

# Van Ruwe Data tot Reddingslijn: Hoe AI Hartziekten Voorspelt

## Diepgaand onderzoek naar neurale netwerken verbetert vroege detectie van cardiale afwijkingen

Frans Harren

29-06-2025

Studentnummer: 189165115

<https://github.com/hufrans/mads-exam-fharren.git>

Dit verslag beschrijft een onderzoek naar het ontwikkelen van een deep learning-model voor het classificeren van hartritmestoornissen op basis van ECG-signalen. Het doel is om automatisch te bepalen of een persoon een normaal of afwijkend hartritme heeft. Bij een afwijking moet het model kunnen aangeven tot welke van de vijf klassen de stoornis behoort, zodat een gerichte behandeling mogelijk is.

Om te beoordelen of een deep learning-model goed presteert, wordt recall als belangrijkste maatstaf gebruikt. Recall geeft aan hoeveel zieke gevallen het model correct herkent; hoe goed het model is in het niet missen van mensen met een hartritmestoornis. In een medische context is dat cruciaal, omdat een gemiste diagnose ernstige gevolgen kan hebben.

Het onderzoek gebruikt de Physionet's MIT-BIH Arrhythmia Dataset met 109446 observaties, onderverdeeld in vijf klassen:

*Tabel 1: Klasseverdeling van de dataset*

Klasse	Ziekte	Observaties	Percentage
0	Normale hartslag – alles ziet er goed uit	72471	82.8%
1	Vroege slag uit het bovenste deel van het hart	2223	2.5%
2	Vroege slag uit het onderste deel van het hart	5788	6.6%
3	Een mengvorm van een normale en een vroege slag	641	0.7%
4	Onbekend of fout in de meting	6431	7.3%

## Datapreparatie

De dataset heeft een duidelijke disbalans in de verdeling van de target klassen. Klasse 0 heeft duidelijk de meerderheid. Hierdoor zal een model getraind op een onbewerkte dataset een grote voorkeur hebben om klasse 0 te kiezen als uitkomst van de classificatie op een testdataset. Dit is ongewenst, want de minderheidsklassen representeren ziektebeelden die gedetecteerd moeten worden. In de dataset moet klasse 0 van de overige klassen gescheiden worden, zodat het model onderscheid kan maken tussen de normale en de zieke hartslagen.

Een manier is om met een autoencoder met unsupervised learning een normale hartslag te vinden en met verschilanalyse de abnormale hartslagen te vinden, anomaliedetectie. Hierbij zouden alle features minus de target gebruikt worden. Alhoewel dit een goede manier is, is er in dit geval niet voor gekozen, want de dataset bevat al een classificatie. Uit de documentatie is duidelijk dat klasse 0 de normale en de overige klassen de ziekteklassen zijn. Een autoencoder gebruiken is in dit geval niet nodig.

Een andere strategie is om voor de niet dominante klassen synthetische data te genereren. SMOTE is hiervoor een algoritme dat vaak gebruikt wordt. Het maakt gebruik van interpolatie tussen bestaande datapunten. Dit is voor hartslagdata ongewenst want het houdt geen rekening met tijdsafhankelijke aspecten van een hartslag. Een hartslag bestaat uit onderdelen die tijdsafhankelijk zijn, zoals de samentrekking van de kamerspier en de boezemspier. SMOTE maakt een lineaire interpolatie en valt daardoor dus af.

Een ander alternatief is Generative Adversarial Network (GAN). Dat model houdt rekening met de tijdsafhankelijk onderdelen van de hartslag. Hierdoor kan realistischere synthetische data gegenereerd worden. Een GAN bestaat uit twee neurale netwerken: Generator en Discriminator. De generator maakt nepdata die lijkt op de echte data. De discriminator probeert te bepalen of de data nep of echt is. De generator leert van de uitkomst van de discriminator, waardoor er een model ontstaat dat steeds beter het origineel benadert. De subtiele patronen in de hartslag worden zo beter in de model gevangen.

Er is gekozen om met GAN synthetische data te maken. Iedere niet dominante klasse heeft een eigen GAN gekregen om zo het aantal observaties te vermeerderen en de verhouding tussen de normale hartslag en de ziekten beter te balanceren. Hierdoor heeft het deep learning-model meer data om op te trainen en de ziektebeelden beter te herkennen.

