UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Een experimentele evaluatie van TOAD voor event log-analyse.

Michiel Borgers

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Mieke JANS

BEGELEIDER :

Mevrouw Manal LAGHMOUCH

2024
2025

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Een experimentele evaluatie van TOAD voor event log-analyse.

Michiel Borgers

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Mieke JANS

BEGELEIDER :

Mevrouw Manal LAGHMOUCH

Een experimentele evaluatie van TOAD voor event log-analyse.

Michiel Borgers, Mieke Jans, Manal Laghmouch

ABSTRACT: Procesafwijkingen komen voor in het merendeel van de bestaande processen. Ze kunnen enerzijds bijdragen aan procesverbetering en positief zijn van aard. Anderzijds kunnen ze in uitzonderlijke gevallen wijzen op ernstige verstoringen zoals fraude, defecte machines of cyberaanvallen. Het onderscheid tussen *exceptions*, procesafwijkingen die als toelaatbaar worden beschouwd, en *anomalieën*, procesafwijkingen die voortkomen uit ernstige oorzaken, is daarbij cruciaal. Anomalieën vereisen verder onderzoek om de oorzaak te kunnen achterhalen en mogelijks aan te pakken. *Exceptions* vereisen geen verder onderzoek. Toch is het essentieel om beide types procesafwijkingen te detecteren, omdat het niet opmerken van anomalieën ernstige gevolgen kan hebben. *Event logs* bevatten informatie over procesuitvoeringen en laten toe om procesafwijkingen te identificeren via analyse. In dit onderzoek wordt Trace Ordering for Anomaly Detection (TOAD) onderzocht, een *unsupervised* clustertechniek die focust op collectieve temporele procesafwijkingen. Collectieve temporele procesafwijkingen zijn groepen van procesafwijkingen die gelijkaardige versnellingen of vertragingen van de procesuitvoering vertonen. De focus op dat type procesafwijkingen maakt TOAD uniek binnen het domein van *anomaly detection*. Het doel van dit onderzoek is om TOAD te evalueren op zijn vermogen om collectieve temporele procesafwijkingen te detecteren en om na te gaan of TOAD een onderscheid kan maken tussen *exceptions* en anomalieën tijdens de clustering. Vijftig experimenten werden uitgevoerd waarin TOAD telkens toegepast werd op één synthetisch gegenereerde *event log*. De prestaties van TOAD werden beoordeeld via vijf evaluatiemetrieken: het aantal gevormde niet-*noise* clusters, *precision*, *recall*, F1-score en cluster-*purity*. De resultaten van dit onderzoek tonen aan dat TOAD slechts beperkt in staat is om de collectieve temporele procesafwijkingen uit de gebruikte *event logs* accuraat te detecteren. Vooral anomalieën bleken moeilijk herkenbaar. Bovendien slaagt TOAD er niet in om systematisch onderscheid te maken tussen *exceptions* en anomalieën. Op basis van die bevindingen worden in deze studie meerdere pistes voor toekomstig onderzoek voorgesteld die kunnen bijdragen aan een beter begrip en een verdere verbetering van het TOAD-model.

Keywords: *TOAD, Trace Ordering for Anomaly Detection, Exceptions, Anomalies, Rare Category Detection, Collectieve procesafwijkingen, Temporele procesafwijkingen, Clustering*

1 Inleiding

Trace Ordering for Anomaly Detection (TOAD) is een *unsupervised* clustertechniek, die gebruikt wordt voor het detecteren van collectieve temporele procesafwijkingen. Procesafwijkingen zijn entiteiten binnen een proces, die abnormaal gedrag vertonen ten opzichte van een normatief procesmodel [16]. Een normatief procesmodel is het meest restrictieve type procesmodel. Het beschrijft het toegelaten procesgedrag en identificeert alle procesafwijkingen [11]. In de praktijk wijken procesuitvoeringen vaak af van het normatief procesmodel [3]. Afwijkende entiteiten binnen een proces kunnen alleenstaande gebeurtenissen of *events* zijn, maar ook opeenvolgingen van gebeurtenissen of *traces* [16]. Een collectieve procesafwijking is een cluster van *traces* die gelijkaardig afwijkend gedrag vertonen [12, 16, 19, 20]. Temporele procesafwijkingen vertonen afwijkend gedrag in de tijdsdimensie. In die tijdsdimensie zijn er twee soorten afwijkingen mogelijk, versnellingen en vertragingen. Temporele procesafwijkingen zijn dus entiteiten in een proces waarbij een versnelling of vertraging van de uitvoering plaatsvond [15]. TOAD detecteert geen individuele *outliers*, maar vormt clusters waarbij de *traces* in een cluster gelijkaardige temporele procesafwijkingen vertonen [2, 16, 20]. De *traces* in een cluster hebben vaak een gemeenschappelijke oorzaak voor hun afwijkend gedrag [15, 16, 20]. Wanneer die oorzaak vervolgens opgelost wordt, wordt het afwijkend gedrag van alle *traces* aangepakt, wat efficiënter is dan een afzonderlijke analyse en oplossing voor elke *trace* [15, 16, 20]. Door die collectieve procesafwijkingen te identificeren op basis van temporele afwijkingen, hanteert TOAD een aanpak die nieuw en uniek is binnen het domein van *process mining* [16].

