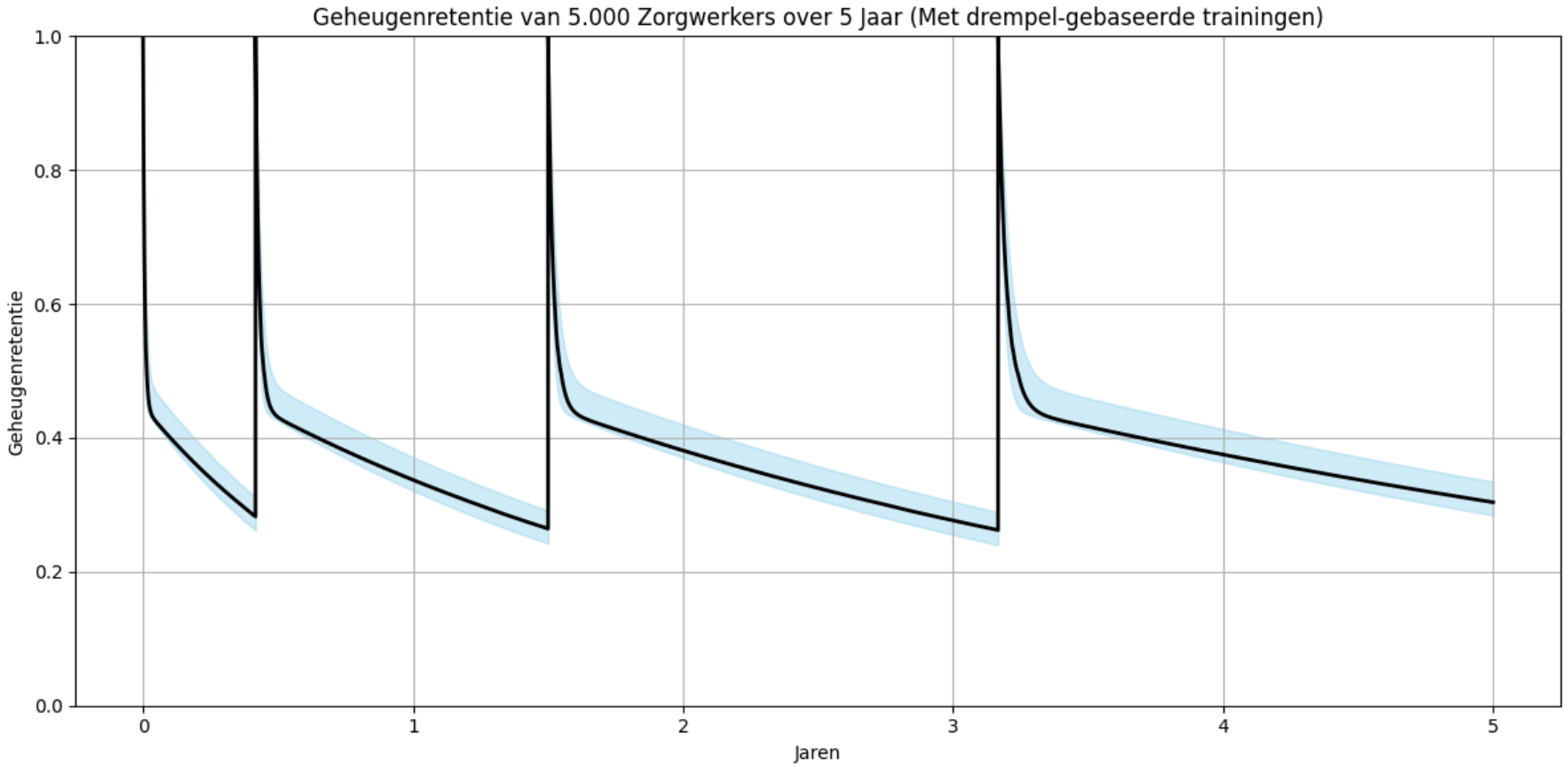
FIGUUR 48: Histogram van 24% drempelwaarde



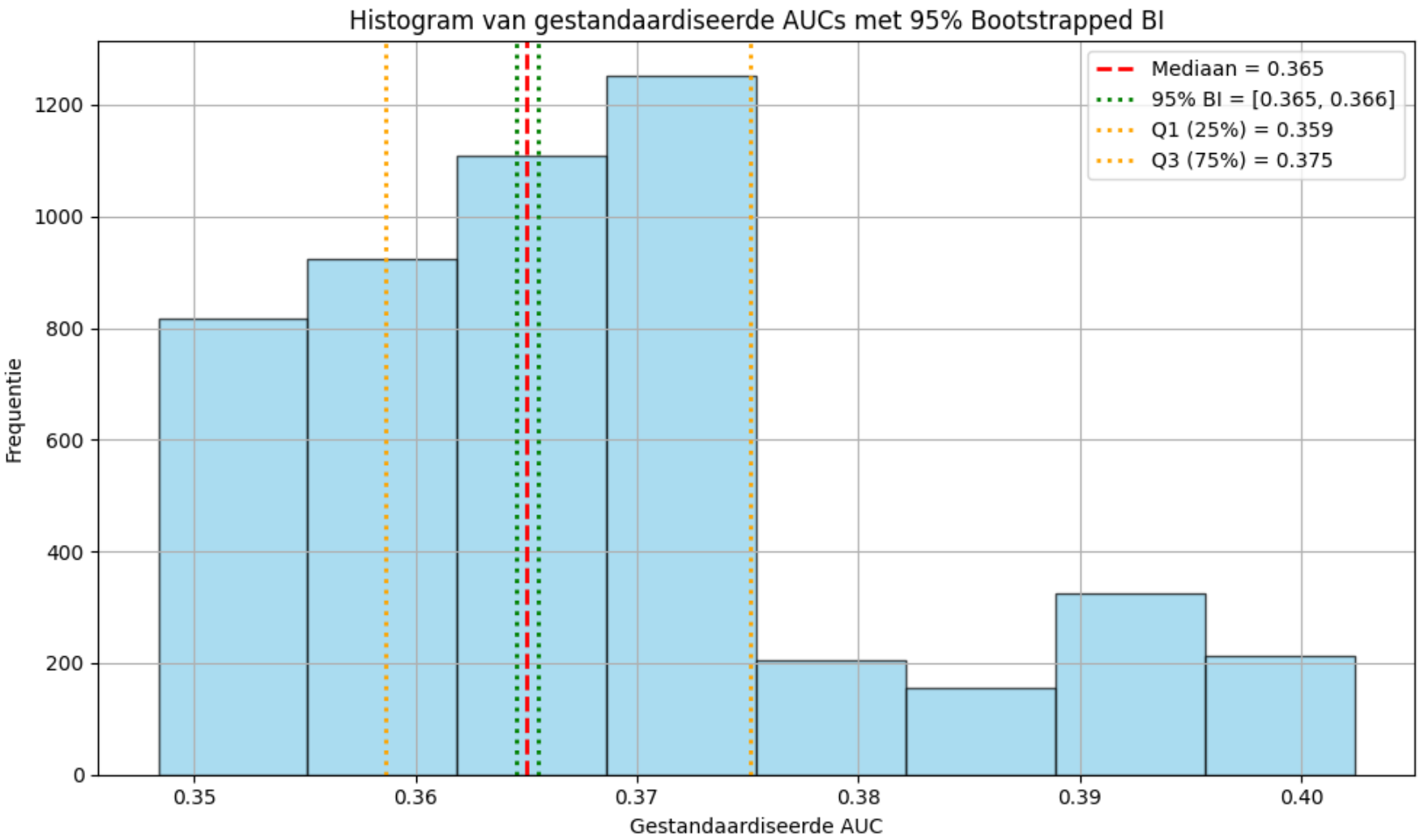
FIGUUR 49: Grafiek van 25% drempelwaarde



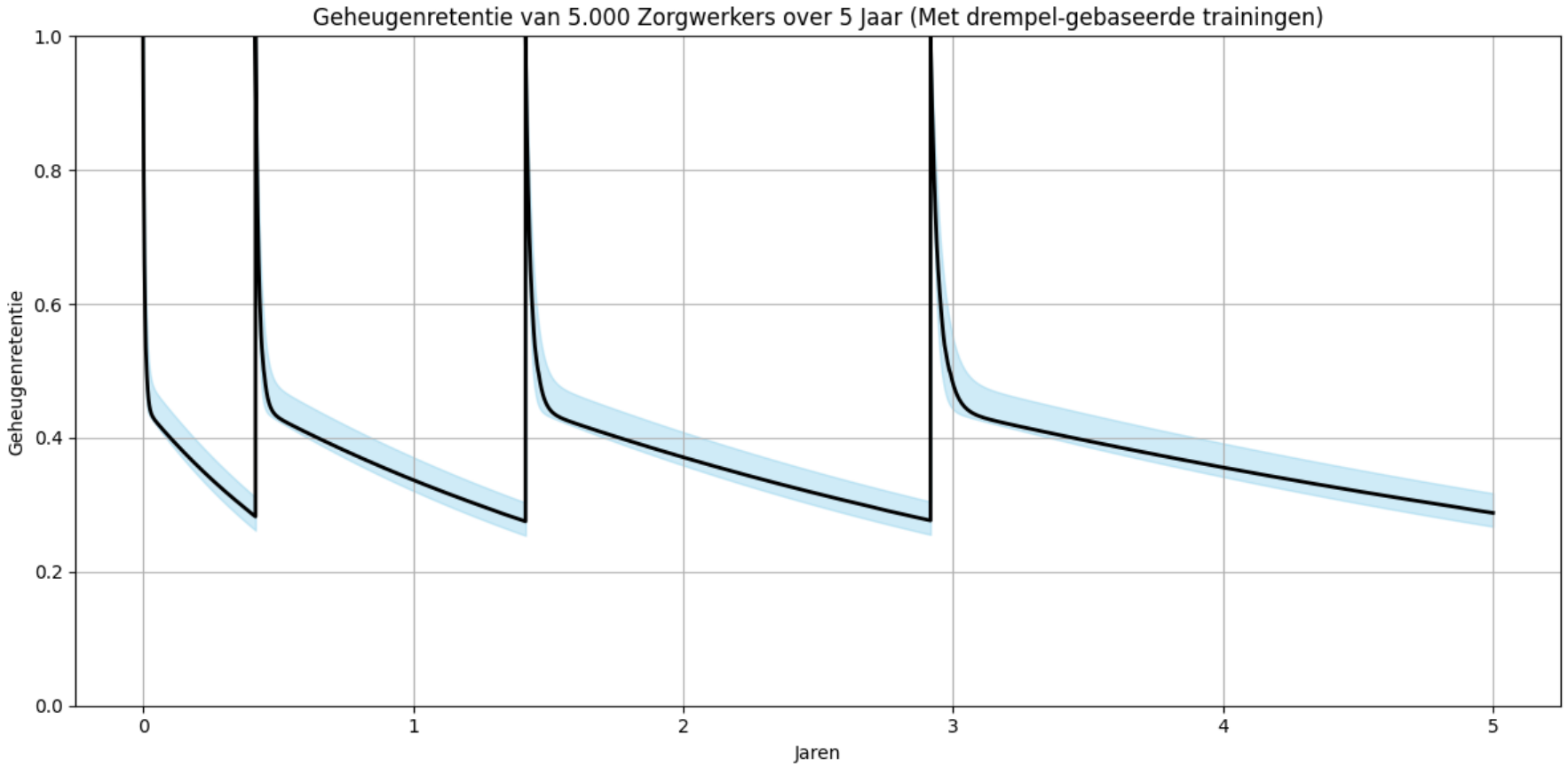
FIGUUR 50: Histogram van 25% drempelwaarde



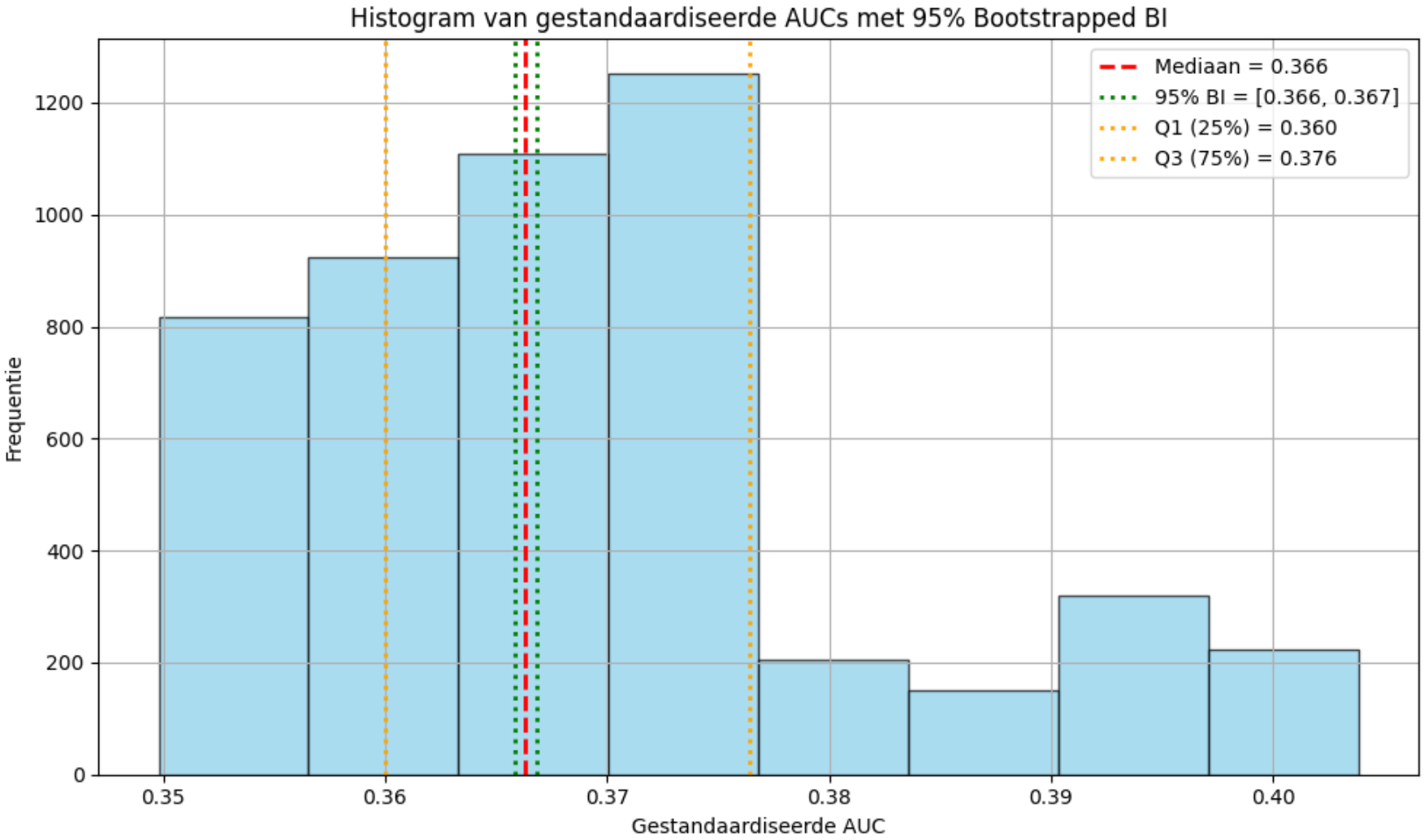
FIGUUR 51: Grafiek van 26% drempelwaarde



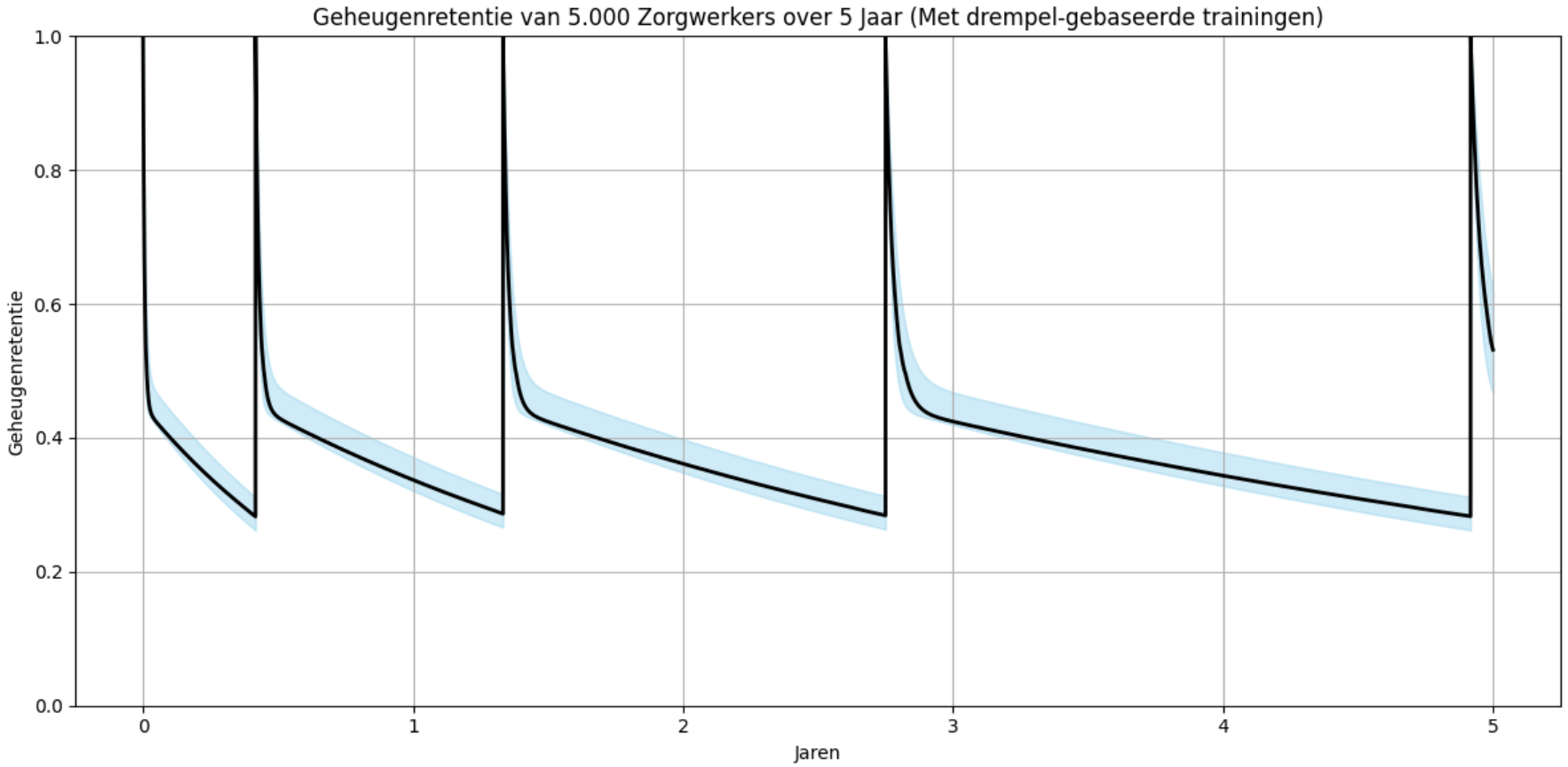
FIGUUR 52: Histogram van 26% drempelwaarde



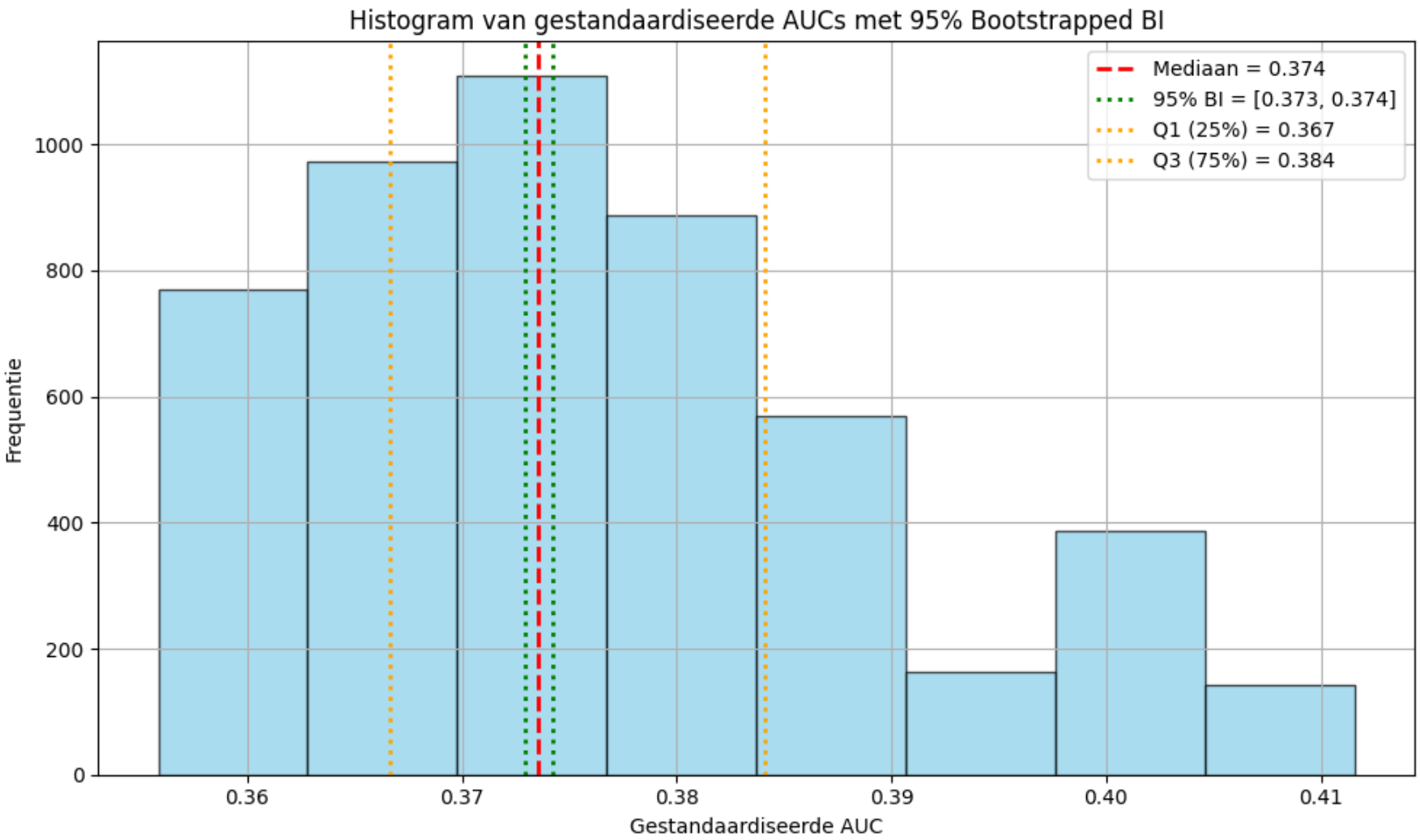
FIGUUR 53: Grafiek van 27% drempelwaarde