Een GAN heeft tijd nodig om te leren. Daarom zijn er duizenden epochs gebruikt om het model te trainen. Hierbij speelt de loss functie voor de generator en de discriminator een belangrijke rol. De optimale waarde van de loss (wanneer de generator perfecte nepdata maakt en de discriminator

niet beter dan gokken) is:  $\text{Loss}_{\min} = -\log(0.5) = \ln(2)$ . Als de discriminator niet kan onderscheiden tussen echt en nep, dan is de kans op correct raden 0.5. Het log-verlies bij die kans is  $\ln(2) \approx 0.693$ . Het doel is om de loss zo dicht mogelijk naar de 0.693 te krijgen. Dat is het criterium voor de training van de GAN. Als de discriminator-loss rond  $\ln(2)$  ligt, betekent dit dat hij niet beter dan gokken presteert. Als de generator erin slaagt om de discriminator op dat punt te brengen, is dat een teken van succes. Dan weet de discriminator niet of het echt of synthetische data is.

Voor elk van de vier ziekteklassen is een GAN getraind, waarmee synthetische data is aangemaakt. De synthetische data is gecombineerd met de originele data om iedere van de vier klasse een gelijk aantal observaties te geven, zodat ze een gelijke kans hebben om door het model herkend te worden. Het aantal observaties van deze klassen wordt de helft van het aantal van klasse 0, zodat klasse 0 nog steeds de dominante klasse blijft. Bij het trainen wordt rekening door een een weighted cross entropy loss functie te gebruiken, waarbij de niet dominante klassen zwaarder meewegen.

## Hypothesen

Dit onderzoek heeft zes hypothesen onderzocht om te bepalen welk model en welke datastrategie de beste recall oplevert voor ECG-classificatie. De hypothesen richten zich op modelarchitectuur en de impact van datasamenstelling, met name het gebruik van synthetische data voor klassenbalans.

*Hypothese 1: Datasets aangevuld met synthetische data (GAN's) presteren beter dan datasets met een sterke disbalans.* Datasetonbalans, veelvoorkomend bij zeldzame medische aandoeningen, bevoordelt vaak de dominante klasse. Het aanvullen van minderheidsklassen met synthetische data (gegenereerd door GAN's) verbetert hun representatie, wat essentieel is voor een hogere recall (Dong et al., 2020).

*Hypothese 2: Een GRU-model presteert beter dan een eenvoudig lineair model voor ECG-classificatie.* Gated Recurrent Units (GRU's) zijn ontworpen voor sequentiële data zoals ECG-signalen en kunnen complexe temporele afhankelijkheden modelleren. Onderzoek bevestigt dat GRU-gebaseerde modellen significant betere resultaten behalen dan lineaire modellen (Sahoo et al., 2021).

*Hypothese 3: Een CNN-model presteert beter dan een eenvoudig lineair model voor ECG-classificatie.* Convolutionele Neurale Netwerken (CNN's) zijn zeer effectief in het detecteren van lokale patronen in signalen, zoals de kenmerkende golven in ECG's. Ze overtreffen lineaire modellen, die deze structurele informatie niet benutten, wat leidt tot hogere nauwkeurigheid en recall (Rana et al., 2023).

*Hypothese 4: Een CNN-model kan beter presteren dan een GRU-model voor ECG-classificatie, met name voor korte, vaste-lengte fragmenten.* Voor korte, vaste-lengte ECG-fragmenten kunnen CNN's soms efficiënter patronen leren dan recurrente netwerken, vooral wanneer relevante patronen lokaal zijn en minder afhankelijk van lange-termijn context. Studies tonen aan dat CNN's in bepaalde medische tijdreeksanalyses superieur kunnen zijn (Sahoo et al., 2021).

*Hypothese 5: Een CNN-model met Squeeze-and-Excitation (SE) blokken presteert beter dan een eenvoudig lineair model.* De integratie van Squeeze-and-Excitation (SE) blokken versterkt de representatiekracht van CNN's aanzienlijk door het model in staat te stellen de meest informatieve featurekanalen adaptief te versterken. Dit resulteert in betere prestaties dan eenvoudige modellen (Sahoo et al., 2021).

*Hypothese 6: Een CNN-model met Squeeze-and-Excitation (SE) blokken presteert beter dan een standaard CNN-model.* Door het toevoegen van SE-blokken leert een CNN zich te concentreren op de meest relevante delen van het signaal. Dit verhoogt de gevoeligheid voor subtiele, diagnostisch belangrijke signalen, wat resulteert in een hogere recall en nauwkeurigheid bij ECG-classificatie (Sahoo et al., 2021).