Procesafwijkingen kunnen in twee groepen onderverdeeld worden [3, 10, 11, 13]. Een eerste groep bestaat uit *exceptions*. Die groep is doorgaans het grootst en bevat procesafwijkingen die irrelevant zijn om verder te onderzoeken [10, 11]. *Exceptions* zijn irrelevant om verder te onderzoeken, omdat ze toelaatbaar afwijkend gedrag vertonen en garanderen dat het proces voldoende flexibiliteit bevat om effectief te functioneren [3]. Een tweede groep bestaat uit de anomalieën. Die groep is vaak veel kleiner en bevat procesafwijkingen die actief onderzocht moeten worden, maar moeilijker te identificeren zijn [10, 11, 13]. Het verder onderzoeken van die anomalieën is belangrijk, omdat ze vaak leiden tot ongewenste bedrijfsresultaten [3].

Procesafwijkingen worden gecapteerd in *event logs* en kunnen door de analyse van die *logs* geïdentificeerd en geanalyseerd worden [8]. Ongeacht de aard, *exception* of anomalie, is het noodzakelijk om alle procesafwijkingen te detecteren [16]. Afwijkend procesgedrag kan enerzijds een teken van procesflexibiliteit zijn, een positieve impact hebben op het proces of leiden tot verbeteringen of vernieuwingen aan het proces [3, 16]. Anderzijds kan afwijkend procesgedrag in uitzonderlijke gevallen een indicatie zijn van fraude, het systematisch niet naleven van vooropgestelde regels, cyberaanvallen of storingen en defecten van machines [3, 8, 16, 19].

In dit onderzoek wordt TOAD toegepast voor *event log*-analyse, met als doel de prestaties van TOAD te evalueren in het detecteren van collectieve temporele procesafwijkingen. Daarnaast wordt ook onderzocht of TOAD in staat is om een onderscheid te maken tussen *exceptions* en anomalieën bij het vormen van clusters. TOAD vormt een interessant onderzoeksonderwerp vanwege de expliciete focus op het collectieve en temporele karakter van de procesafwijkingen. Die twee kenmerken onderscheiden TOAD

van andere technieken en bieden een vernieuwende invalshoek voor het detecteren van procesafwijkingen [16]. De bijdrage van dit onderzoek ligt in de beoordeling van een relatief nieuwe en unieke clustertechniek die in de literatuur nog maar beperkt empirisch werd geëvalueerd. Daarnaast wordt voor het eerst onderzocht of TOAD in de clustering ook een onderscheid kan maken tussen *exceptions* en anomalieën, wat een duidelijke *research gap* adresseert. Met dit onderzoek wordt ook breder onderzocht of TOAD überhaupt toepasbaar is binnen het *process mining*-domein.

Voor de uitvoering van dit onderzoek werden vijftig experimenten opgezet waarbij telkens een *event log* met procesafwijkingen werd geanalyseerd met TOAD. Binnen elk experiment werd expliciet gewerkt met een vooraf gedefinieerd onderscheid tussen *exceptions* en anomalieën. Er werd gekozen voor vijftig experimenten, om mogelijke bias te reduceren en om voldoende empirische onderbouwing te garanderen. De resultaten tonen aan dat TOAD minder goed presteert dan verwacht in het detecteren van collectieve temporele procesafwijkingen. Hoewel een deel van de afwijkingen correct werd geïdentificeerd, bleef de totale detectie accuraatheid beperkt. Wat betreft het onderscheid maken tussen *exceptions* en anomalieën presteerde TOAD zeer zwak. TOAD slaagde er slechts zeer beperkt in om dit onderscheid op een duidelijke en correcte manier te maken.

De rest van de paper is als volgt opgebouwd. In sectie 2 wordt er gekeken naar gerelateerd onderzoek. Sectie 3 zal verdere toelichting geven bij de werking van het TOAD-model. Sectie 4 gaat vervolgens dieper in op de methodologie die in dit onderzoek gehanteerd werd. De resultaten van de experimenten worden toegelicht in sectie 5. Sectie 6 reflecteert op de bevindingen uit die resultaten en bespreekt de beperkingen van dit onderzoek. In sectie 7 wordt de conclusie van dit onderzoek geformuleerd en worden er toekomstige onderzoekspistes geïdentificeerd.