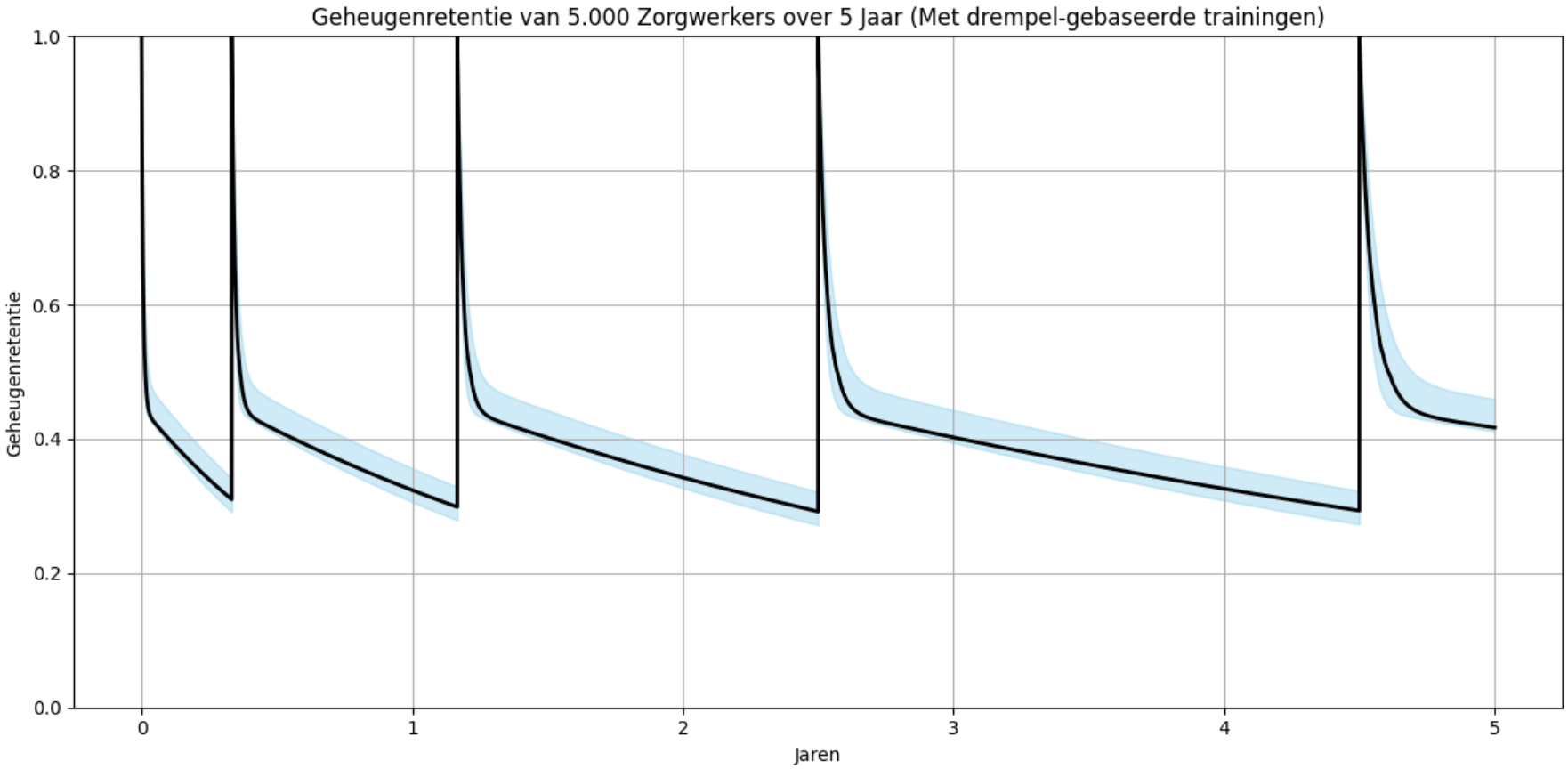
FIGUUR 54: Histogram van 27% drempelwaarde



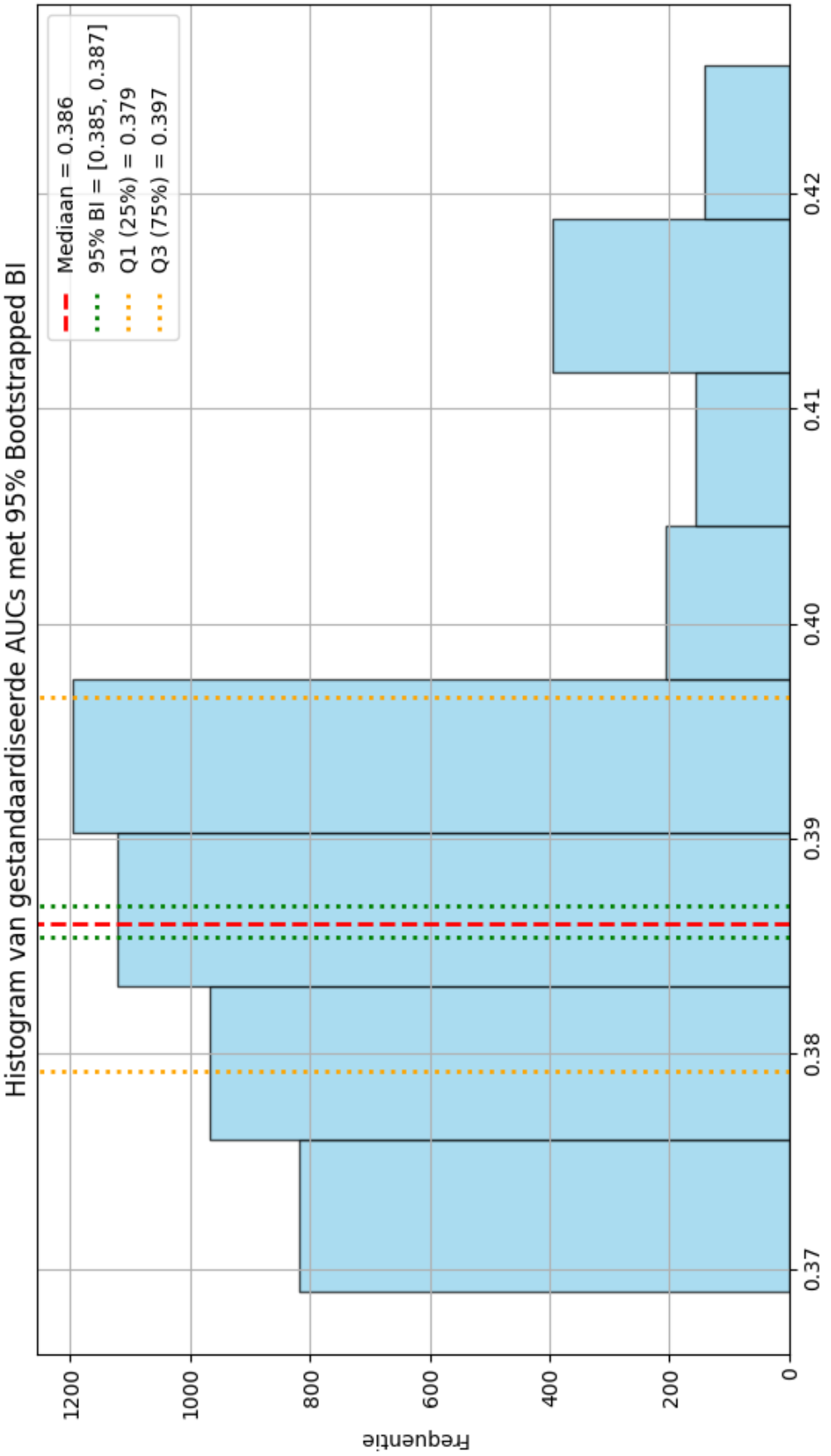
FIGUUR 55: Grafiek van 28% drempelwaarde



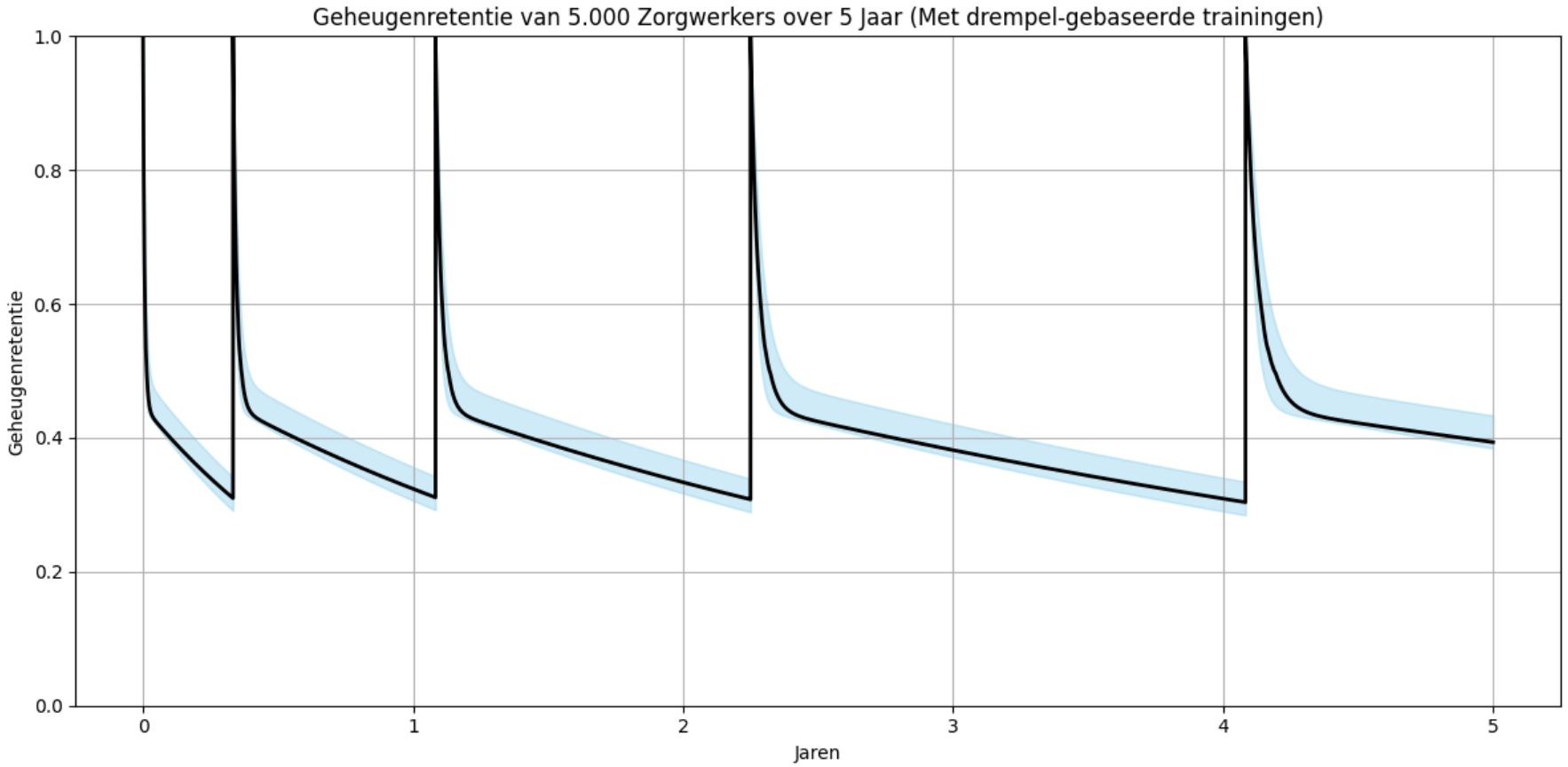
FIGUUR 56: Histogram van 28% drempelwaarde



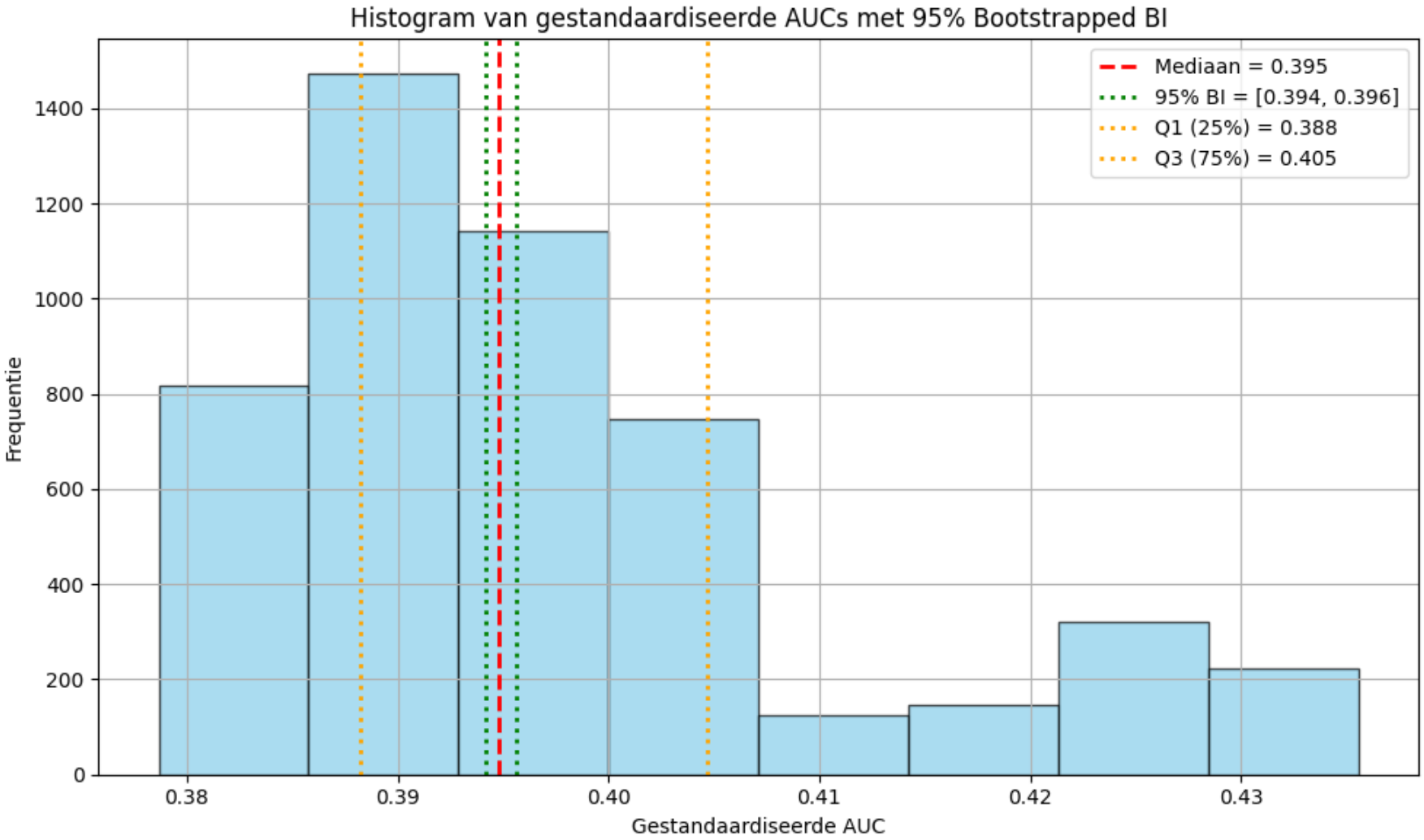
FIGUUR 57: Grafiek van 29% drempelwaarde



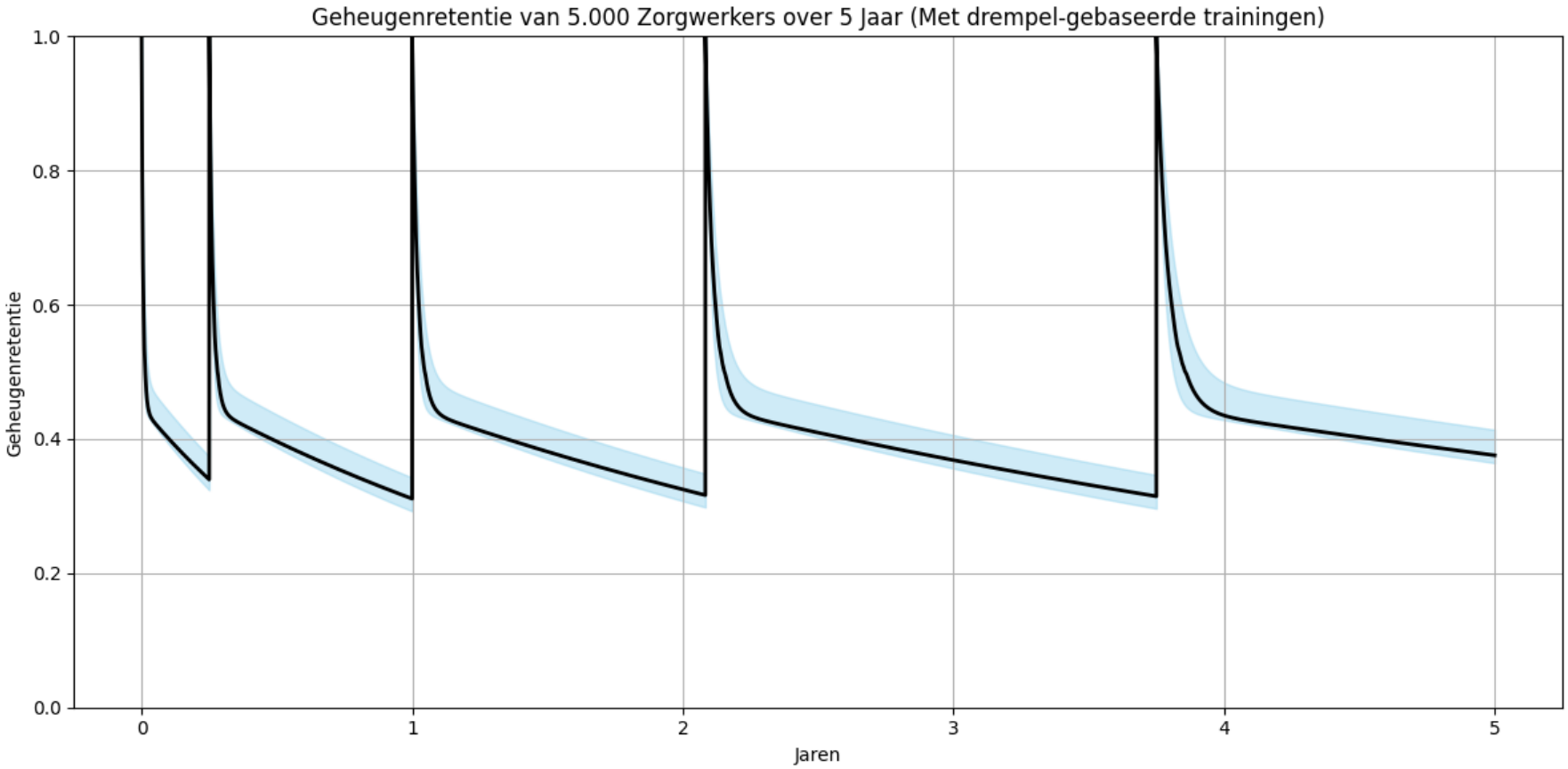
FIGUUR 58: Histogram van 29% drempelwaarde



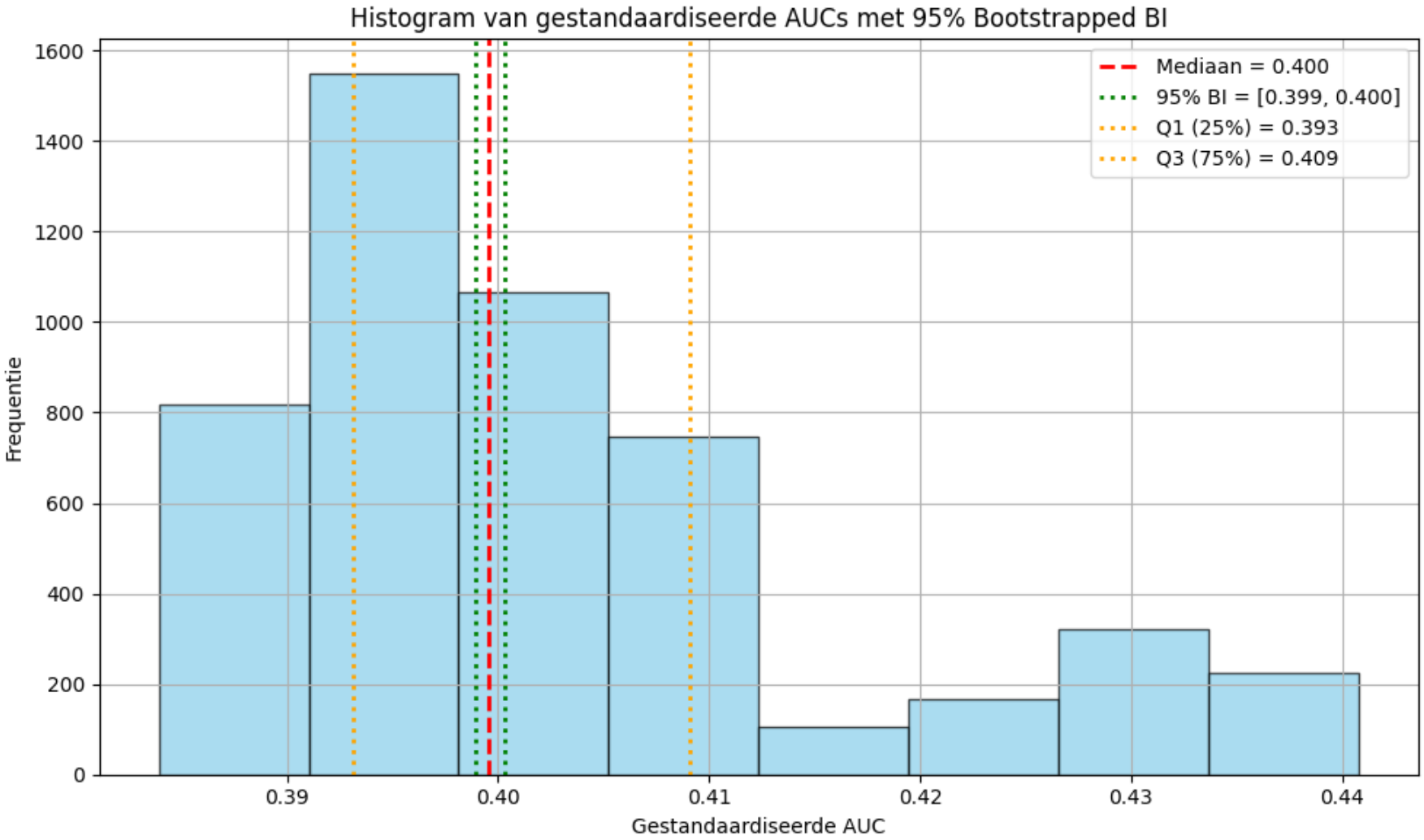
FIGUUR 59: Grafiek van 30% drempelwaarde



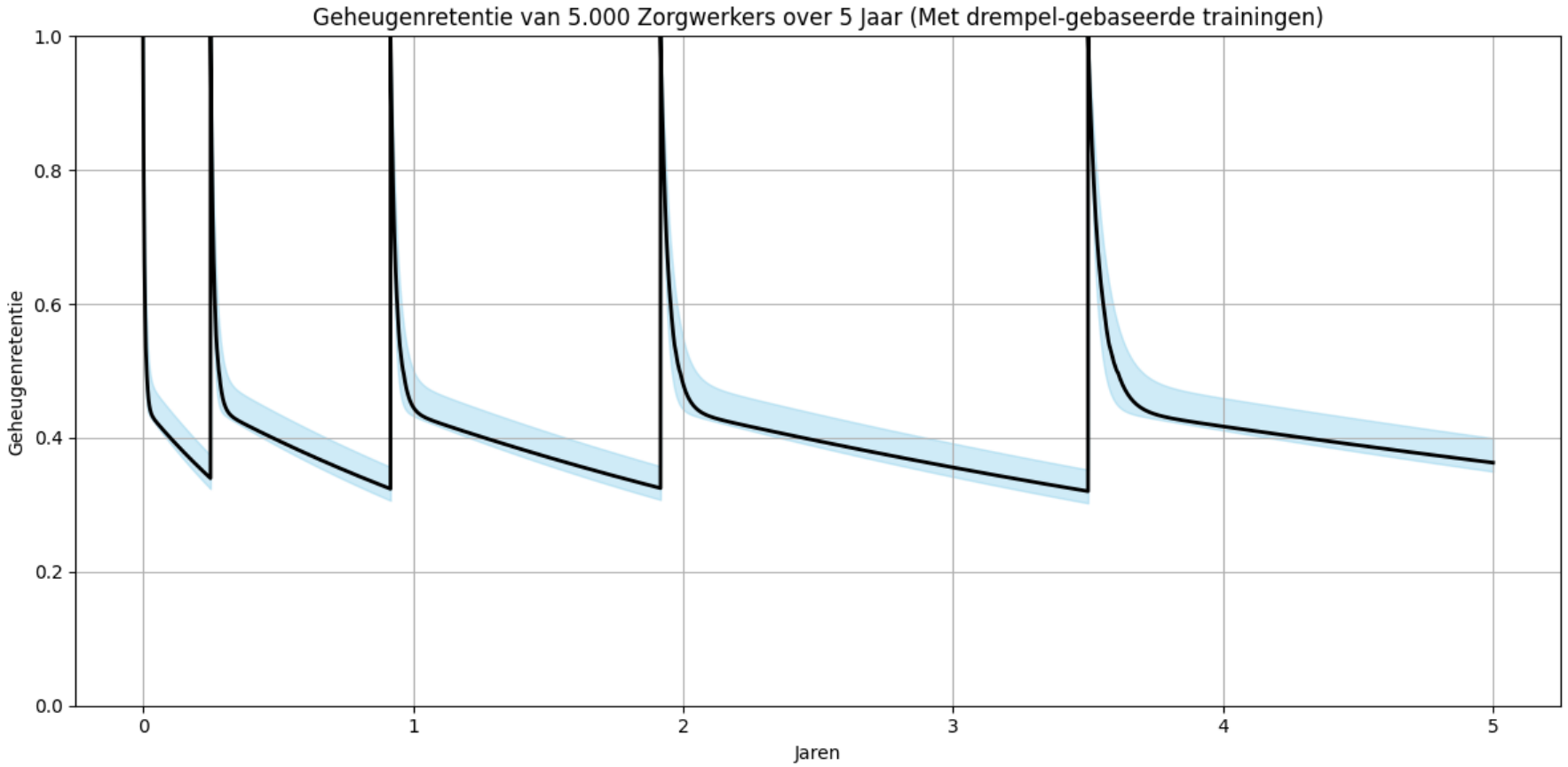
FIGUUR 60: Histogram van 30% drempelwaarde