## Modellen

Om de opgestelde hypothesen te toetsen, is een raamwerk van vier specifieke modellen ontwikkeld. Elk model beschikt over een unieke architectuur van lagen en activatiefuncties, en is geoptimaliseerd met hyperparameters om hun effecten te kunnen analyseren.

Het Baseline Model is een eenvoudig feedforward neuraal netwerk dat bestaat uit twee volledig verbonden (lineaire) lagen, gescheiden door een ReLU (Rectified Linear Unit) activatiefunctie. Het model heeft een inputgrootte van 187 en een outputgrootte van 5, wat overeenkomt met het aantal invoerkenmerken en de te voorspellen klassen.

Het CNN-model (Convolutioneel Neuraal Netwerk) is ontworpen voor patroonherkenning in sequentiële data. Het maakt gebruik van 1D-convolutie voor kenmerkextractie, gevolgd door een ReLU-activatie en MaxPooling voor reductie van de dimensies. De data bestaat uit opeenvolgende waarden en is geen afbeelding, waarvoor CNN's vaak worden gebruikt. Omdat de data geen meerdere lagen bevat, is er slechts één inputkanaal. Het netwerk heeft twee lagen met respectievelijk 32 en 64 convolutiefilters en 64 neuronen. De eerste laag leert eenvoudige kenmerken, de tweede laag complexere patronen. Voor lokale patroondetectie wordt een kernel van 3 bij 3 gebruikt. Overfitting wordt tegengegaan met een dropout van 40%, waarbij willekeurig neuronen worden uitgeschakeld.

Het GRU-model (Gated Recurrent Unit) is gericht op sequentiële data en kan afhankelijkheden over langere tijd vastleggen. De kern is een GRU-laag met interne activatiefuncties zoals sigmoid en tanh, en een inputgrootte van 187, wat overeenkomt met het aantal kenmerken. Elke GRU-neuron verwerkt informatie uit 128 eerdere neuronen in de verborgen toestand. Het model bevat twee GRU-lagen, waardoor het in staat is complexere patronen te leren. Om overfitting te beperken, wordt in de eerste laag een dropout van 20% toegepast waarbij willekeurig neuronen worden uitgeschakeld.

Het CNN-SE-Skip-model is geavanceerder dan een standaard CNN en combineert CNN-componenten met "Squeeze-and-Excitation" (SE) blokken en skip-verbindingen. Het bestaat uit meerdere CNN-SE-Skip-blokken die gebruikmaken van 1D-convoluties, MaxPooling en een SqueezeExcitation-



Afbeelding 1: Schematische weergave van de modellen

module. Deze module past adaptieve gemiddelde pooling toe en bevat lineaire lagen met ReLU- en sigmoid-activaties. Skip-verbindingen verbeteren de informatieoverdracht tussen lagen en verlagen zo de loss functie. Het model verwerkt invoer met één kanaal, wat het geschikt maakt voor eendimensionale data zoals tijdreeksen. Elke laag bevat 32 filters en gebruikt een kernel van 3 om lokale patronen te herkennen. Het netwerk telt twee lagen en de verborgen lagen hebben een grootte van 64, wat de leercapaciteit en diepte bepaalt. Een dropout van 30% helpt overfitting te voorkomen. Elke laag gebruikt een reductieratio van 8 om kanaalgewichten efficiënt te leren. Tot slot schat een Softmax-laag de uitvoerclassen.

Tijdens de training wordt een batchgrootte van 32 gebruikt, wat zorgt voor een goede balans tussen geheugengebruik en stabiliteit. Het model traint maximaal 100 epochs, waarbij early stopping voorkomt dat het onnodig doorgaat zonder verbetering. Een ReduceLRonPlateau-scheduler verlaagt de leersnelheid met een factor 0.1 als de validatieloss drie opeenvolgende epochs niet verbetert. De initiële learning rate is 0.001, een gangbare startwaarde voor de Adam-optimizer. Early stopping treedt in werking bij stagnerende validatieprestaties, met een wachttijd van 20 epochs en een minimale verbetering van 0.001. L2-regularisatie wordt toegepast via een weight decay van 0.0005 om overfitting te beperken. Daarnaast worden class weights dynamisch berekend om onevenwichtige klassen in de dataset te compenseren en bias te voorkomen.