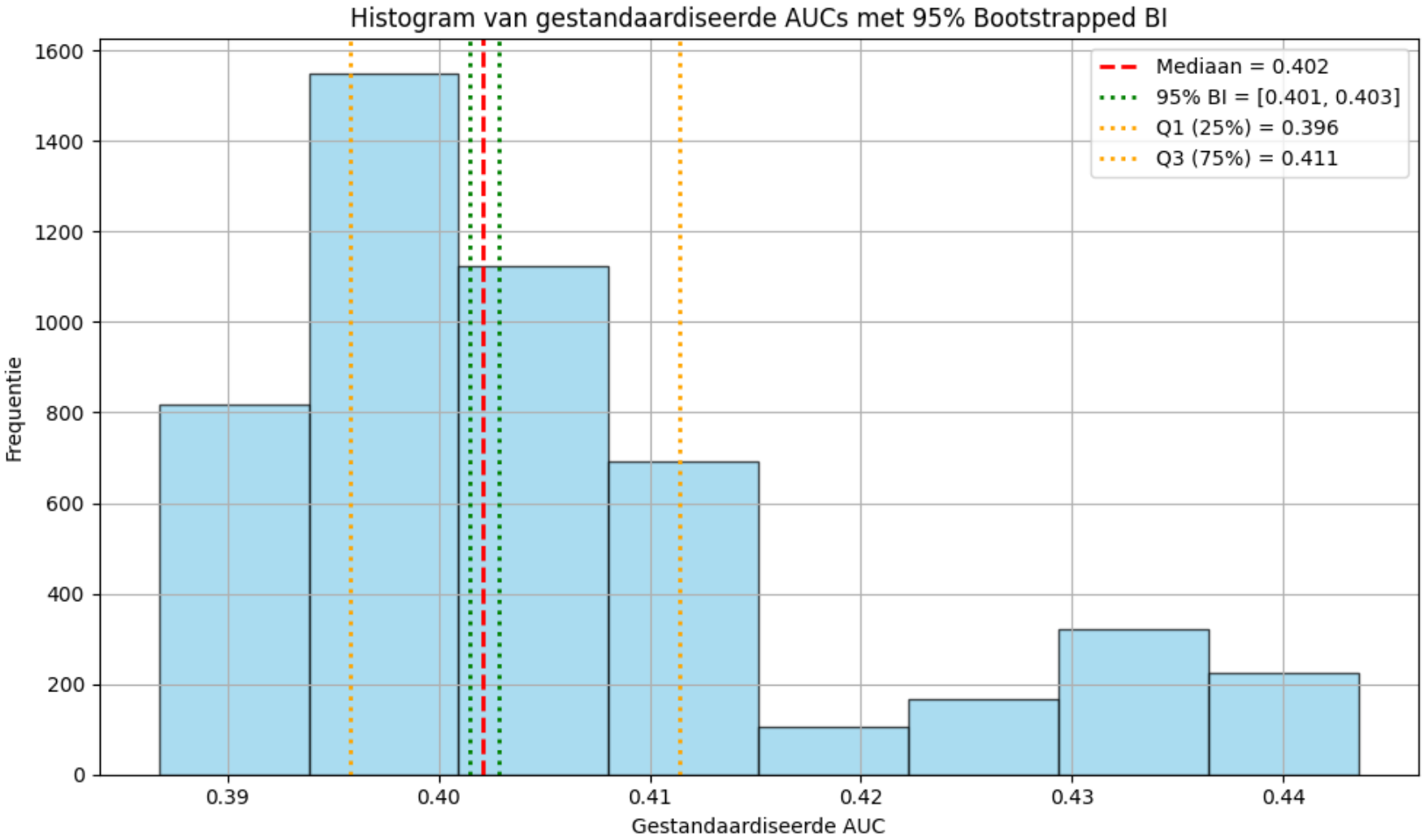
FIGUUR 61: Grafiek van 31% drempelwaarde



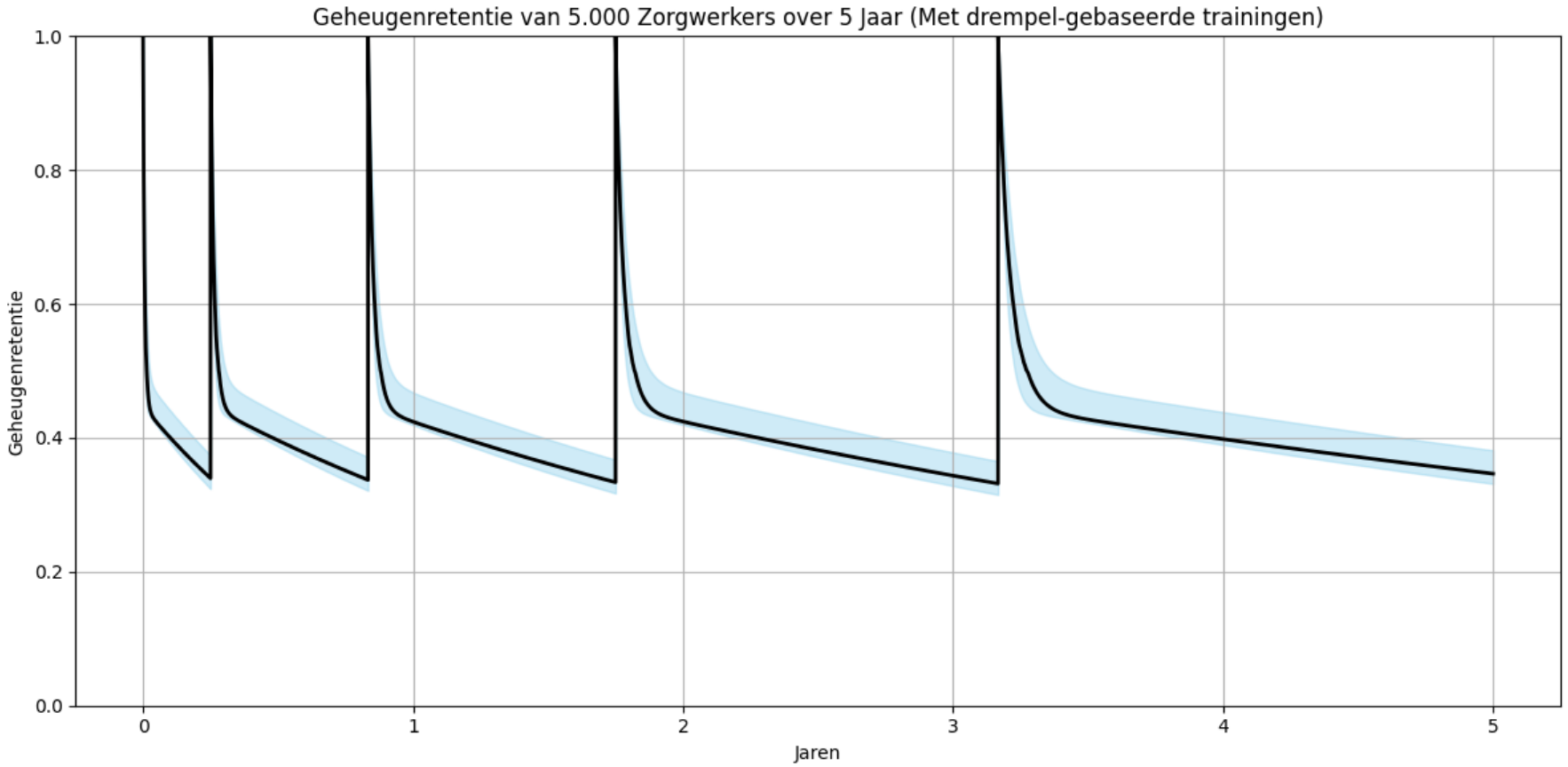
FIGUUR 62: Histogram van 31% drempelwaarde



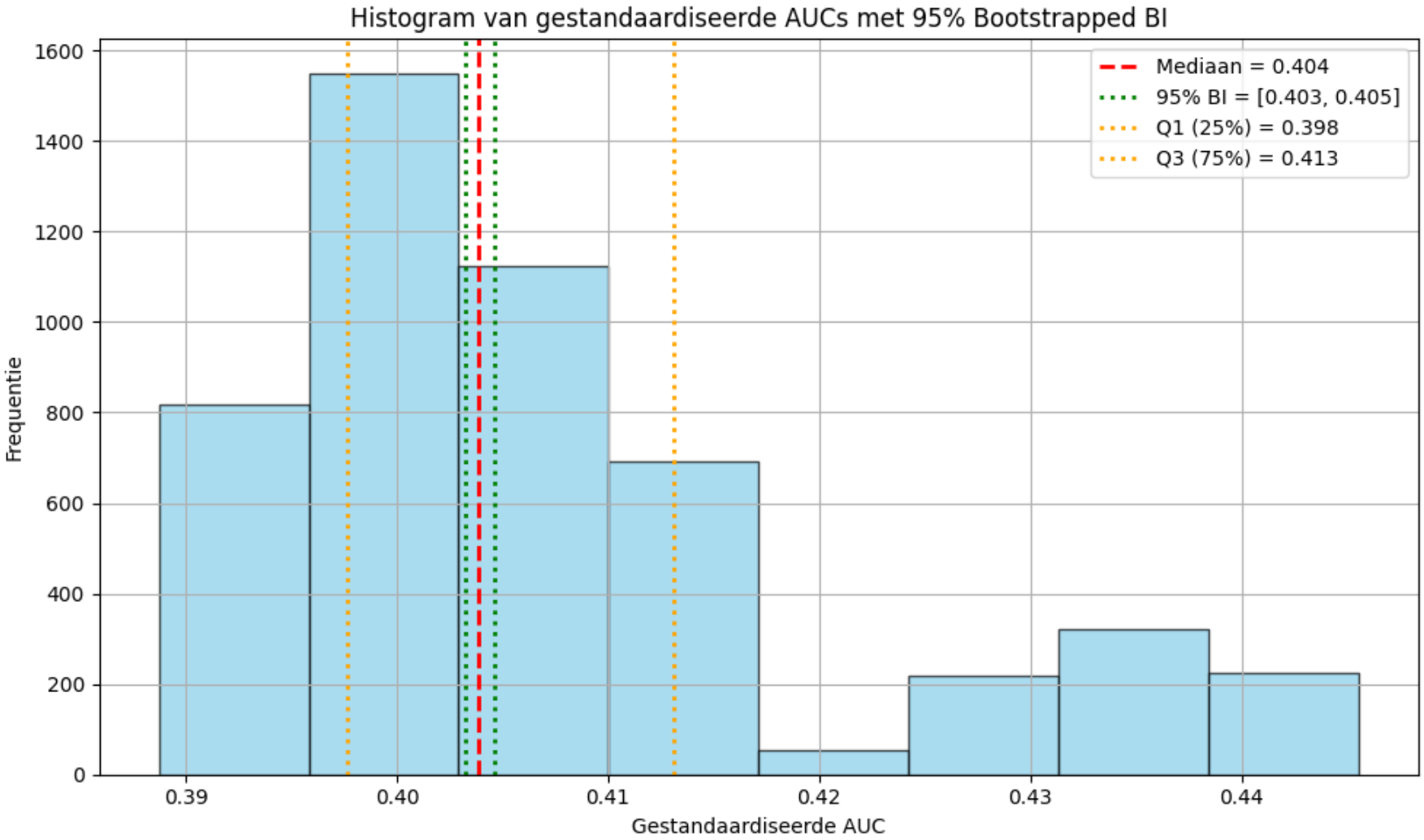
FIGUUR 63: Grafiek van 32% drempelwaarde



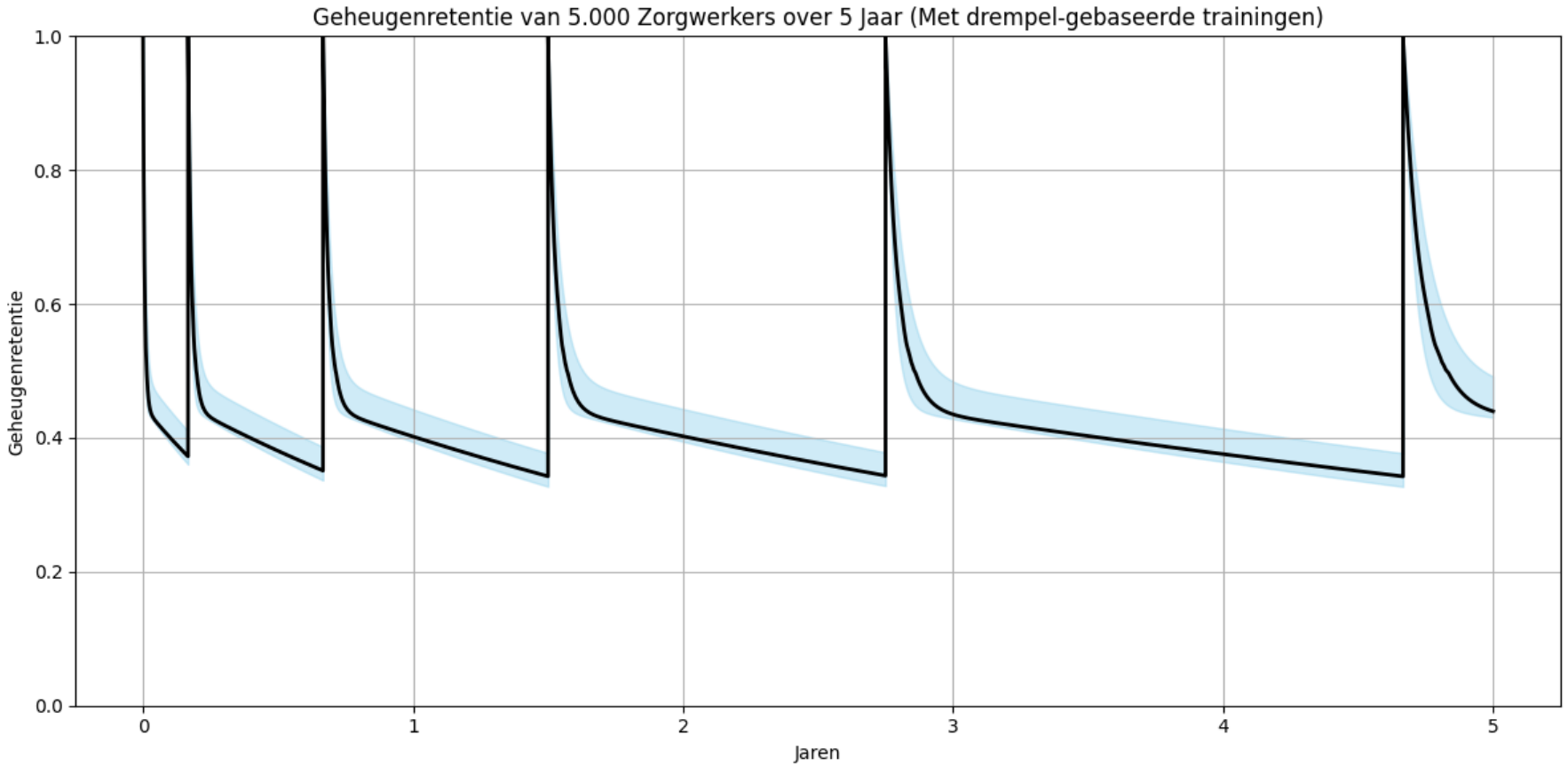
FIGUUR 64: Histogram van 32% drempelwaarde



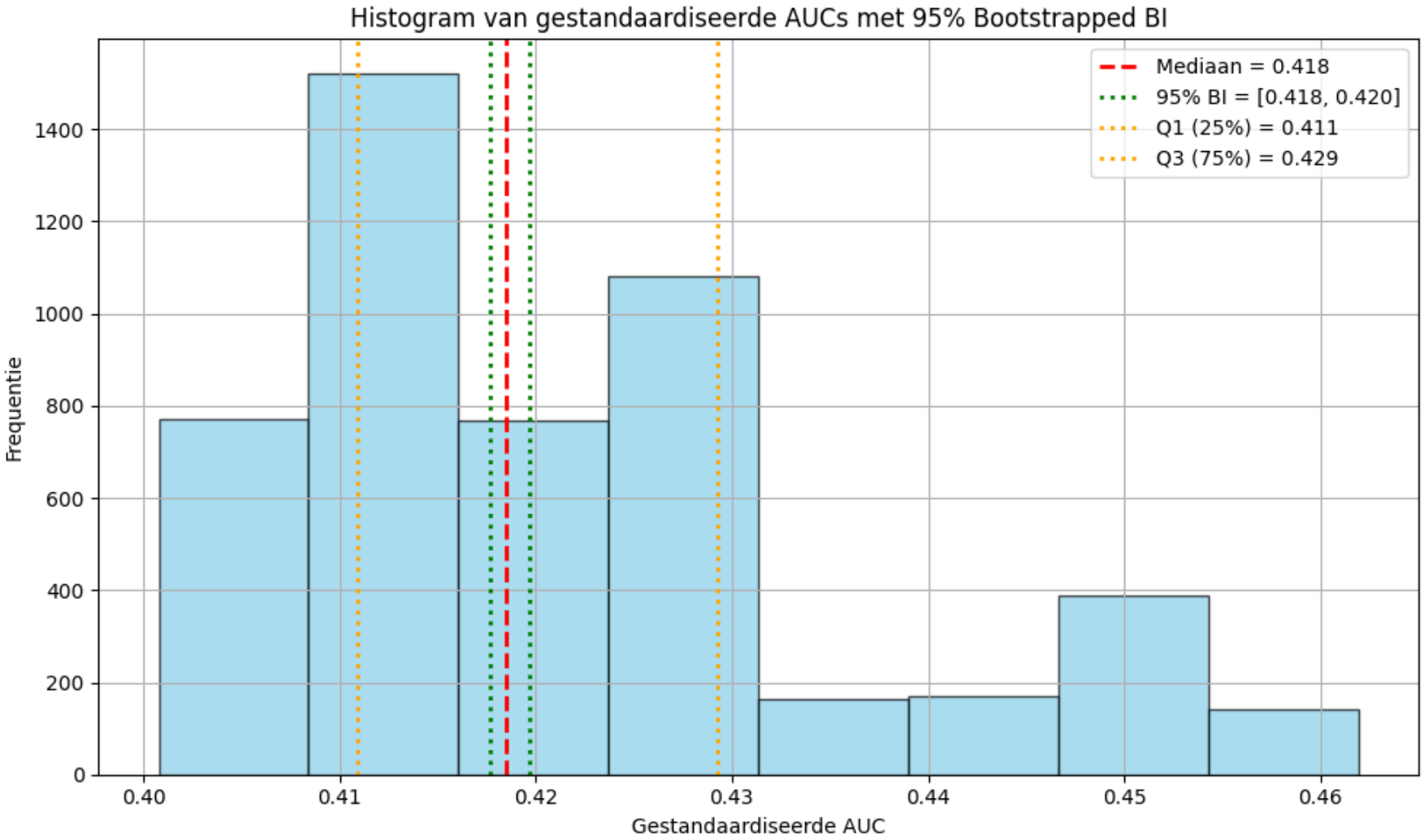
FIGUUR 65: Grafiek van 33% drempelwaarde



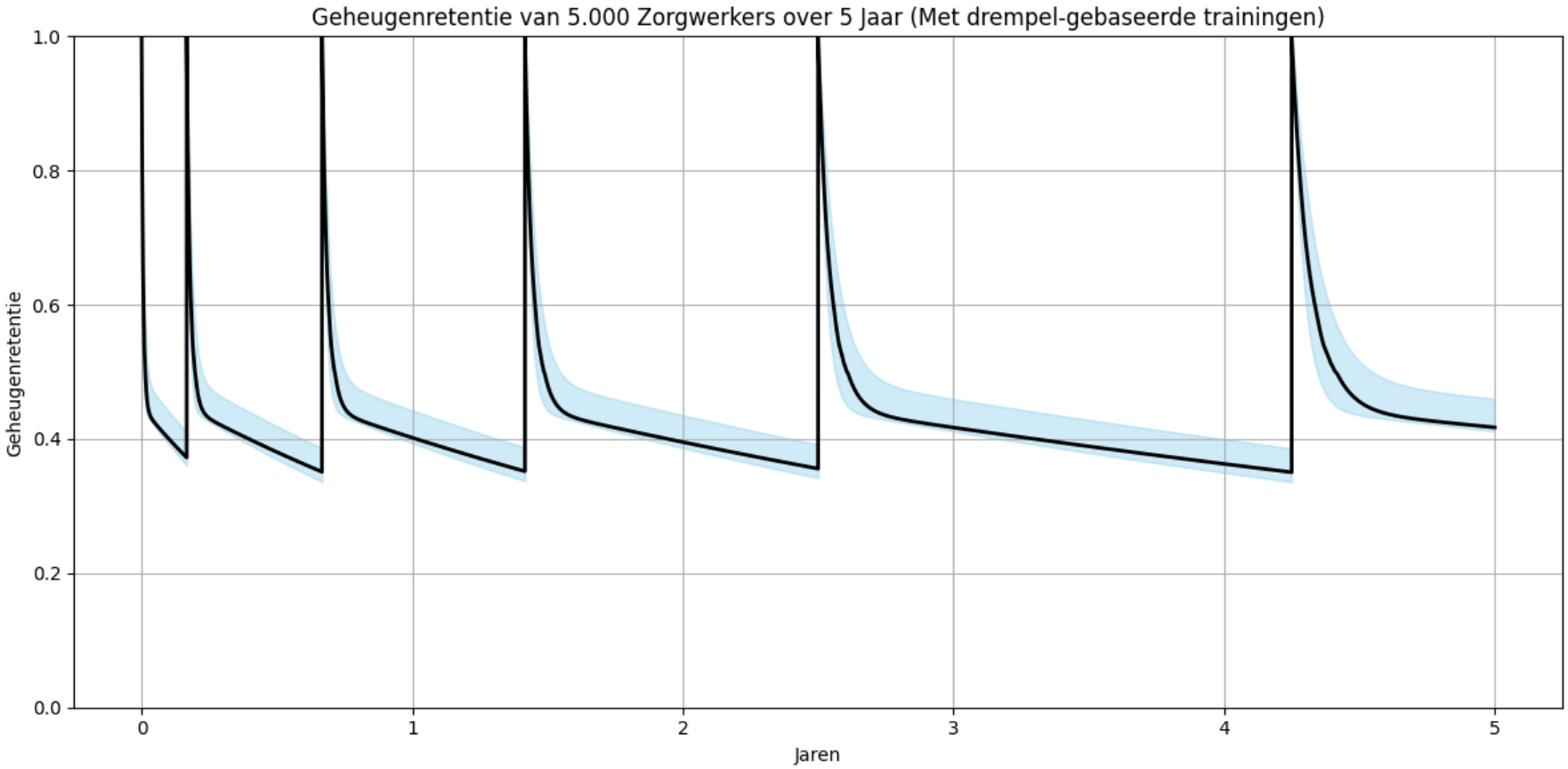
FIGUUR 66: Histogram van 33% drempelwaarde



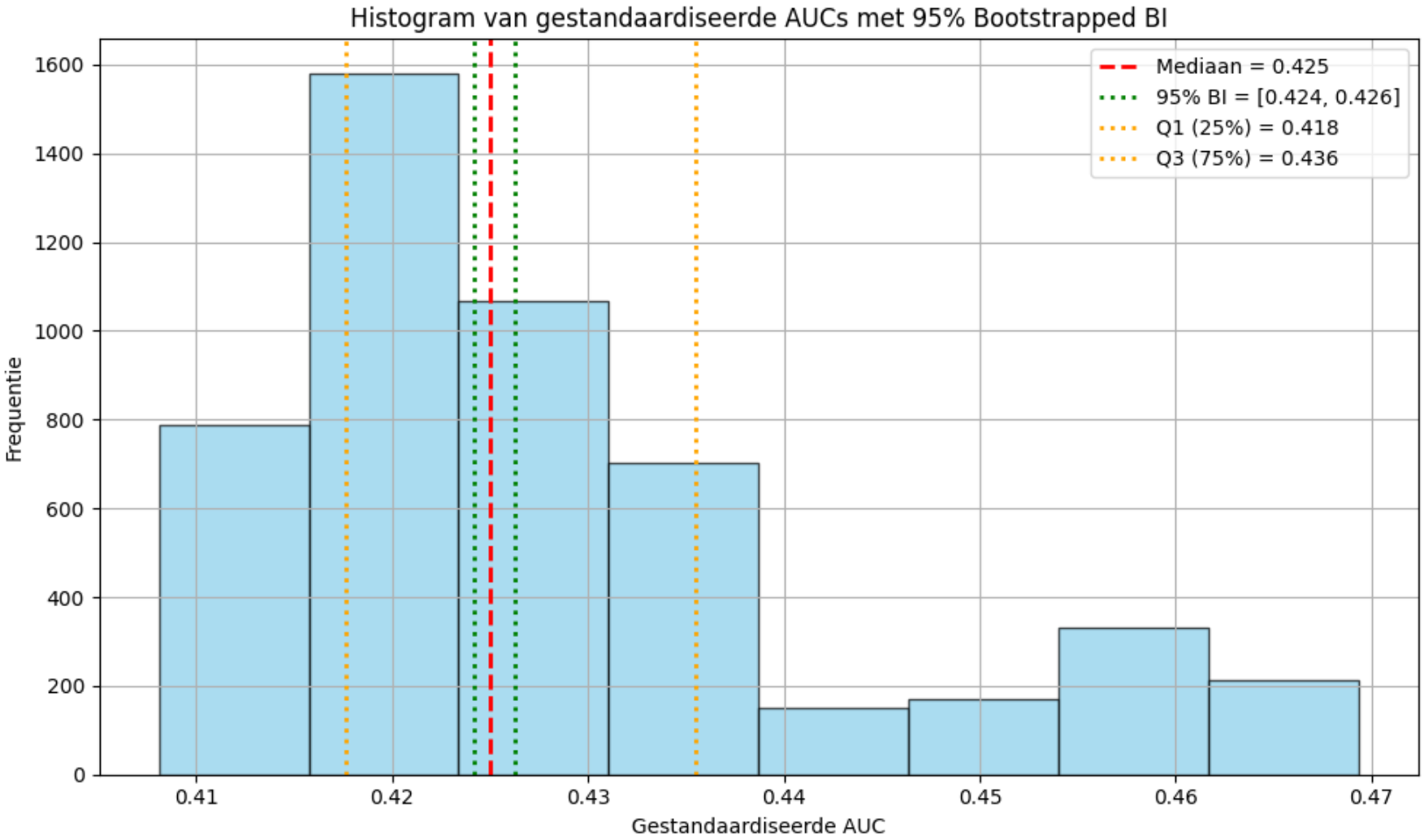
FIGUUR 67: Grafiek van 34% drempelwaarde



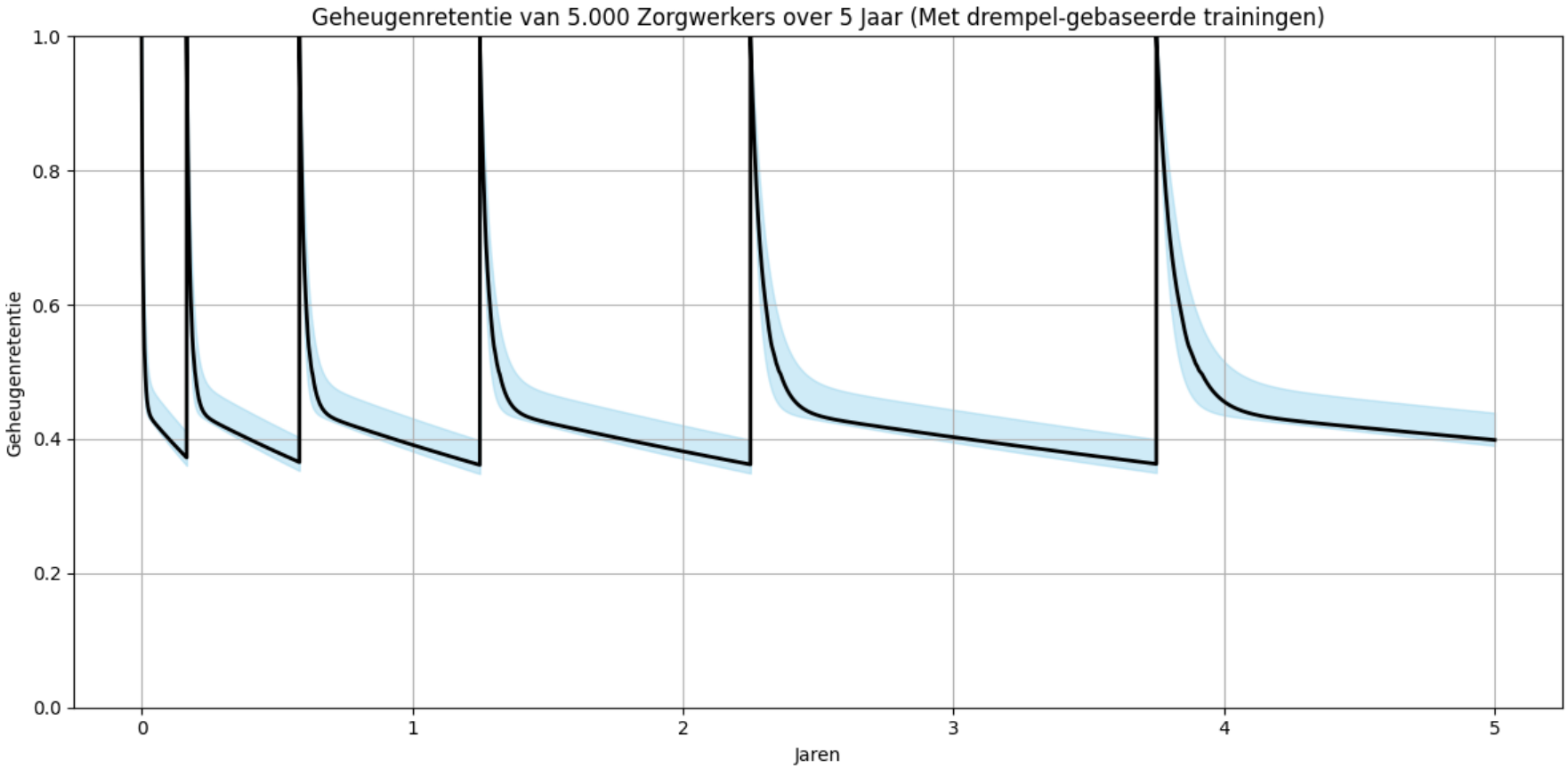
FIGUUR 68: Histogram van 34% drempelwaarde



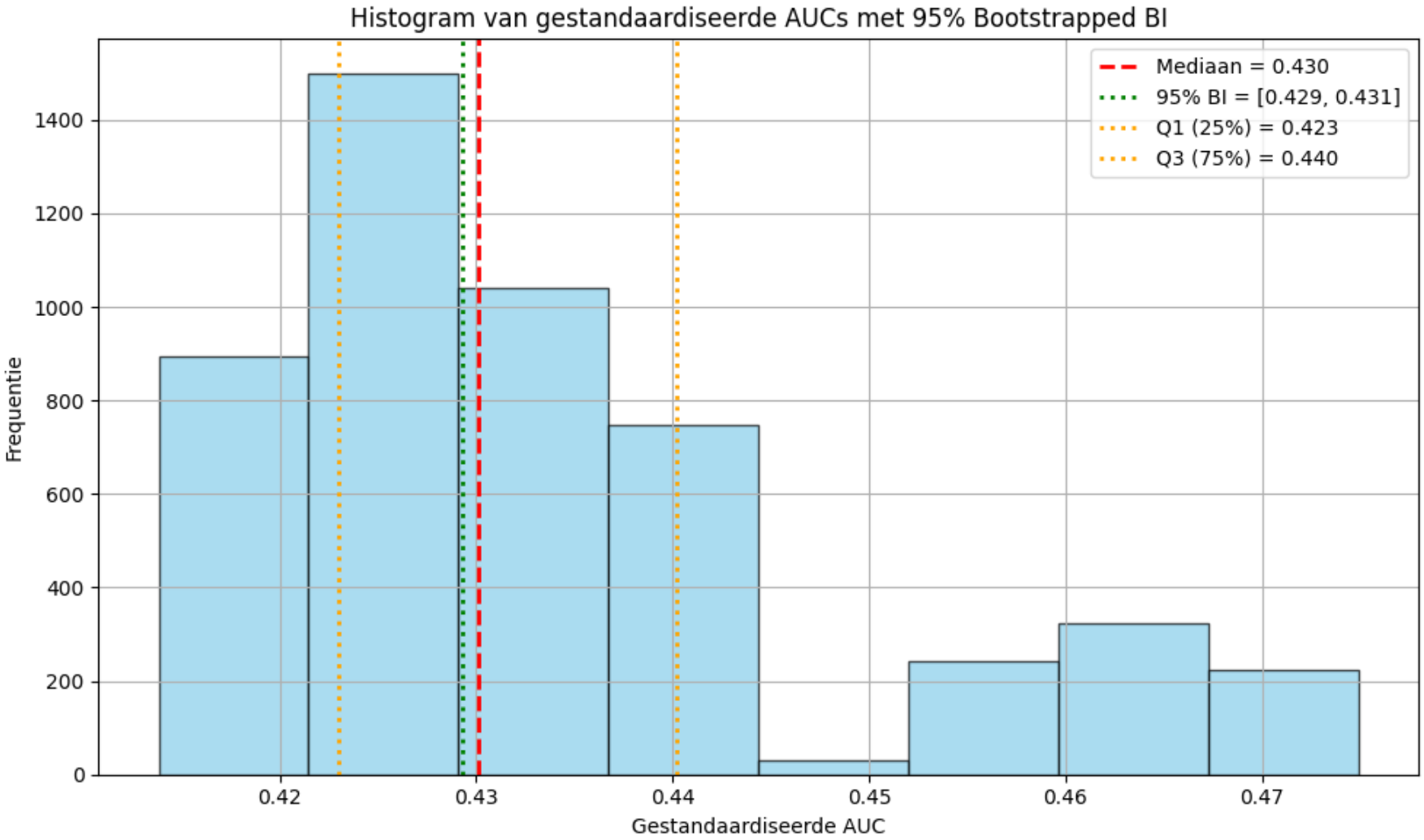
FIGUUR 69: Grafiek van 35% drempelwaarde



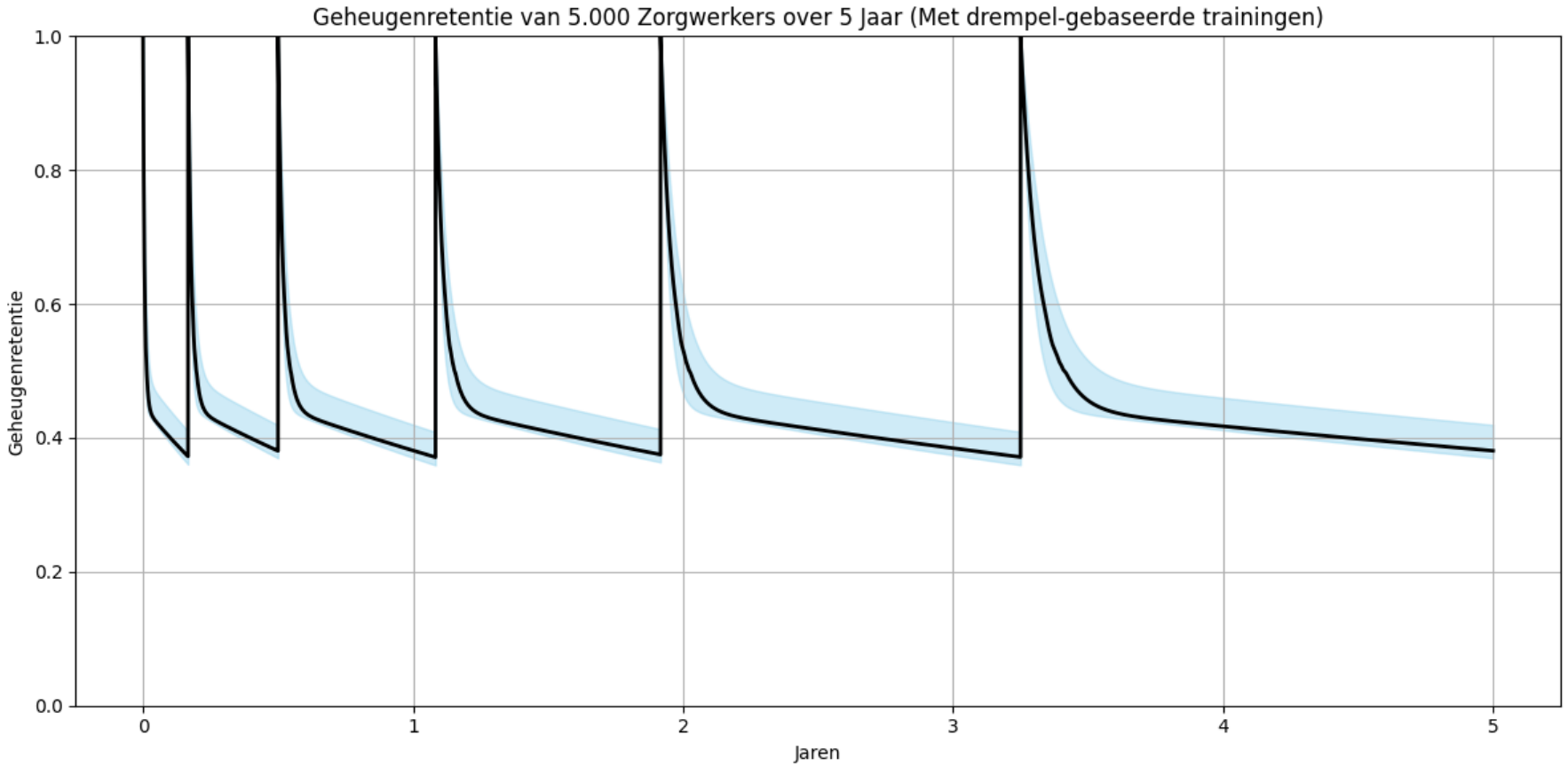
FIGUUR 70: Histogram van 35% drempelwaarde



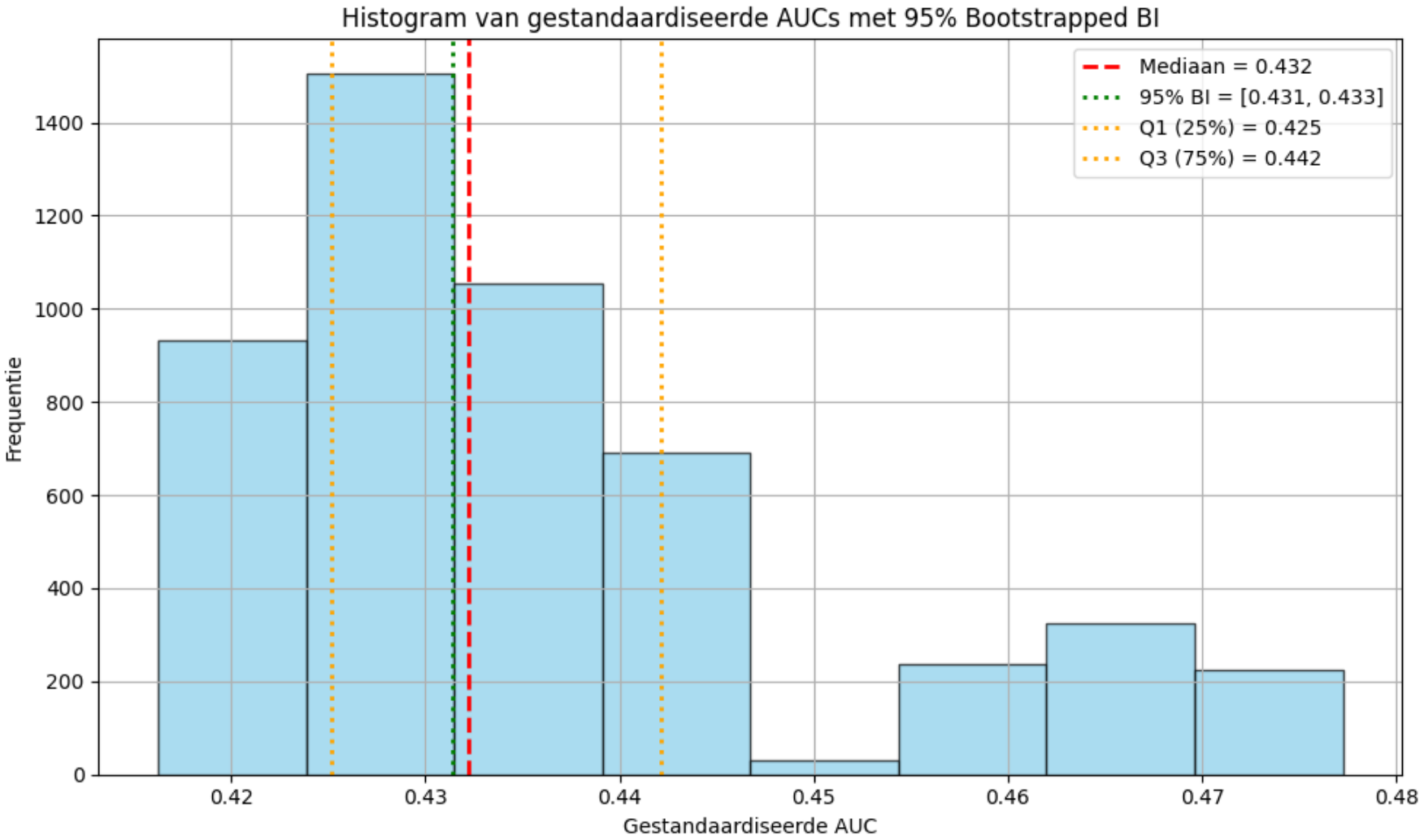
FIGUUR 71: Grafiek van 36% drempelwaarde



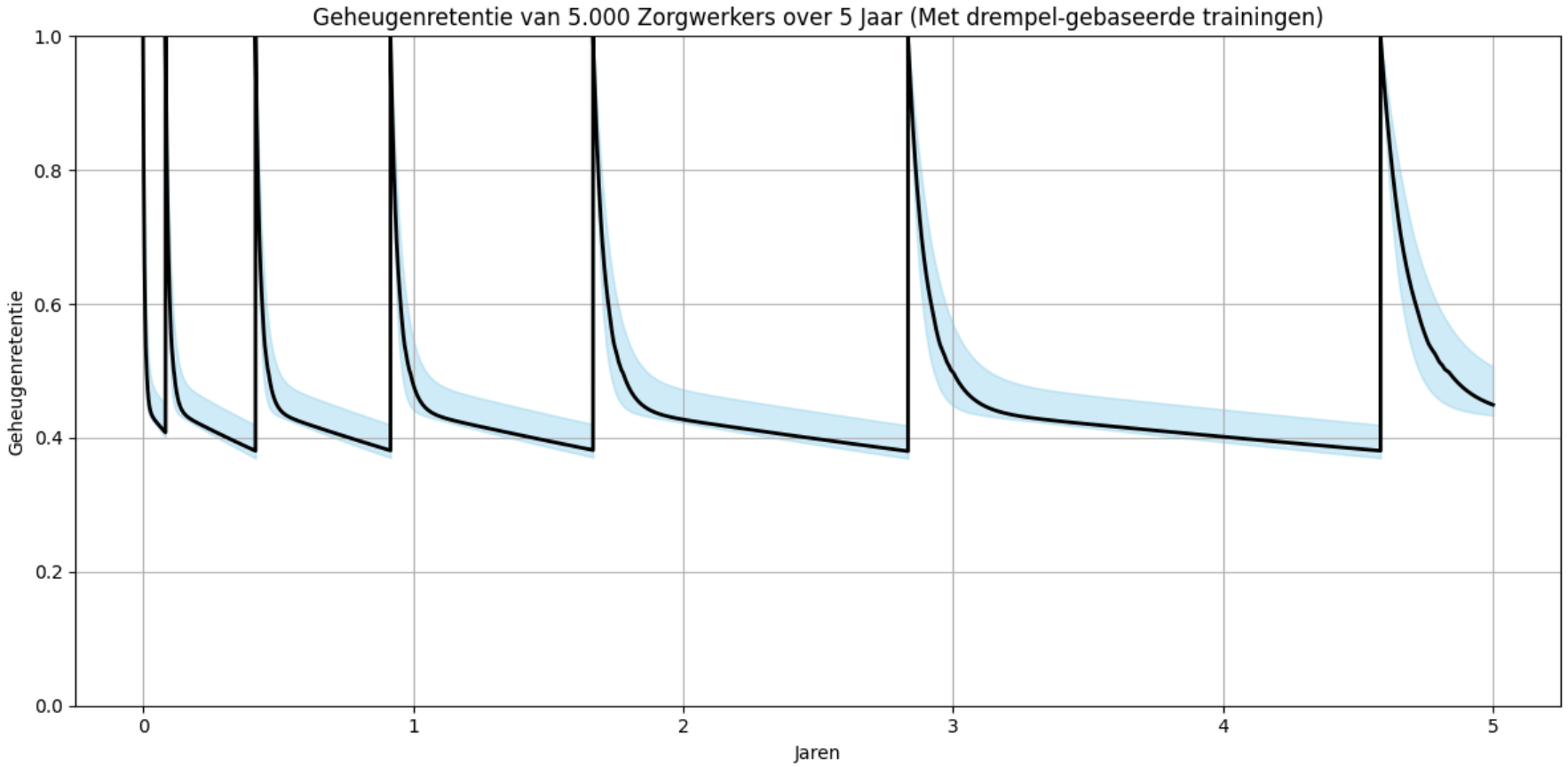
FIGUUR 72: Histogram van 36% drempelwaarde