Tabel 2: Hyperparameteroverzicht

Parameter	Baseline Model	CNN Model	GRU Model	CNN-SE-Skip Model
input_size	187		187	
output_size	5	5	5	5
input_channels		1		1
hidden_size		64	128	64
num_layers		2	2	2
conv_filters		[32, 64]		32
kernel_size		3		3
dropout_rate		0.4	0.2	0.3
reduction_ratio				8

## Beste model selectie

De modellen worden onafhankelijk van elkaar getraind op twee varianten van dezelfde dataset: de oorspronkelijke data en een verrijkte met synthetische data versie. Om een objectieve vergelijking van de prestaties door dataset mogelijk te maken, blijven de hyperparameters gelijk. Dit helpt om de invloed van de data op de modelprestaties aan te tonen.

De doelstelling van dit onderzoek is het identificeren van de best presterende model, dat het beste voorspelt op onzichtbare testdata. Modellen worden getraind voor maximaal honderd epochs. De zoektocht naar het best presterende model houdt rekening met overfitting door een steeds de relatie tussen test- en training loss te vergelijken. Een early stopping mechanisme is geïmplementeerd om de efficiëntie van de training te optimaliseren. Indien de test loss binnen twintig opeenvolgende epochs geen duidelijke verbetering toont, wordt de training beëindigd. Dit signaleert dat verdere iteraties geen verdere prestatiewinst zullen opleveren en voorkomt onnodig trainen. De "beste epoch" wordt vastgesteld op basis van de laagste geregistreerde test loss.

De selectie van het optimale model richt zich op generalisatie, wat betekent dat het model goed presteert op ongeziene data. Het model wordt beter bij een afnemende test loss. Een patience counter bewaakt de test loss en wordt gereset bij daling en verhoogd bij stijging. De test loss zal doorgaans hoger zijn dan de training loss. Overfitting treedt pas op wanneer de training loss blijft dalen, terwijl de test loss niet meer verbetert of zelfs stijgt, wat duidt op memorisatie in plaats van generalisatie.

## Klassebalancerings

Deze twee tabellen tonen de verdeling van observaties (Tabel 3) over verschillende klassen binnen zowel de oorspronkelijke trainingsdata (traindata), de gecombineerde trainings- en synthetische data, als de testdata. De aanpassing in class weights (Tabel 4) helpt de invloed van dominante klassen op de berekening van de training loss te verminderen. Dit stelt het model in staat om nauwkeurigere voorspellingen te doen, vooral voor de niet-dominante klassen, wat resulteert in een evenwichtiger model.

Tabel 3: Aantallen en verdeling van observaties in datasets

dataset	observaties	class 0		class 1		class 2		class 3		class 4	
train	87554	72471	82.77%	2223	2.54%	5788	6.61%	641	0.73%	6431	7.35%
train+synthetic	217411	72471	33.33%	36235	16.67%	36235	16.67%	36235	16.67%	36235	16.67%
test	21892	18118	82.76%	556	2.54%	1448	6.61%	162	0.74%	1608	7.35%

Tabel 4: Class weights op basis van aantal observaties

train dataset	test dataset	train observaties	test observaties	class 0 weight	class 1 weight	class 2 weight	class 3 weight	class 4 weight
train	test	87554	21892	0.241625	7.8771	3.02536	27.3179	2.72287
train+synthetic	test	217411	21892	0.599994	1.20001	1.20001	1.20001	1.20001

# Analyse

De resultaten van het onderzoek zijn opgenomen in de bijlage. In deze analyse worden de experimentele bevindingen besproken met als doel de effectiviteit van verschillende modelarchitecturen en methoden van datapreparatie voor ECG-classificatie te evalueren. Voor elke opgestelde hypothese worden de resultaten afzonderlijk toegelicht, met nadruk op belangrijke prestatiestatistieken zoals recall, accuracy en het verloop van de lossfuncties tijdens het trainingsproces.

*Hypothese 1: Datasets aangevuld met synthetische data (GAN's) presteren beter dan datasets met een sterke disbalans.* De resultaten van het onderzoek geven een sterke ondersteuning van de hypothese voor de gebruikte modellen. Modellen die zijn getraind met de synthetisch aangevulde dataset lieten vrijwel altijd hogere recall- en accuracy-scores zien dan dezelfde modellen getraind op de kleinere, originele dataset. De toename in recall is hierbij bijzonder significant, wat wijst op een verbeterde herkenning van minderheidsklassen. Hoewel het relatieve verschil tussen de train loss en de test loss bij sommige modellen toenam bij gebruik van de grotere dataset, wat zou kunnen duiden op een complexer leerproces, doet dit niet af aan het feit dat synthetische data duidelijke verbeteringen opleveren in zowel recall als accuracy.

*Hypothese 2: Een GRU-model presteert beter dan een eenvoudig lineair model voor ECG-classificatie.* De hypothese wordt overtuigend ondersteund door de resultaten. De GRU-modellen presteerden consistent beter dan de eenvoudige lineaire modellen, zowel in termen van recall als accuracy. Dit gold voor diverse configuraties van hyperparameters en voor zowel de originele als de synthetisch aangevulde datasets. De kracht van GRU's om tijdsafhankelijkheden in sequentiële data, zoals ECG-signalen, te modelleren, komt duidelijk tot uiting in de verbeterde classificatieresultaten. Dit vormt een belangrijk voordeel ten opzichte van lineaire modellen, die daartoe niet in staat zijn.

*Hypothese 3: Een CNN-model presteert beter dan een eenvoudig lineair model voor ECG-classificatie.* Ook deze hypothese wordt duidelijk ondersteund door de resultaten. De CNN-modellen behaalden over de hele linie hogere recall- en accuracy-scores dan de baseline-modellen. Dit bevestigt de effectiviteit van Convolutionele Neuronale Netwerken in het herkennen van lokale patronen binnen ECG-signalen, zoals specifieke golfvormen. Het vermogen van CNN's om relevante kenmerken rechtstreeks uit ruwe signaaldata te extraheren, leidt tot een duidelijke verbetering in classificatieprestaties ten opzichte van eenvoudigere modellen, die deze complexiteit niet kunnen benutten.

*Hypothese 4: Een CNN-model kan beter presteren dan een GRU-model voor ECG-classificatie, met name voor korte, vaste-lengte fragmenten.* De resultaten bieden geen volledige ondersteuning voor deze hypothese. Hoewel de hoogste individuele recall-scores werden behaald door specifieke CNN-modellen (bijvoorbeeld 0.977114928), presteerden de GRU-modellen (met recalls tot 0.967522383) eveneens uitzonderlijk goed en kwamen ze dicht in de buurt van de prestaties van de CNN's. Er is geen consistent bewijs dat CNN's systematisch beter presteren dan GRU's in alle scenario's, zelfs niet bij korte, vaste-lengte fragmenten, waarvoor CNN's theoretisch in het voordeel zouden zijn. Beide architecturen blijken zeer geschikt voor de classificatietask, waarbij de uiteindelijke prestaties sterk afhangen van de gekozen hyperparameters en architectuurstellingen.

*Hypothese 5: Een CNN-model met Squeeze-and-Excitation (SE) blokken presteert beter dan een eenvoudig lineair model.* Deze hypothese wordt niet ondersteund en zelfs verworpen door de resultaten. De CNN-SE-Skip-modelconfiguraties presteerden in veel gevallen aanzienlijk slechter dan het lineaire model, zowel in termen van recall als accuracy. Dit was vooral zichtbaar bij modellen getraind op de kleinere dataset, waar de laagste recall van de gehele analyse werd gemeten (0.779599854). De veronderstelde voordelen van SE-blokken, zoals het adaptief versterken van informatieve kanalen, lijken in deze context niet tot hun recht te komen. Integendeel, ze kunnen zelfs leiden tot een verslechtering van de prestaties ten opzichte van eenvoudiger modellen zoals het lineaire model.

*Hypothese 6: Een CNN-model met Squeeze-and-Excitation (SE) blokken presteert beter dan een standaard CNN-model.* Ook deze hypothese wordt niet ondersteund en wordt verworpen door de resultaten. De CNN-SE-Skip-modellen vertonen geen consistente verbetering in recall en accuracy ten opzichte van het standaard CNN-model. Sterker nog, de standaard CNN-modellen behaalden in de meeste gevallen betere prestaties, waaronder de hoogste algemene recall (0.977114928). De toevoeging van SE-blokken lijkt binnen de geteste configuraties geen voordeel te bieden. Dit staat haaks op de theoretische voordelen van SE-blokken, die gericht zijn op het adaptief versterken van informatieve kanalen om de signaalgevoeligheid te verbeteren.