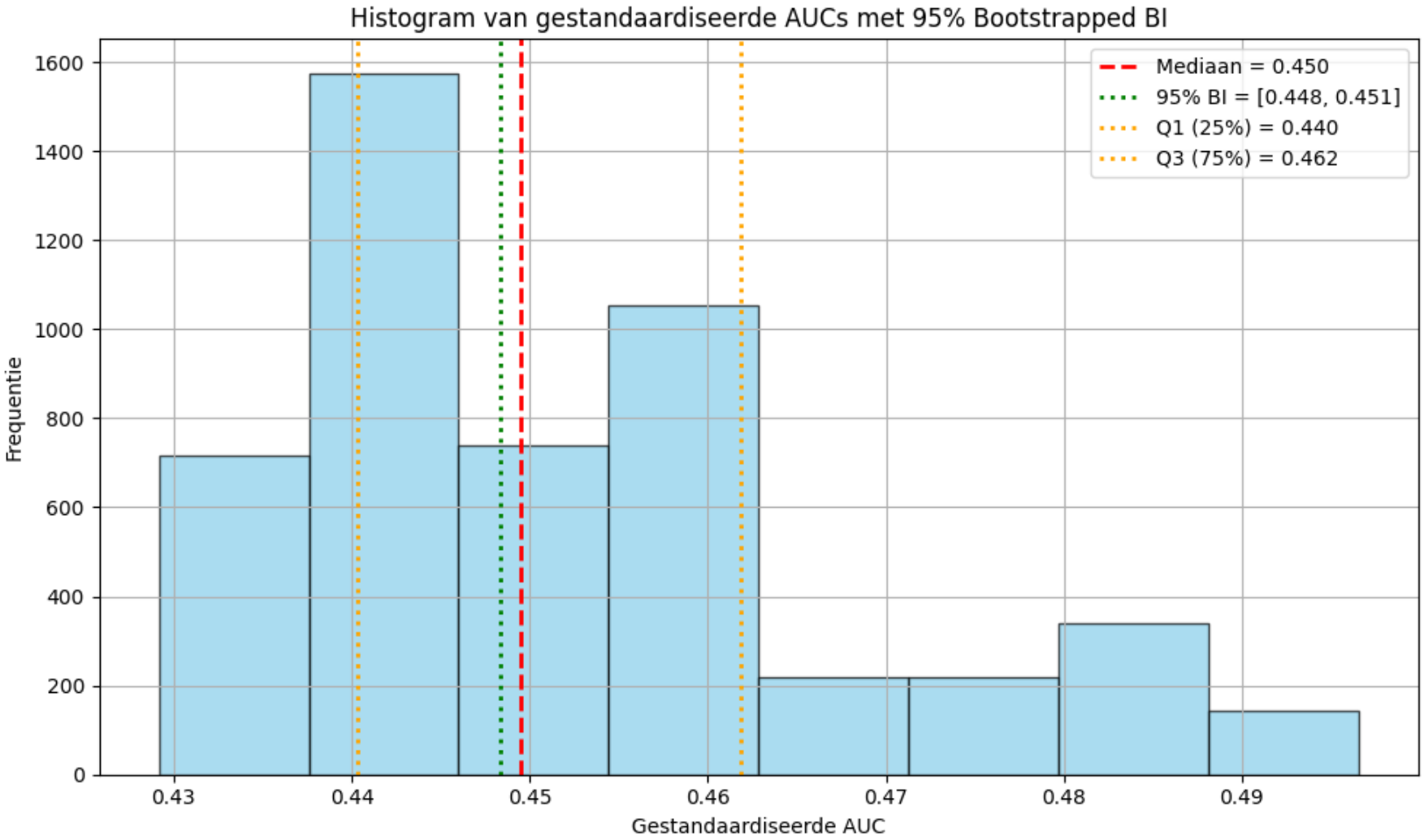
FIGUUR 73: Grafiek van 37% drempelwaarde



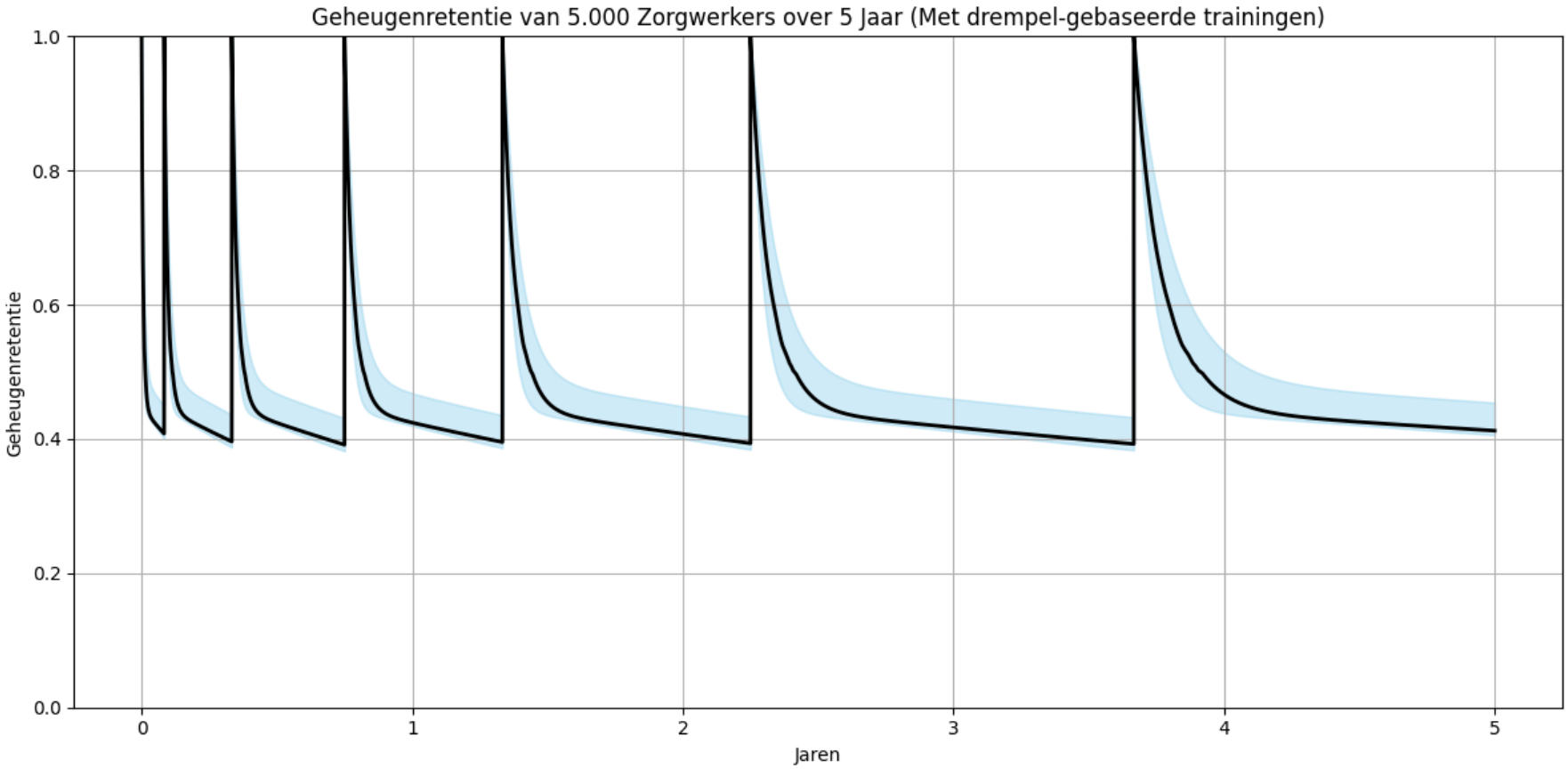
FIGUUR 74: Histogram van 37% drempelwaarde



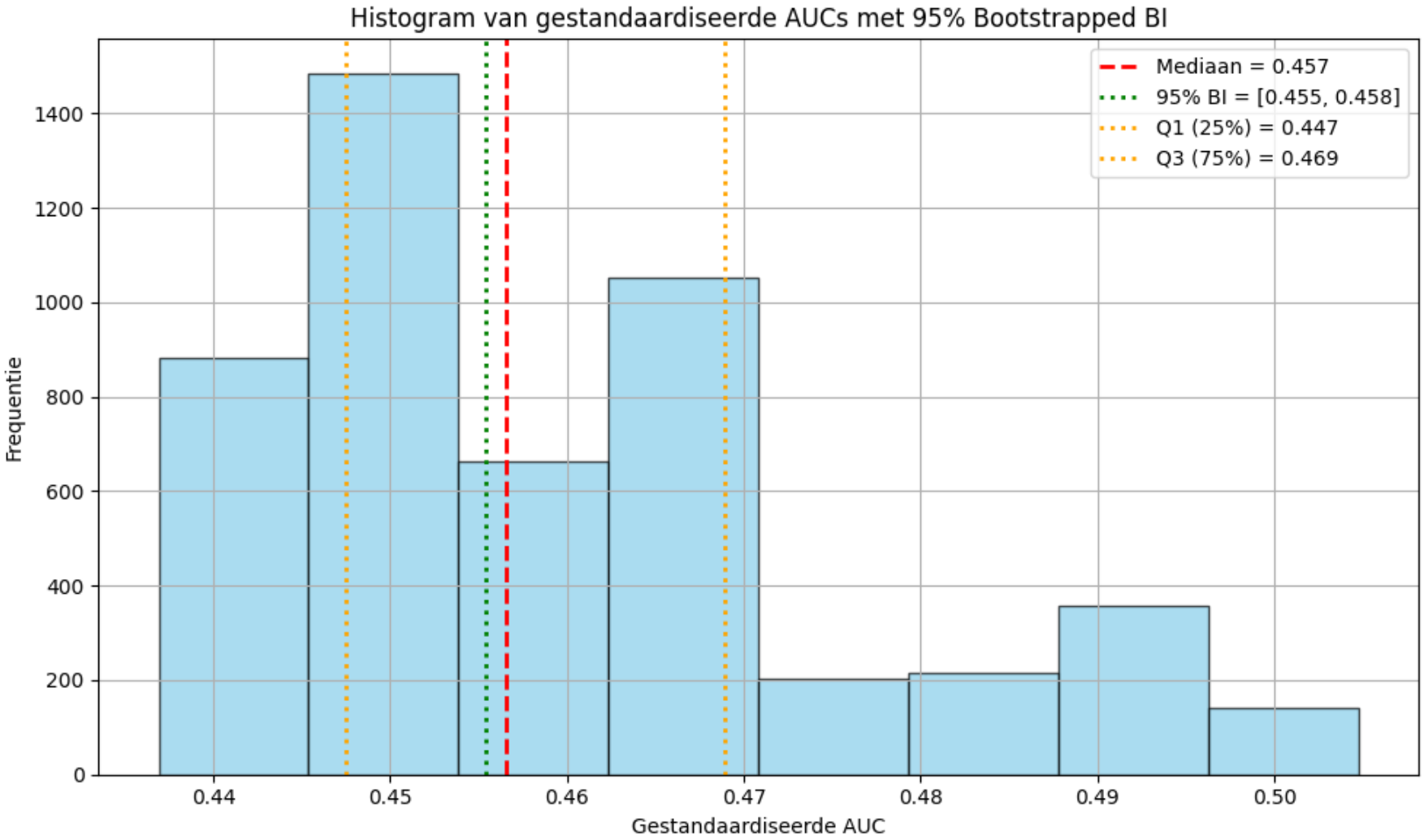
FIGUUR 75: Grafiek van 38% drempelwaarde



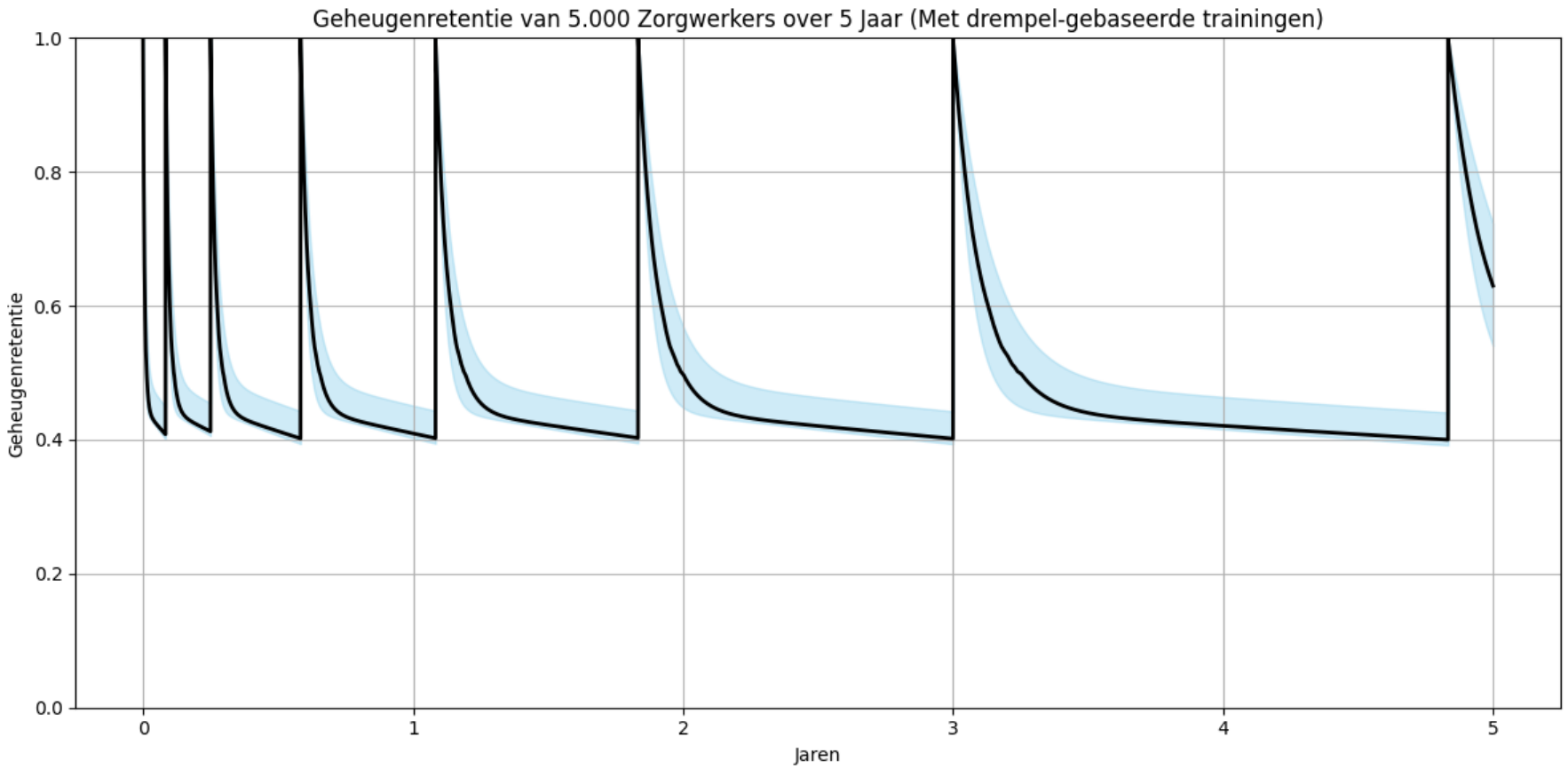
FIGUUR 76: Histogram van 38% drempelwaarde



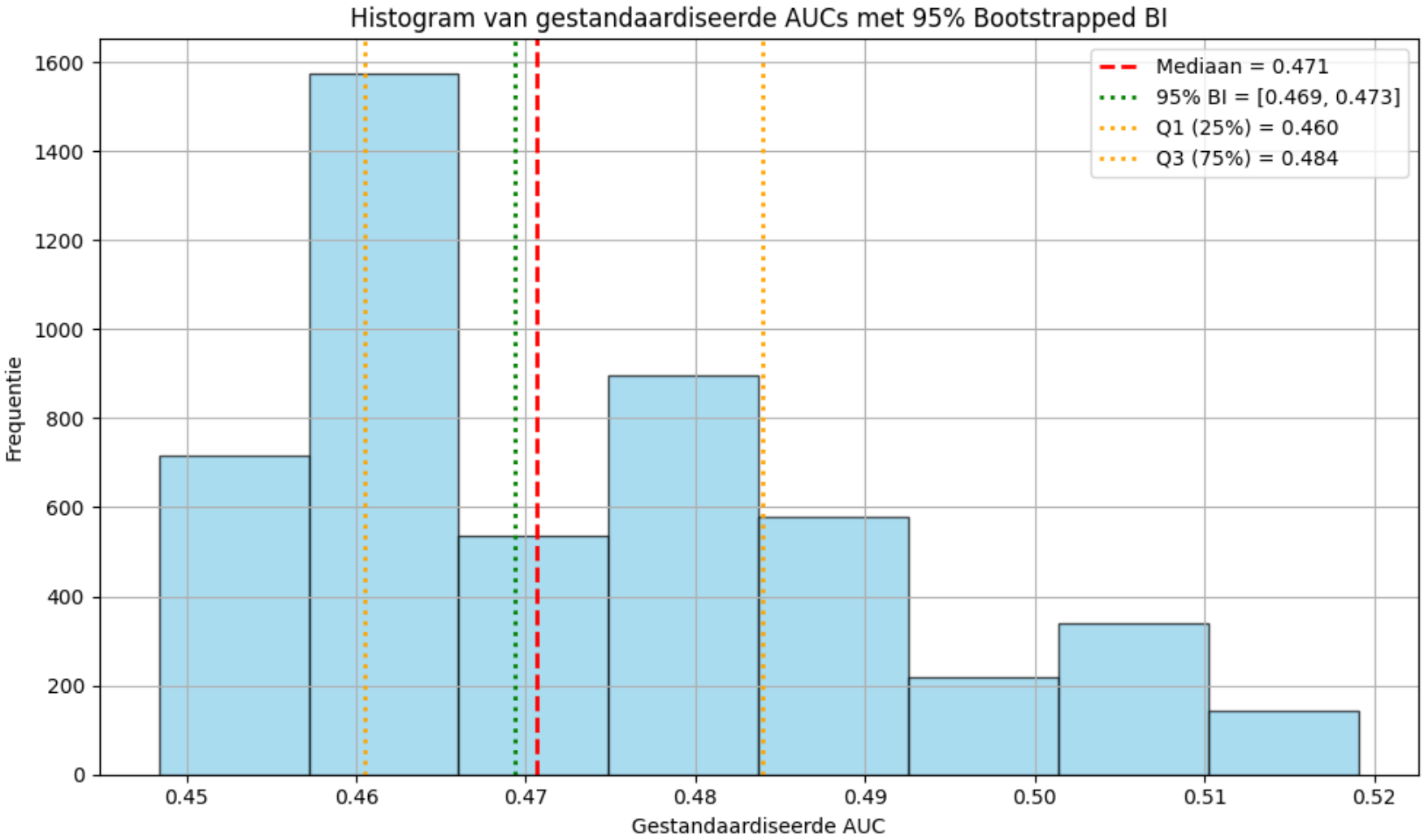
FIGUUR 77: Grafiek van 39% drempelwaarde



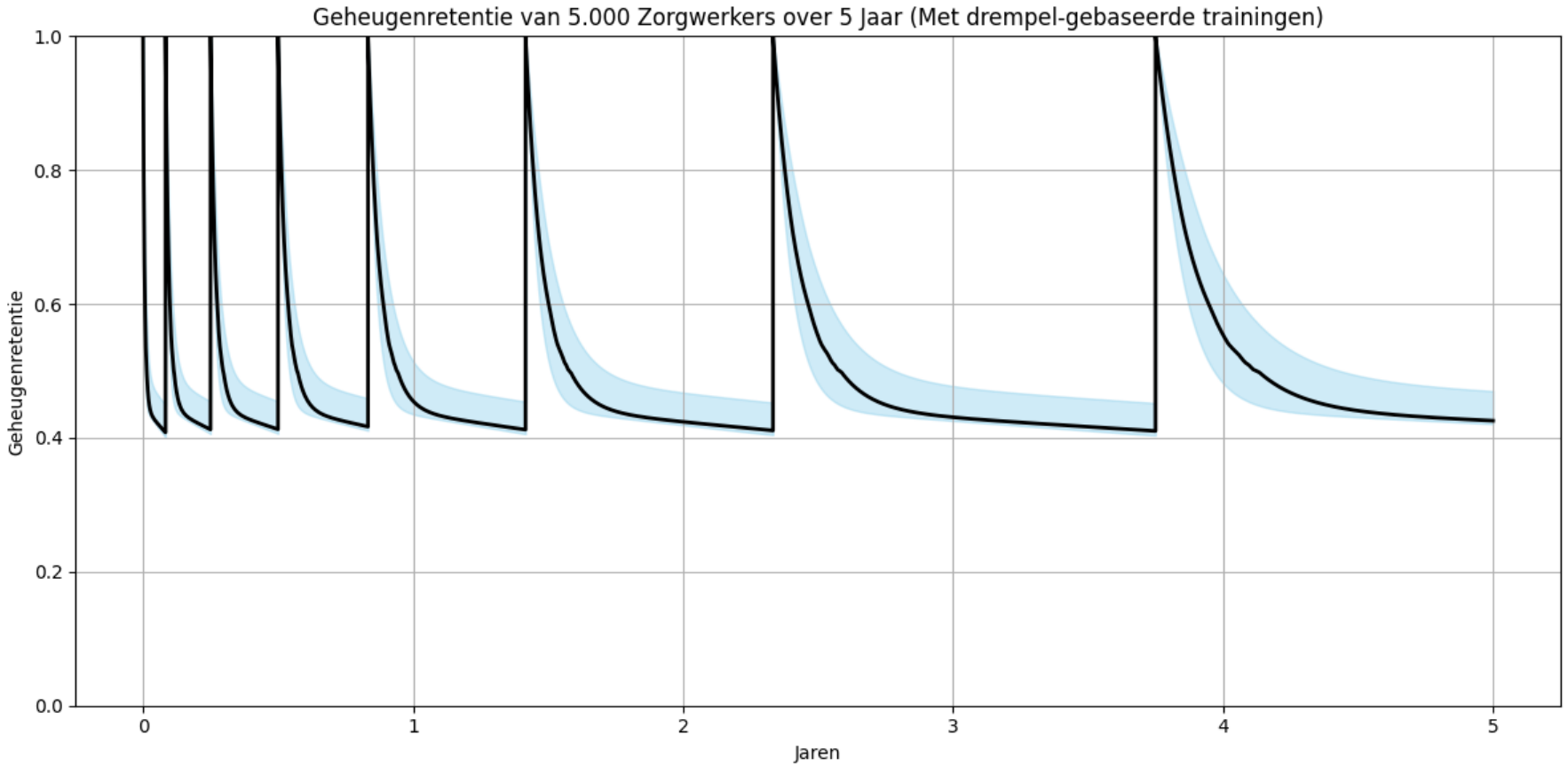
FIGUUR 78: Histogram van 39% drempelwaarde



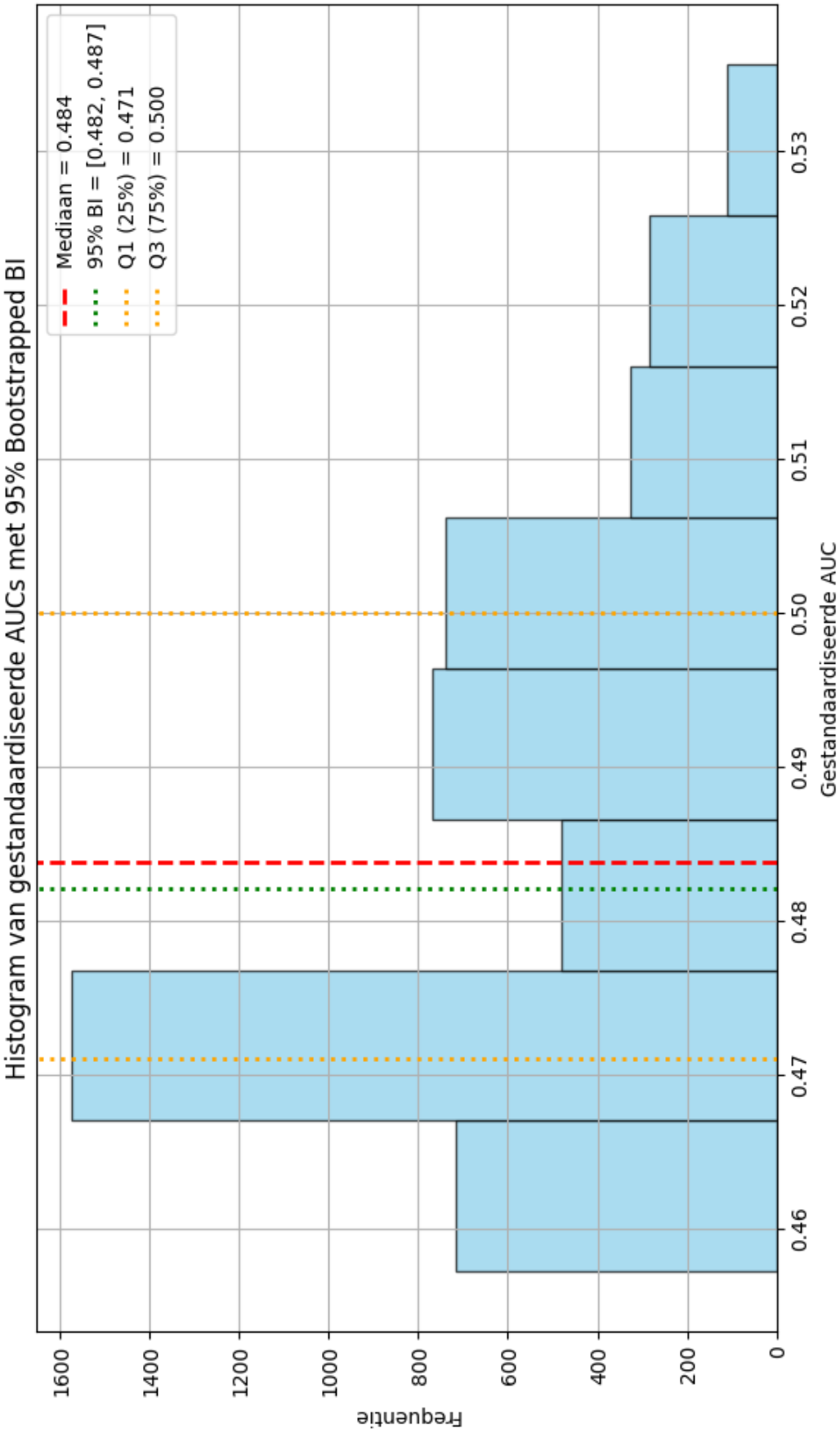
FIGUUR 79: Grafiek van 40% drempelwaarde



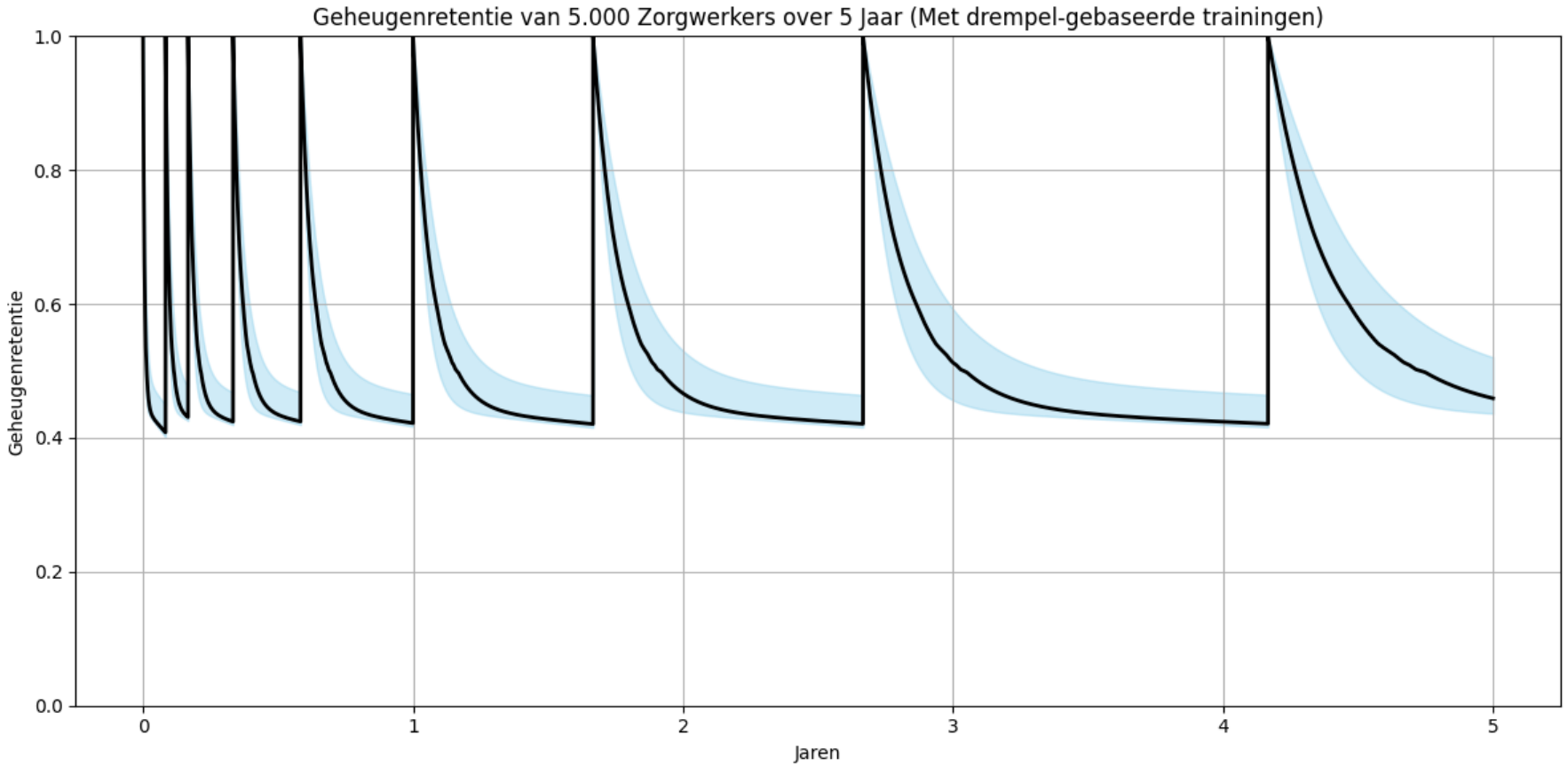
FIGUUR 80: Histogram van 40% drempelwaarde



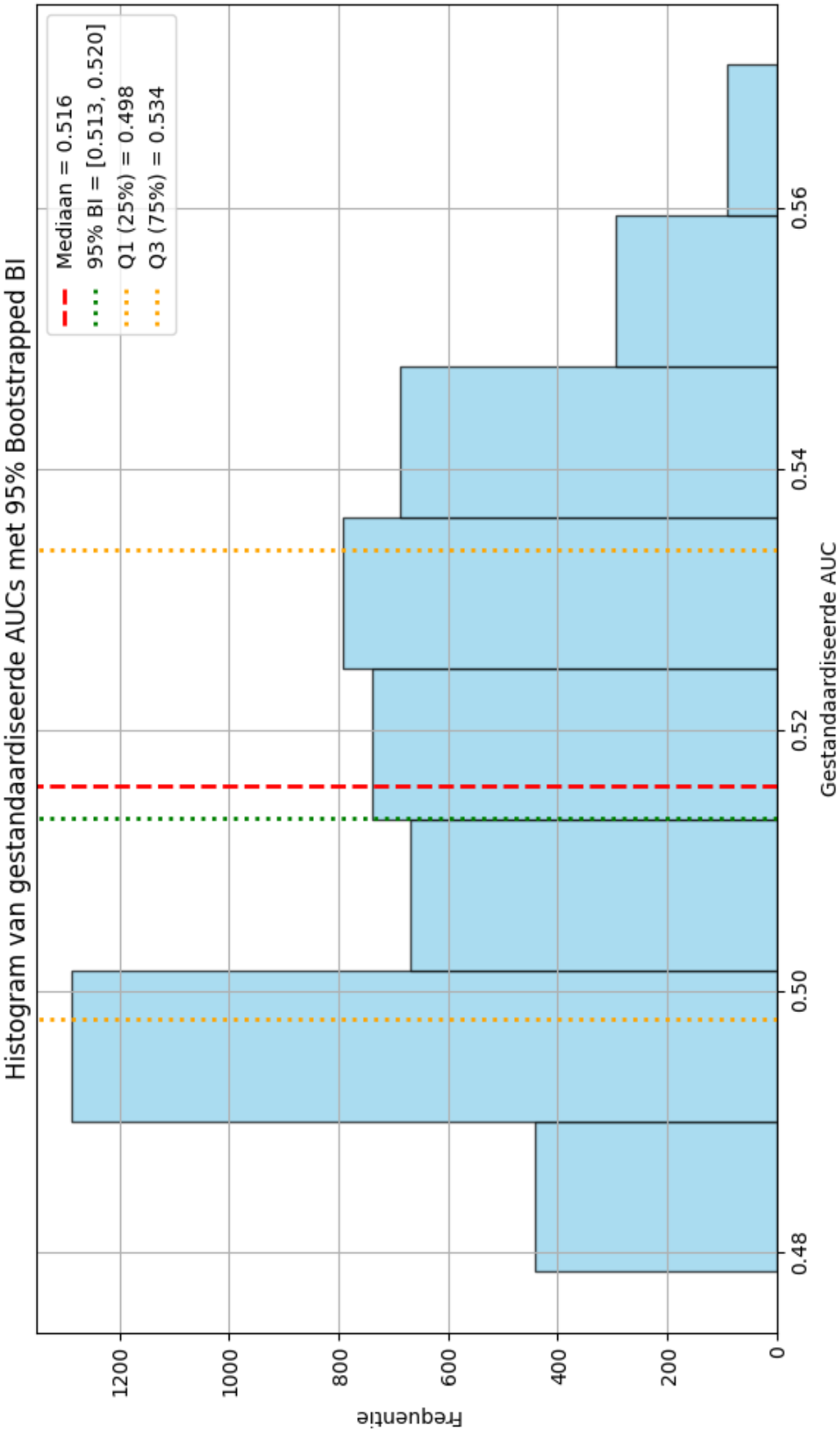
FIGUUR 81: Grafiek van 41% drempelwaarde



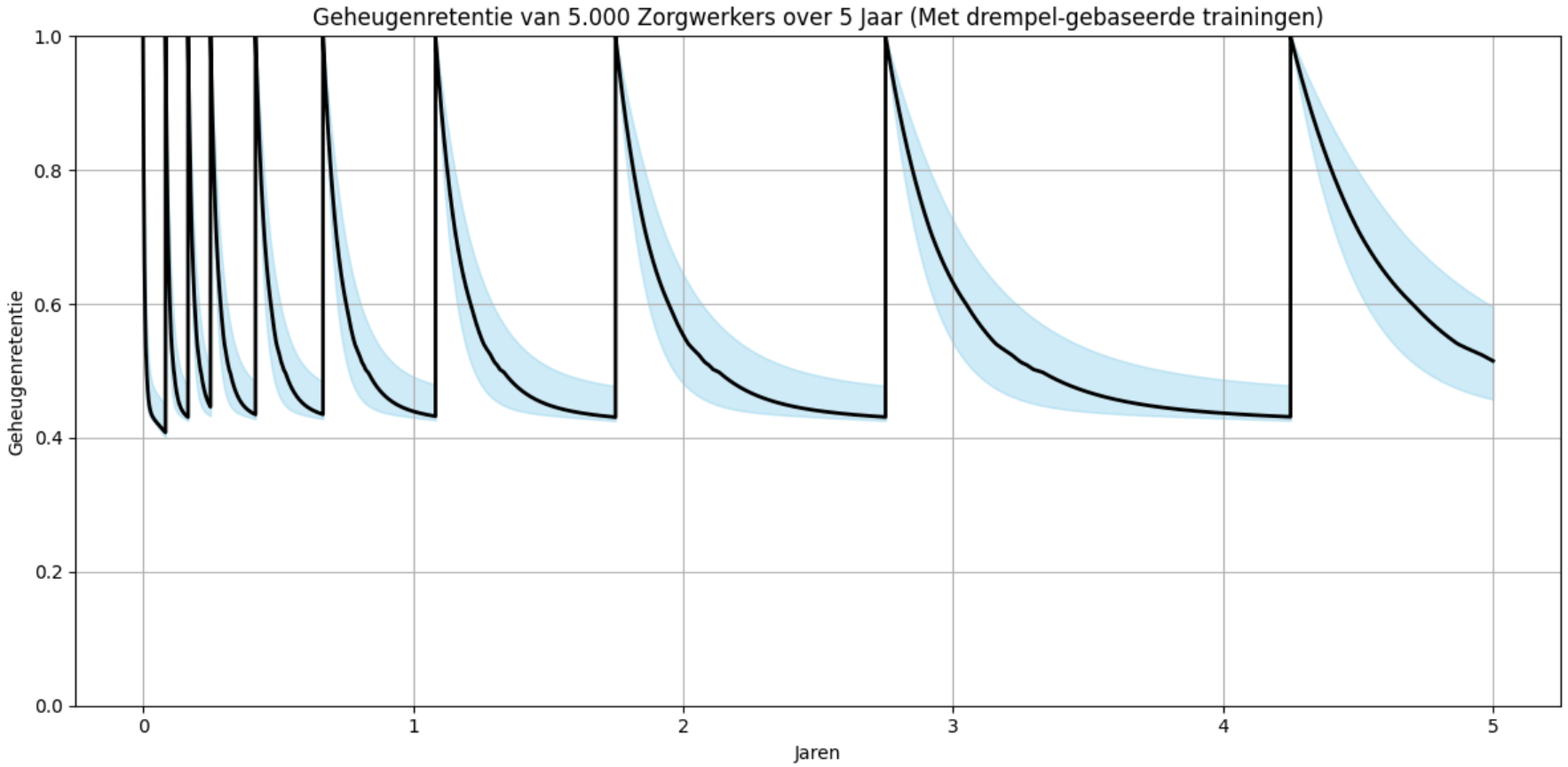
FIGUUR 82: Histogram van 41% drempelwaarde



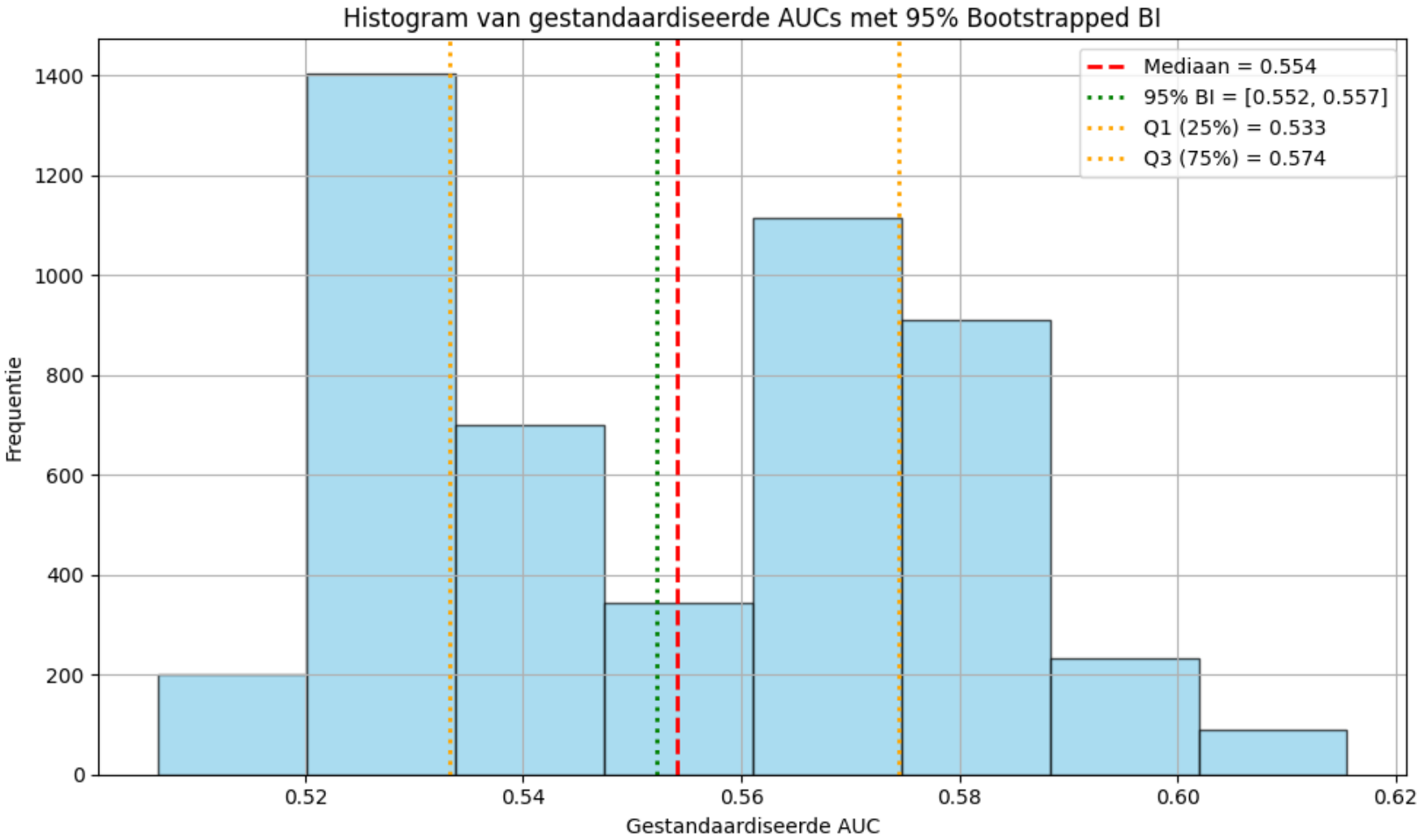
FIGUUR 83: Grafiek van 42% drempelwaarde



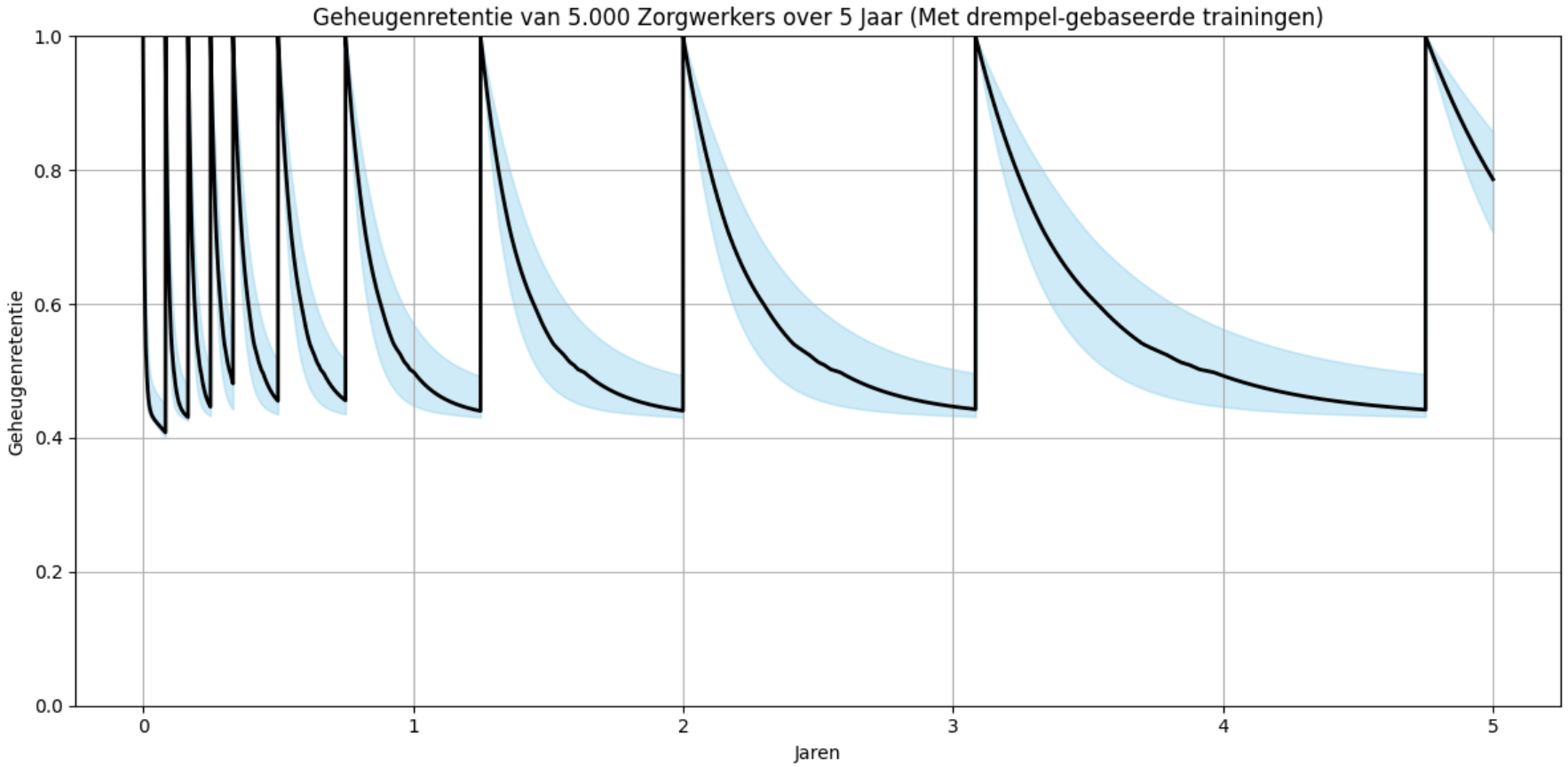
FIGUUR 84: Histogram van 42% drempelwaarde



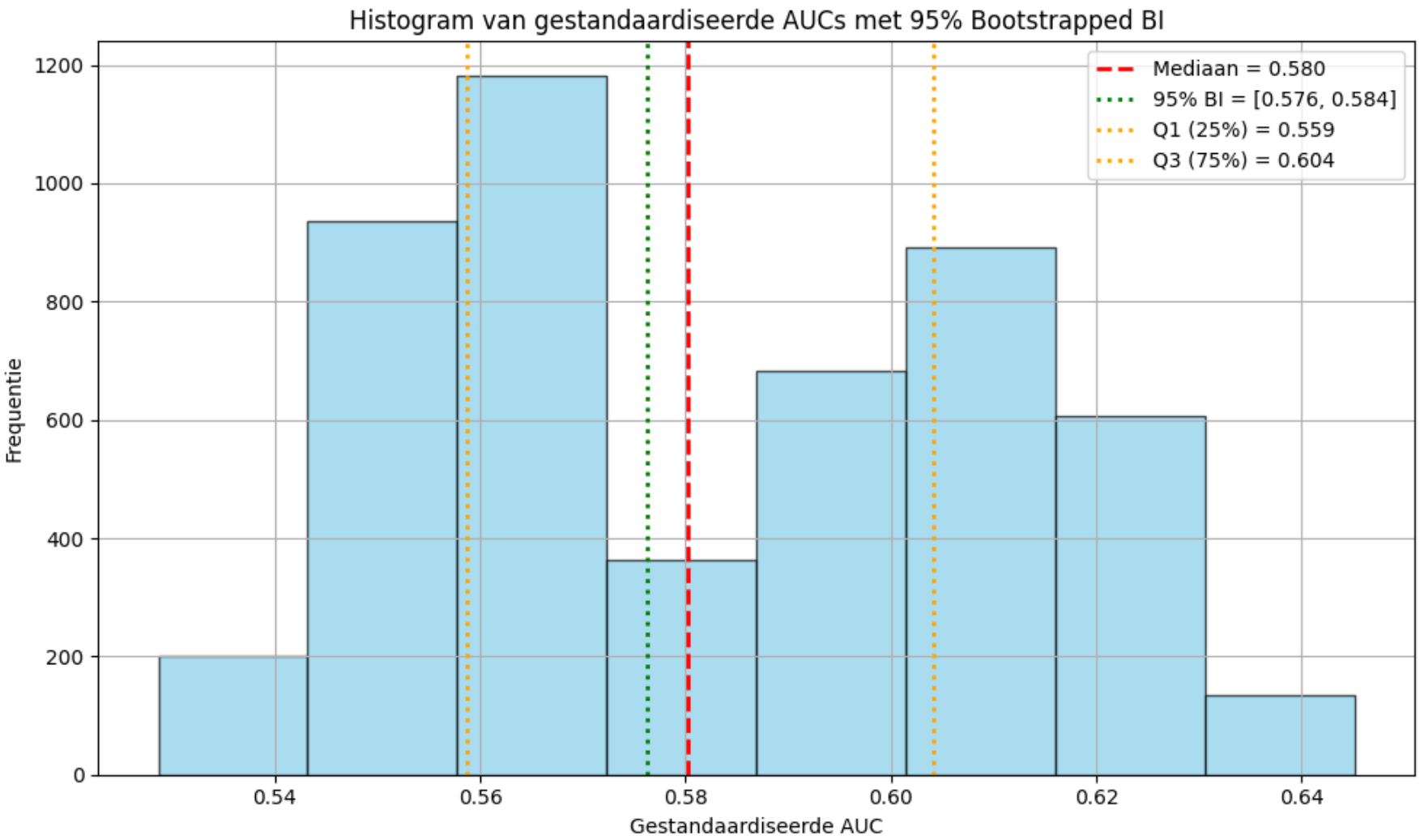
FIGUUR 85: Grafiek van 43% drempelwaarde