## Reflectie

De analyse van de resultaten biedt waardevolle inzichten in de effectiviteit van diverse benaderingen voor ECG-classificatie. Het meest overtuigende bewijs komt voort uit Hypothese 1, die de kracht van het uitbreiden van de dataset met synthetische data aantoonde en sterk wordt ondersteund. Dit onderstreept het belang van het aanpakken van data-onbalans, een veelvoorkomend probleem in medische toepassingen, waar zeldzame aandoeningen vaak ondervertegenwoordigd zijn. De resulterende toename in recall is van essentieel klinisch belang. Daarnaast bevestigen de resultaten van Hypothese 2 en Hypothese 3 dat complexere modellen, zoals Convolutionele Neuronale Netwerken (CNN's) en Gated Recurrent Units (GRU's), significant beter presteren dan eenvoudige lineaire modellen bij het verwerken van sequentiële ECG-data. Dit benadrukt het belang van het kiezen van een modelarchitectuur die van nature geschikt is om tijdsgebonden en lokale patronen te herkennen.

Hoewel de verschillen tussen CNN's en GRU's (Hypothese 4) op basis van de resultaten minder uitgesproken zijn dan soms verwacht, tonen beide modeltypen aan dat ze tot zeer hoge prestaties in staat zijn, afhankelijk van de specifieke implementatie en optimalisatie. Dit benadrukt dat modelkeuze niet los gezien kan worden van de context waarin het wordt toegepast. De prestaties van de CNN-modellen met Squeeze-and-Excitation (SE) blokken (Hypothese 5 en 6) vormen de meest opvallende bevinding. Ondanks hun theoretische voordelen en bewezen effectiviteit in andere domeinen, lieten deze modellen geen verbetering zien en presteerden ze zelfs slechter dan zowel het lineaire model als de standaard CNN-modellen. Dit suggereert dat het toevoegen van geavanceerde architectuurcomponenten niet automatisch leidt tot betere resultaten. Factoren zoals de aard van de ECG-data, de implementatie van de SE-blokken, de afstemming van hyperparameters of de interactie met andere architectuurelementen kunnen hierbij een rol hebben gespeeld. Het onderstreept dat "complexer" niet per definitie "beter" betekent, en dat zorgvuldige validatie en optimalisatie in de praktijk essentieel blijven.

Een opvallende uitkomst was dat het CNN-SE-Skip model niet beter presteerde dan simpelere modellen, ondanks dat we dat wel hadden verwacht. Dit kan betekenen dat de extra onderdelen zoals Squeeze-and-Excitation (SE) blokken en skip-verbindingen niet goed passen bij de eigenschappen van één-dimensionale ECG-signalen met slechts één kanaal. Het laat zien dat een complexer model niet altijd betere resultaten geeft. Het is daarom belangrijk om goed te kijken of geavanceerde technieken echt passen bij het type data. Ook is het mogelijk dat de instellingen (hyperparameters) van dit model nog niet optimaal waren.

Een tweede opvallende bevinding was dat alle modellen, zelfs het eenvoudige lineaire model, duidelijk beter presteerden na toevoeging van synthetische data. Dat was verrassend, omdat de verwachting was dat vooral complexe neurale netwerken hiervan zouden profiteren. Dit resultaat laat zien dat de gegenereerde data van hoge kwaliteit is: het hielp niet alleen om meer voorbeelden te creëren, maar maakte het ook makkelijker voor de modellen om de verschillende klassen van elkaar te onderscheiden. Dit benadrukt hoe belangrijk een evenwichtige en representatieve dataset is. Synthetische data genereren met slimme technieken zoals GAN's blijkt heel nuttig, vooral in de medische wereld.

## Aanbevelingen

De tegenvallende prestaties van het CNN-model met SE-blokken lijken vooral te verklaren door een te hoge complexiteit van het model in verhouding tot de beschikbare hoeveelheid trainingsdata, zelfs met synthetische uitbreidingen. Dit leidde tot hogere testverliezen en tekenen van onderfitting of instabiele training. Daarnaast waren de gekozen hyperparameters waarschijnlijk niet optimaal afgestemd op deze complexere architectuur. Modellen met SE-blokken zijn gevoeliger voor instellingen zoals de learning rate, weight decay, dropout rate en de zogeheten reduction ratio binnen de SE-blokken. Zonder zorgvuldige afstemming kunnen deze instellingen de prestaties negatief beïnvloeden. Het is ook goed mogelijk dat de voordelen van SE-blokken, zoals het versterken van belangrijke signaalkenmerken per kanaal, minder goed tot hun recht komen bij korte, één-dimensionale ECG-fragmenten. In dat geval kan een eenvoudiger CNN-model, zonder deze extra lagen, juist effectiever zijn.