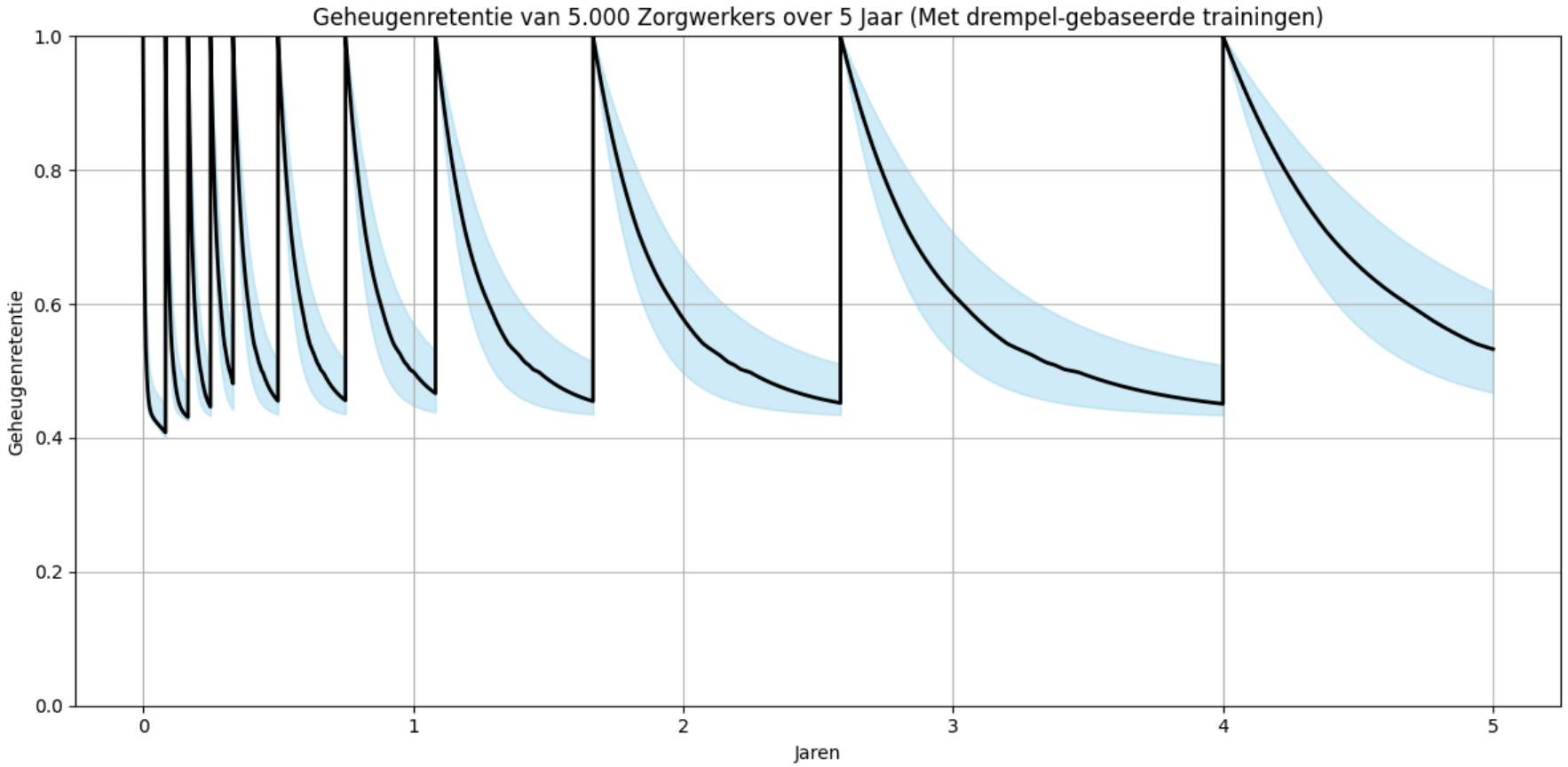
FIGUUR 86: Histogram van 43% drempelwaarde



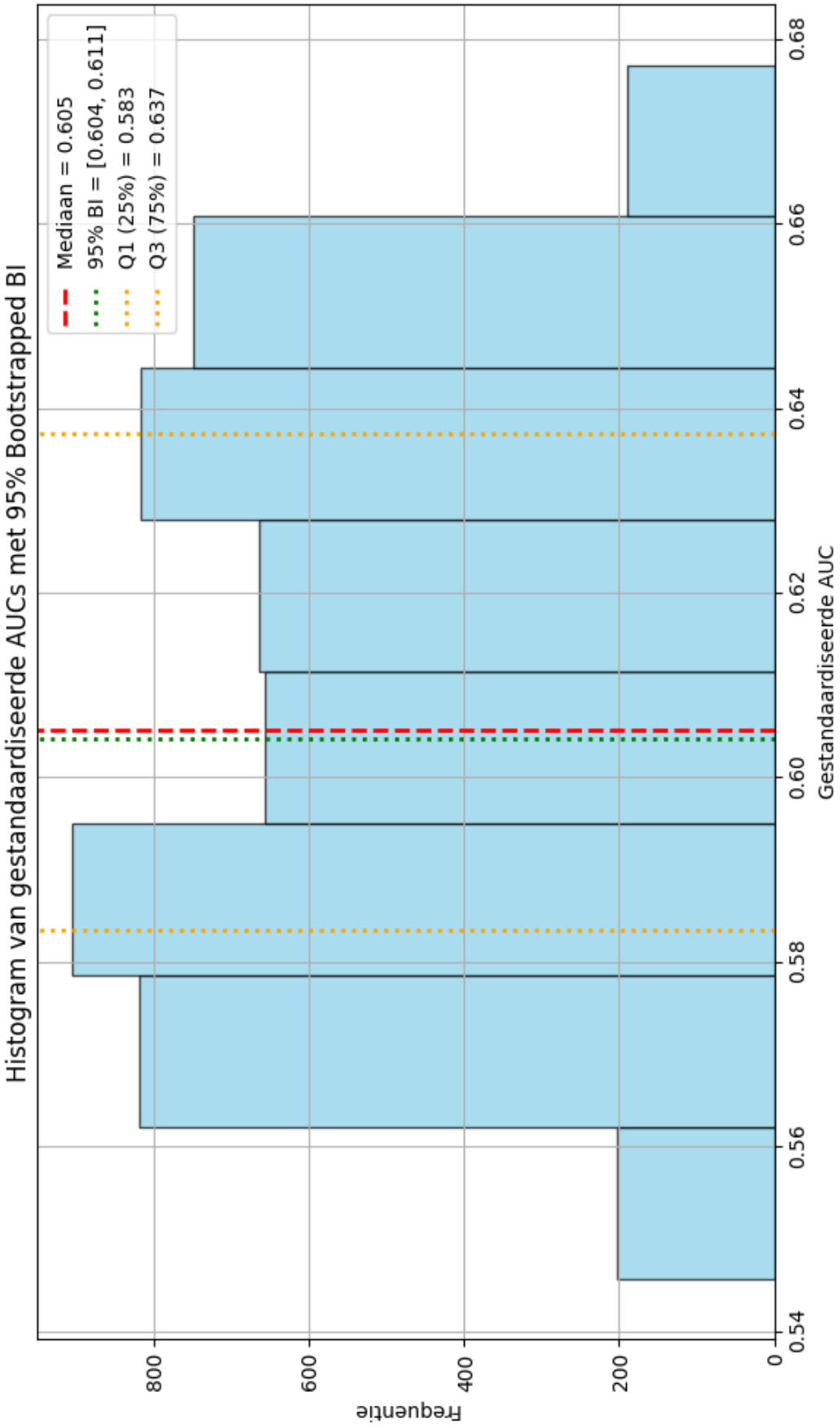
FIGUUR 87: Grafiek van 44% drempelwaarde



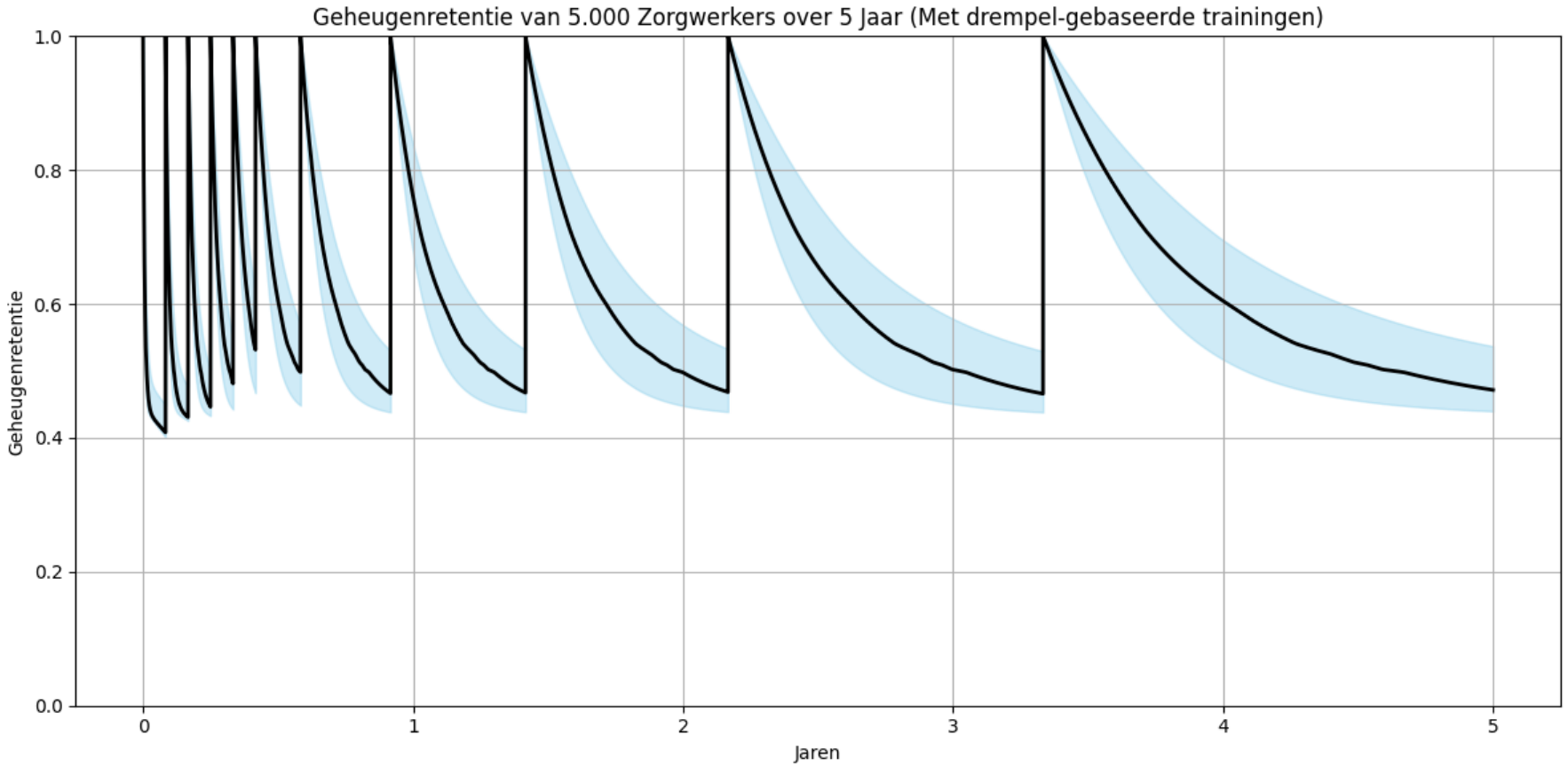
FIGUUR 88: Histogram van 44% drempelwaarde



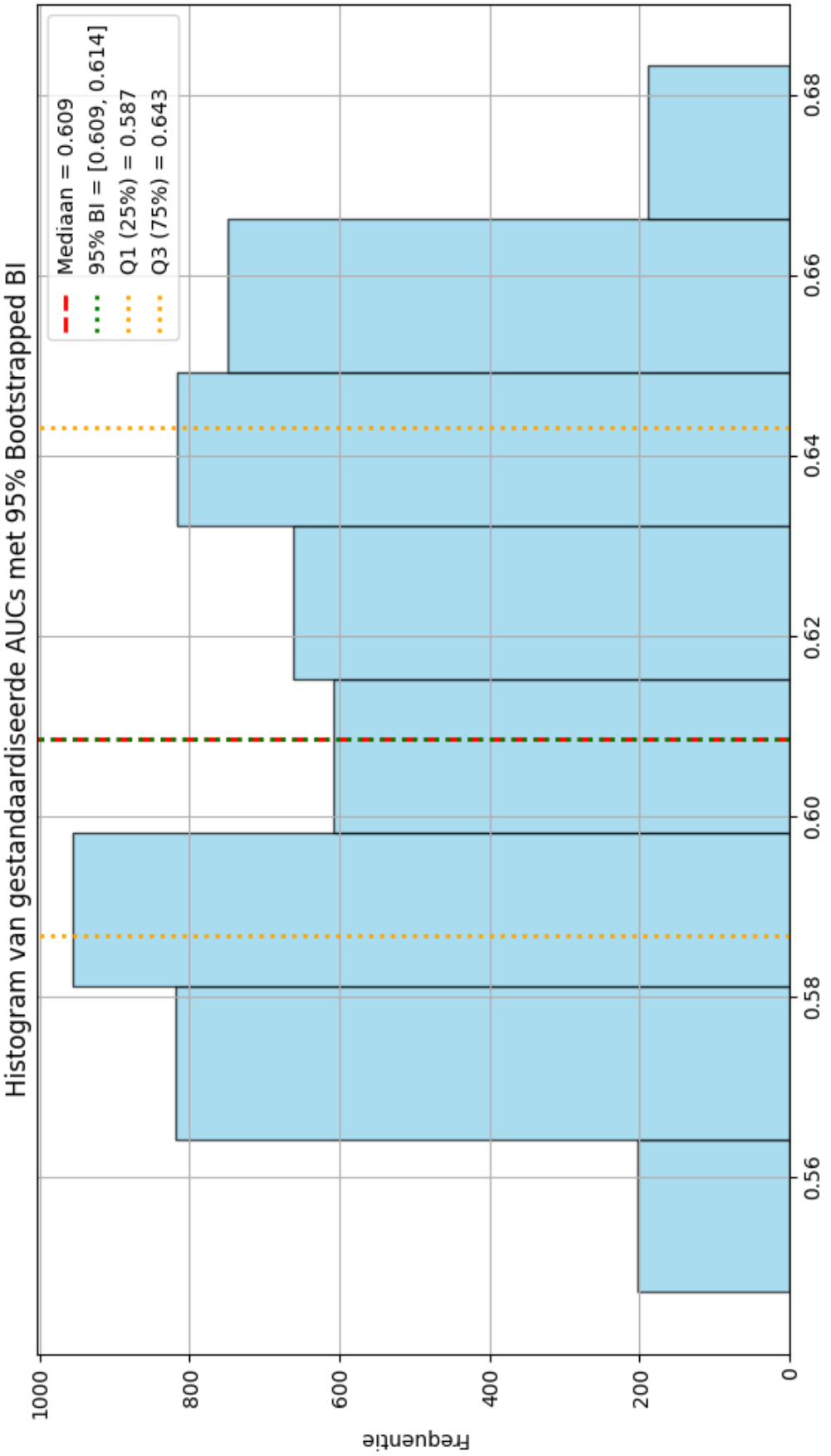
FIGUUR 89: Grafiek van 45% drempelwaarde



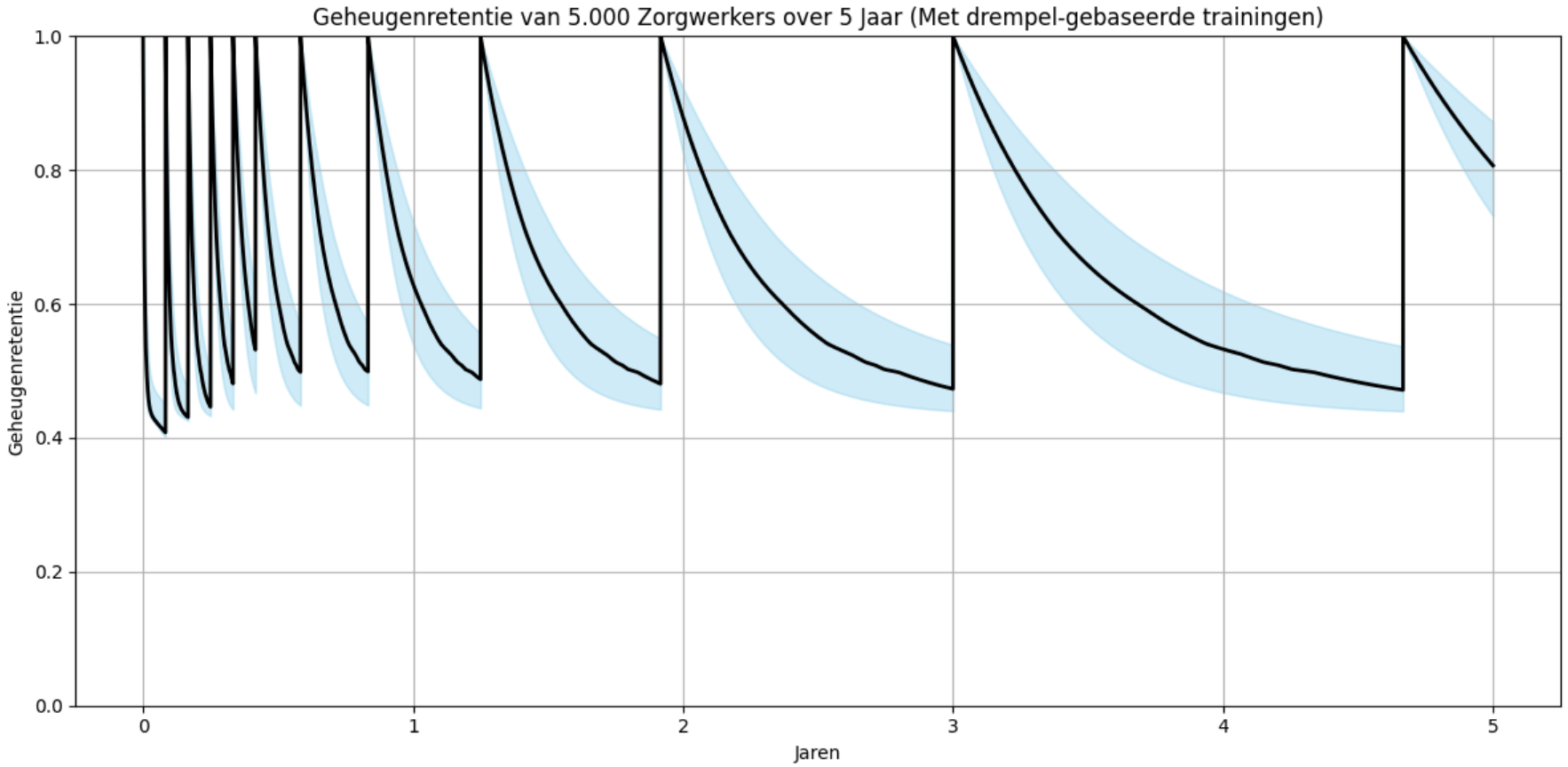
FIGUUR 90: Histogram van 45% drempelwaarde



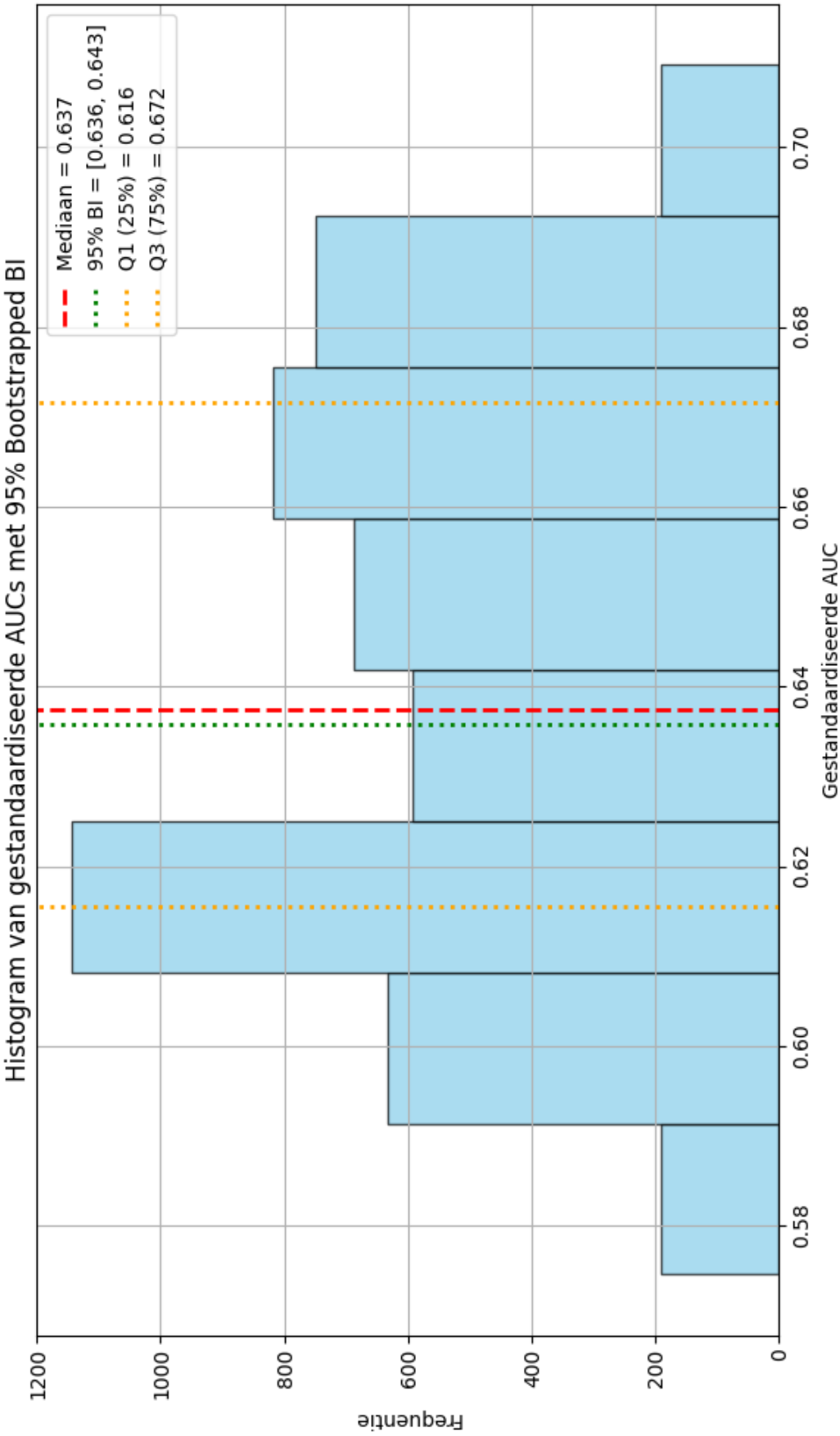
FIGUUR 91: Grafiek van 46% drempelwaarde