Voor toekomstige verbetering zijn er daarom duidelijke aanknopingspunten:

- Hyperparameter tuning: systematisch experimenteren met instellingen zoals learning rate schedules, dropout en reduction ratio's.
- Architectuurevaluatie: kritisch kijken naar de opbouw van het model, het aantal lagen beperken of de plaatsing van SE-blokken heroverwegen.
- Datakwaliteit en -hoeveelheid: uitbreiden of verbeteren van de dataset om complexere modellen beter te ondersteunen.

Door deze aspecten zorgvuldig te onderzoeken, kan mogelijk alsnog het potentieel van SE-blokken voor ECG-classificatie worden benut.

## Conclusie

Dit onderzoek richtte zich op het voorspellen van hartziekten via ECG-signalen met deep learning. Het doel was automatische classificatie van normale en afwijkende hartritmes, inclusief de aard van de afwijking. De resultaten tonen aan dat synthetische data, gegenereerd met GAN's, de prestaties verbeteren, vooral de recall, welke cruciaal om gemiste diagnoses te voorkomen. GRU- en CNN-modellen presteerden duidelijk beter dan het lineaire model bij het herkennen van complexe patronen in ECG-data. De veronderstelling dat CNN's met SE-blokken superieur zouden zijn, werd echter niet bevestigd. Dit benadrukt dat meer complexiteit niet altijd betere resultaten oplevert en dat zorgvuldige afstemming van architectuur en hyperparameters essentieel blijft.

## Referenties

Dong, J., Yu, H., & Huang, J. (2020). *Deep Learning for ECG Analysis*. IEEE.

Rana, M., Siddique, A., & Ahmad, N. (2023). Interpretation of Electrocardiogram Heartbeat by CNN and GRU. *arXiv preprint arXiv:2302.17887*.

Sahoo, D., Dash, P., Barik, R., & Mohanty, S. (2021). CNN-GRU model for ECG signal classification using UCR time series data. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-6). IEEE.