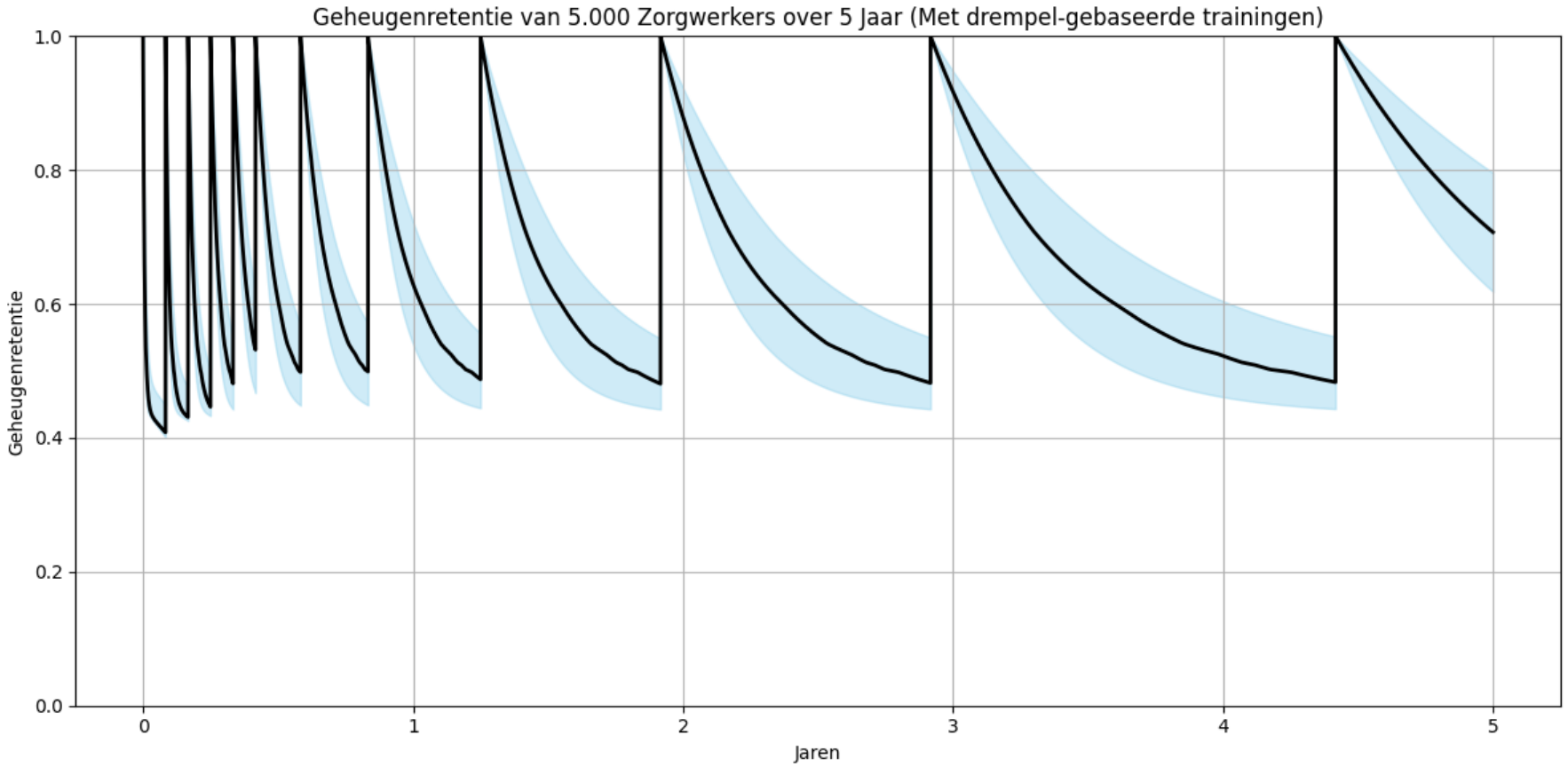
FIGUUR 92: Histogram van 46% drempelwaarde



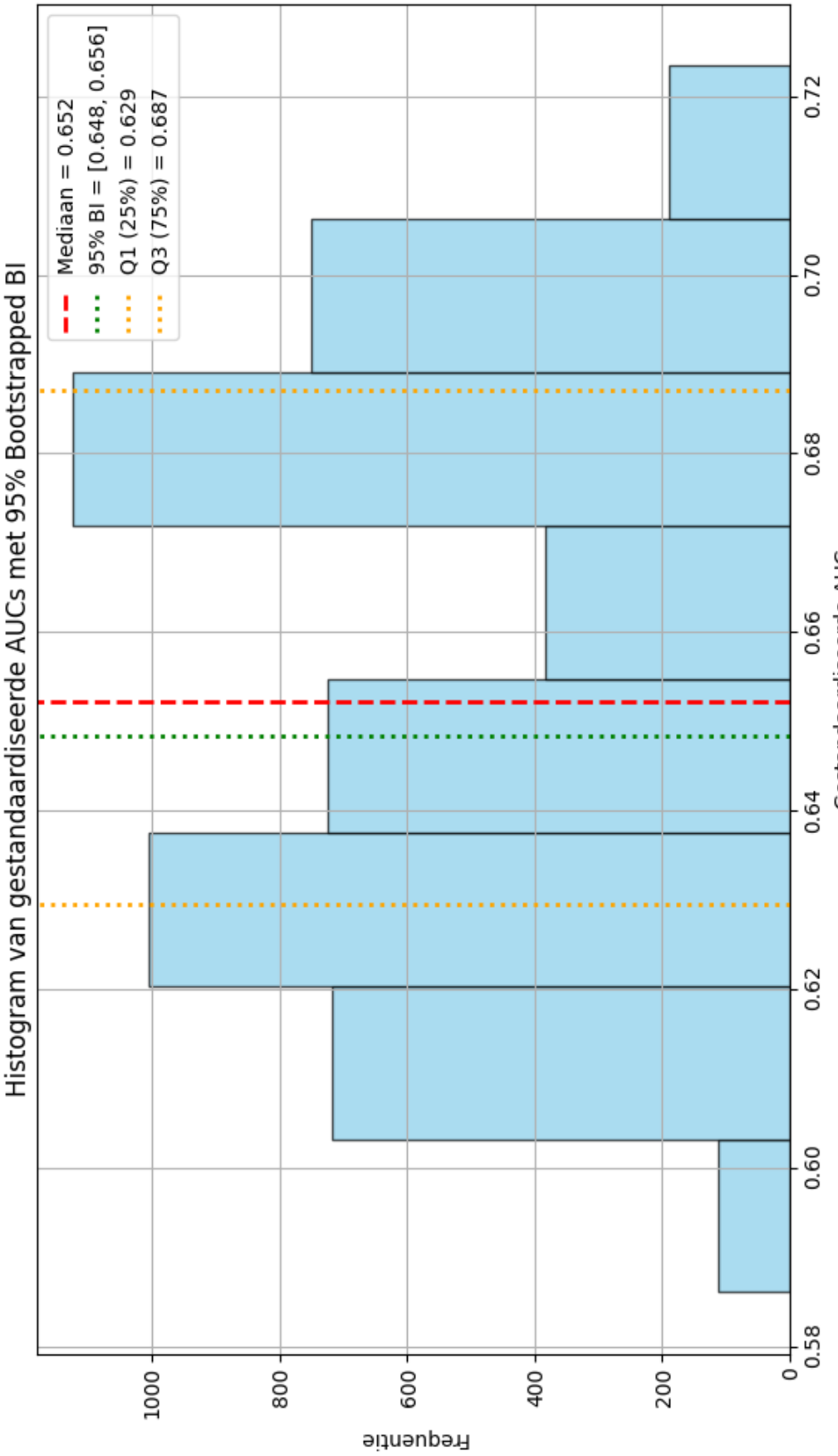
FIGUUR 93: Grafiek van 47% drempelwaarde



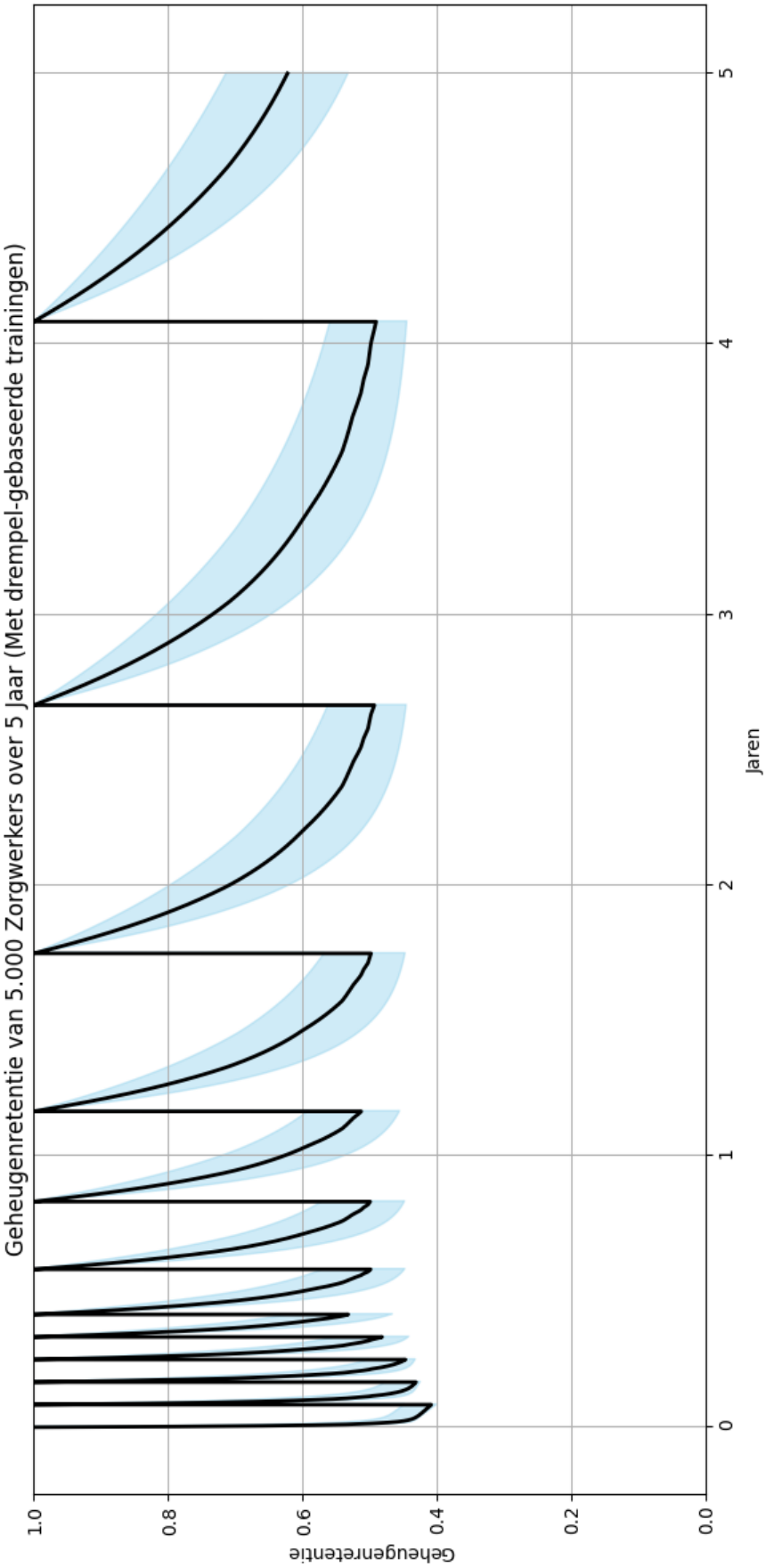
FIGUUR 94: Histogram van 47% drempelwaarde



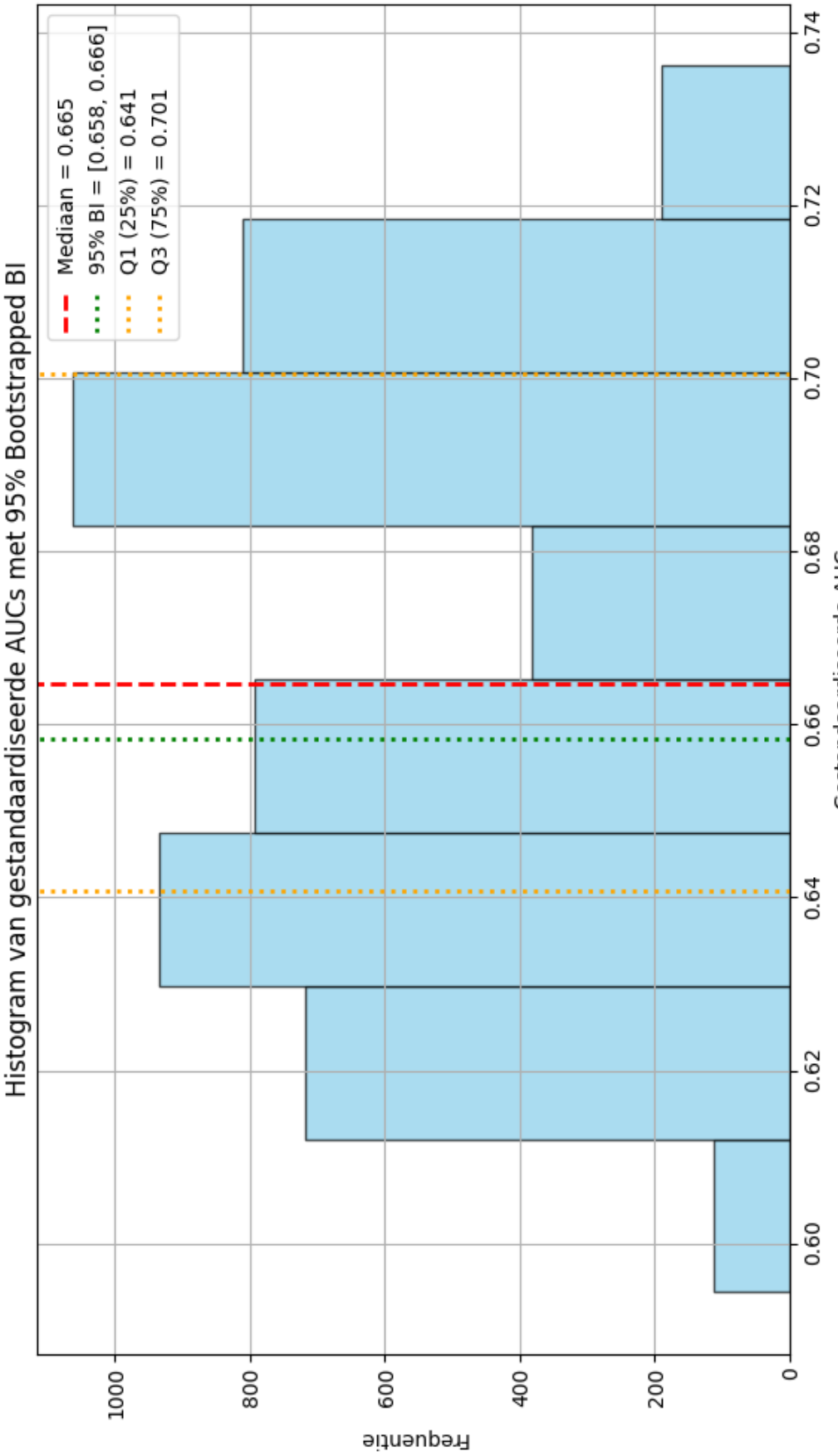
FIGUUR 95: Grafiek van 48% drempelwaarde



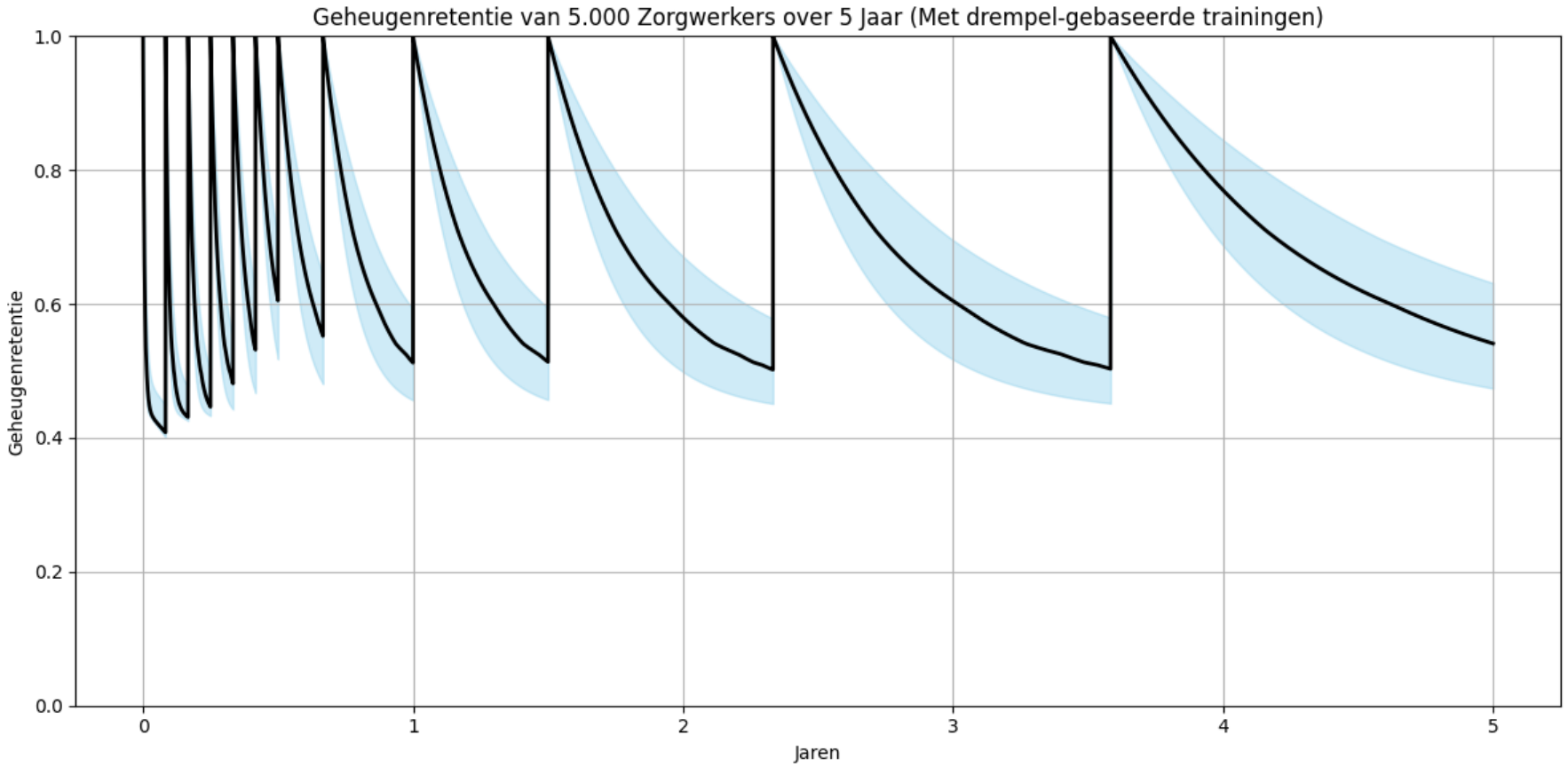
FIGUUR 96: Histogram van 48% drempelwaarde



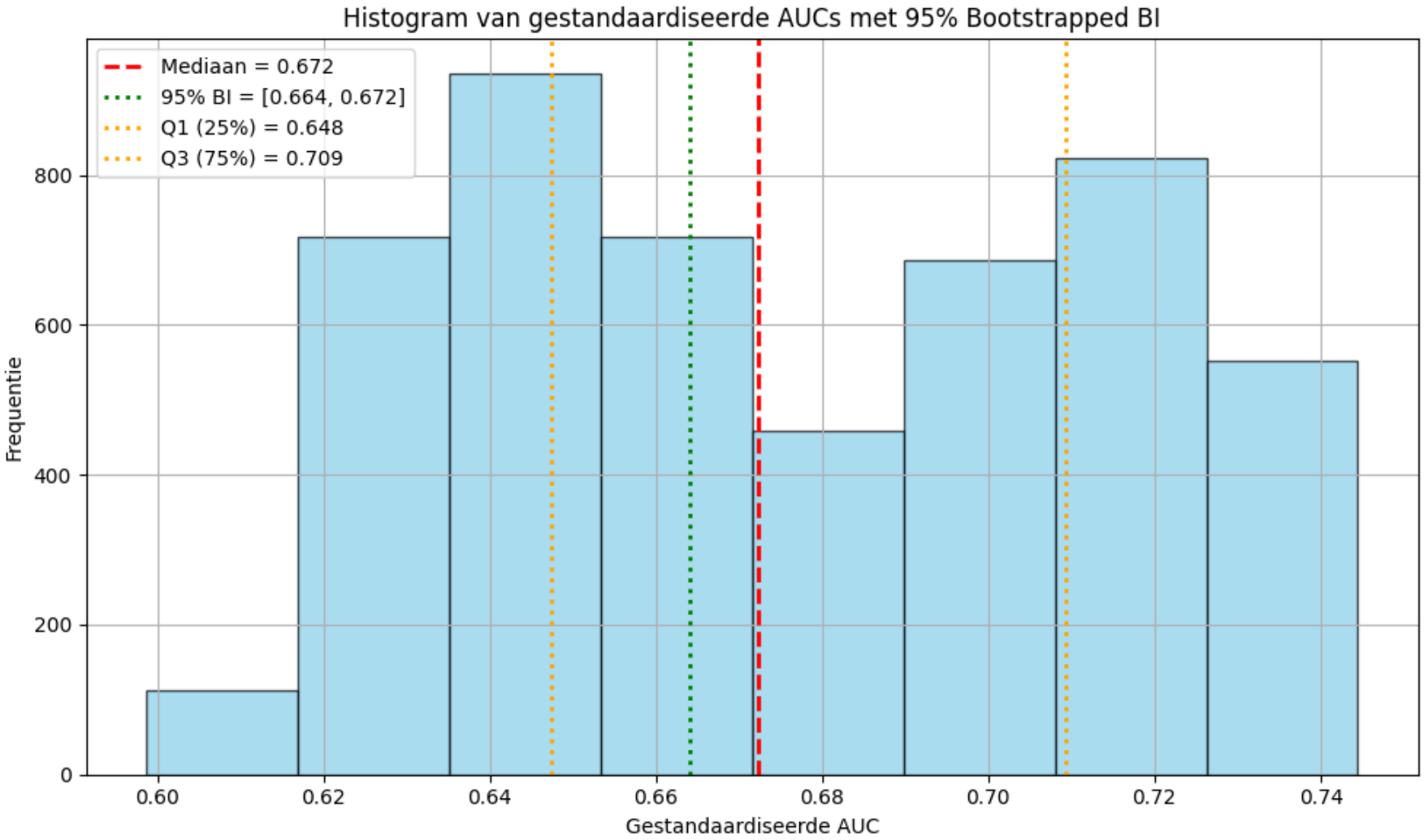
FIGUUR 97: Grafiek van 49% drempelwaarde



FIGUUR 98: Histogram van 49% drempelwaarde



FIGUUR 99: Grafiek van 50% drempelwaarde



FIGUUR 100: Histogram van 50% drempelwaarde

"Tijdens de voorbereiding van dit onderzoek heeft de auteur ChatGPT en Copilot gebruikt voor het opzoeken van bronnen en het coderen. Na het gebruik van deze hulpmiddelen heeft de auteur de inhoud beoordeeld en, indien noodzakelijk, aangepast. De auteur neemt volledige verantwoordelijkheid voor de inhoud van dit onderzoek."