Resultaten

model_type	train_samples	test_samples	epochs_run	best_epoch	best_learning_rate	weight_decay	conv_filters	dropout_rate	hidden_size	input_channels	input_size	input_size after_flattening	kernel_size	num_features	num_layers	output_size	reduction_ratio	use_dropout	best_test_loss	best_accuracy	best_recall	best_precision	best_f1	best_loss difference_abs	best_loss difference_rel_perc
baseline_model	87554	21892	43	8	0,001	0,0001					187					5			0,22961934	0,92682258	0,92682258	0,95427893	0,93722419	0,03105088	13,52276340
baseline_model	217411	21892	38	9	0,001	0,0001					187					5			0,17132946	0,94961630	0,94961630	0,95575993	0,95201593	0,07921491	46,23543120
baseline_model	87554	21892	45	5	0,001	0,0005					187					5			0,26258429	0,91261648	0,91261648	0,94616717	0,92511430	0,04388090	16,71116900
baseline_model	217411	21892	41	7	0,001	0,0005					187					5			0,22098454	0,93563859	0,93563859	0,95112494	0,94145639	0,10552157	47,75065770
cnn_model	87554	21892	40	12	0,001	0,0001	16, 32	0,5	32	1	1472	3	187	2	5			ONWAAR	0,12512941	0,96327426	0,96327426	0,96956720	0,96552165	0,02725234	21,77932060
cnn_model	217411	21892	38	10	0,001	0,0001	16, 32	0,5	32	1	1472	3	187	2	5			ONWAAR	0,09457621	0,97432852	0,97432852	0,97388315	0,97402386	0,06265915	66,25254980
cnn_model	87554	21892	54	12	0,001	0,0005	16, 32	0,4	32	1	1472	3	187	2	5			ONWAAR	0,14509746	0,95569158	0,95569158	0,96717846	0,95978637	0,01954219	13,46831850
cnn_model	217411	21892	49	10	0,001	0,0005	16, 32	0,4	32	1	1472	3	187	2	5			ONWAAR	0,11201363	0,96921250	0,96921250	0,97043762	0,96960208	0,06201825	55,36669720
cnn_model	87554	21892	39	10	0,001	0,0001	32, 64	0,3	64	1	2944	5	187	2	5			WAAR	0,12641183	0,95614836	0,95614836	0,96940394	0,96102984	0,00089838	0,71067772
cnn_model	217411	21892	45	10	0,001	0,0001	32, 64	0,3	64	1	2944	5	187	2	5			WAAR	0,08176687	0,97711493	0,97711493	0,97738910	0,97719780	0,04572868	55,92568670
cnn_model	87554	21892	52	6	0,001	0,0005	32, 64	0,3	64	1	2944	5	187	2	5			WAAR	0,12554890	0,96025946	0,96025946	0,96860239	0,96325449	0,07034767	56,03208770
cnn_model	217411	21892	55	9	0,001	0,0005	32, 64	0,3	64	1	2944	5	187	2	5			WAAR	0,09332449	0,97268409	0,97268409	0,97216643	0,97234820	0,03759367	40,28274500
cnn_se_skip_model	87554	21892	68	11	0,001	0,0001	32	0,3	64	1	1472	3	187	2	5	8		ONWAAR	1,02613679	0,87817468	0,87817468	0,94129226	0,90296420	0,00262101	0,25542512
cnn_se_skip_model	217411	21892	41	8	0,001	0,0001	32	0,3	64	1	1472	3	187	2	5	8		ONWAAR	0,94119070	0,96386808	0,96386808	0,96535882	0,96446708	0,01467143	1,55881632
cnn_se_skip_model	87554	21892	49	26	0,00001	0,0005	32	0,3	64	1	1472	3	187	2	5	8		ONWAAR	0,96586944	0,94381509	0,94381509	0,96054934	0,94979539	0,00638874	0,66144996
cnn_se_skip_model	217411	21892	48	8	0,001	0,0005	32	0,3	64	1	1472	3	187	2	5	8		ONWAAR	0,96334182	0,94294720	0,94294720	0,95394055	0,94720868	0,02270957	2,35737383
cnn_se_skip_model	87554	21892	46	13	0,001	0,0001	64	0,4	128	1	1472	5	187	3	5	16		WAAR	1,10808227	0,79732322	0,79732322	0,88005398	0,82367824	0,14657433	13,22774790
cnn_se_skip_model	217411	21892	19	4	0,001	0,0001	64	0,4	128	1	1472	5	187	3	5	16		WAAR	1,01550709	0,88918326	0,88918326	0,92501699	0,90299571	0,03109076	3,06159925
cnn_se_skip_model	87554	21892	25	21	0,00001	0,0005	64	0,4	128	1	1472	5	187	3	5	16		WAAR	1,12454862	0,77959985	0,77959985	0,86002949	0,81164492	0,14902965	13,25239760
cnn_se_skip_model	217411	21892	91	18	0,001	0,0005	64	0,4	128	1	1472	5	187	3	5	16		WAAR	0,99469997	0,90941897	0,90941897	0,93600429	0,91977930	0,00066882	0,06723872
gru_model	87554	21892	52	14	0,001	0,0001	0	0	64		187			1	5				0,16639980	0,94970766	0,94970766	0,96240495	0,95423311	0,02953212	17,74768840
gru_model	217411	21892	40	8	0,001	0,0001	0	0	64		187			1	5				0,13797341	0,95875206	0,95875206	0,96169882	0,95986338	0,06708270	48,62001900
gru_model	87554	21892	40	7	0,001	0,0005	0	0	64		187			1	5				0,21686033	0,92910652	0,92910652	0,95441654	0,93810798	0,00413763	1,90797183
gru_model	217411	21892	49	17	0,0001	0,0005	0	0	64		187			1	5				0,15064450	0,95491504	0,95491504	0,96135948	0,95741850	0,07680801	50,98626620
gru_model	87554	21892	58	12	0,001	0,0001	0	0,2	128		187			2	5				0,15951874	0,94724100	0,94724100	0,96470686	0,95345987	0,01002267	6,28306885
gru_model	217411	21892	63	31	0,001	0,0001	0	0,2	128		187			2	5				0,10585022	0,96752238	0,96752238	0,96968028	0,96840984	0,03852238	36,39329450
gru_model	87554	21892	54	17	0,0001	0,0005	0	0,2	128		187			2	5				0,17324428	0,94564224	0,94564224	0,96317682	0,95182576	0,00573547	3,31062696
gru_model	217411	21892	49	15	0,001	0,0005	0	0,2	128		187			2	5				0,14473344	0,95468664	0,95468664	0,95929008	0,95650920	0,02769734	19,13679330