2 Gerelateerd onderzoek

De detectie van anomalieën heeft tot doel ongewone gebeurtenissen in data te herkennen, omdat die afwijkingen kunnen duiden op ernstige gebeurtenissen zoals netwerkinbraken, frauduleuze transacties of afwijkingen in industriële processen. Anomaliedetectie valt doorgaans uiteen in verschillende categorieën, waarbij *machine learning*-technieken een prominente rol spelen. Die technieken onderscheiden zich door hun vermogen om patronen in data te leren, zodat normale en abnormale instanties van elkaar gescheiden kunnen worden. Hierbij wordt in de literatuur vaak onderscheid gemaakt tussen *supervised*, *semi-supervised* en *unsupervised* methoden, afhankelijk van de beschikbaarheid van gelabelde anomalieen normdata [5, 7, 12]. In de volgende secties worden vier veelvoorkomende soorten *machine learning*-benaderingen voor anomaliedetectie behandeld: classificatie, nearest neighbor, statistische methoden en clustering.

2.1 Classificatie

Classificatiegebaseerde anomaliedetectie berust op het idee dat een *classifier* in de *feature*-ruimte onderscheid kan maken tussen normale en abnormale gegevenspunten. Kort gezegd leert een classificatiemodel op basis van getrainde voorbeelden welke patronen overeenkomen met 'normaal' en welke daarvan afwijken. Tijdens de testfase kent het model vervolgens een nieuw waargenomen instantie het

label ‘normaal’ of ‘anomalie’ toe . In de praktijk worden diverse classificatietechnieken ingezet. Enkele voorbeelden zijn neurale netwerken, Support Vector Machines (SVM), Bayesiaanse netwerken en *rule-based* technieken. Een belangrijk voordeel van classificatietechnieken is dat ze zeer krachtige algoritmen bieden om onderscheid te maken tussen verschillende dataklassen, waardoor ze bij voldoende trainingsvolume en representatieve labels vaak hoge detectienauwkeurigheden bereiken. Classificatiemethoden hebben echter ook nadelen. Ze vereisen doorgaans zorgvuldig gelabelde trainingsdata, maar in anomaliedetectietoepassingen is het vaak erg lastig om representatieve anomalielabels te verkrijgen, omdat afwijkingen per definitie zeldzaam zijn. Wanneer de labels onvolledig of verkeerd zijn, kan de prestaties van het model ook sterk verslechteren [5, 7, 12, 19].

2.2 *Nearest Neighbor*

Nearest neighbor–gebaseerde anomaliedetectie gaat ervan uit dat normale datapunten zich in dichtbevolkte buurten bevinden, terwijl anomalieën ver van hun dichtstbijzijnde buren staan. Door voor elke instantie de afstand tot de k-dichtstbijzijnde buren te meten, zoals bij KNN, of de lokale dichtheid te vergelijken met die van de omgeving, zoals bij LOF, kunnen punten met abnormaal grote afstands- of lage dichtheidsscores als anomalieën worden beschouwd. Een groot voordeel is dat deze methoden *unsupervised* zijn en geen aannames doen over de onderliggende distributie, maar ze kunnen veel *false positives* genereren wanneer er onvoldoende vergelijkbare normale buren zijn en zijn rekenintensief voor grote of hoge-dimensionale datasets [5, 12, 19].

2.3 Statistische methoden

Statistische anomaliedetectietechnieken veronderstellen dat normale datapunten zich bevinden in regio’s met hoge kansdichtheid van een stochastisch model, terwijl anomalieën in regio’s met lage kansdichtheid vallen. Parametrische methoden gaan uit van een vooraf gedefinieerde verdeling en voeren hypothesetoetsen uit om punten als anomalieën te bestempelen wanneer de null-hypothese wordt verworpen . Niet-parametrische technieken doen geen eerdere aannames over de onderliggende verdeling en detecteren anomalieën op basis van data-gestuurde dichtheidskenmerken in lineaire tijd . Hybride benaderingen combineren statistische tests met *machine learning*, bijvoorbeeld door een SVM te koppelen aan lineaire regressie . Een belangrijk voordeel van parametrische methoden is dat een robuuste kansverdeling kan fungeren als een *unsupervised* detector, maar in hoge-dimensionale realistische datasets voldoet de data vaak niet aan de veronderstelde distributie, wat meerdere hypothesetests vereist en het modelleren van gecombineerde eigenschappen bemoeilijkt. Niet-parametrische technieken maken minder aannames, maar zijn gevoelig voor parameterkeuzes en generaliseren soms niet goed naar verschillende datatypen [5, 12, 19].

2.4 Clustering

Clusteringgebaseerde anomaliedetectie begint met de veronderstelling dat normale datapunten grote, dichte clusters vormen, terwijl anomalieën zich buiten deze clusters bevinden of slechts kleine, verspreide clusters vormen. Een veelgebruikte clustertechniek is K-means. Een belangrijk voordeel van

clusteringgebaseerde benaderingen is dat de testfase doorgaans snel verloopt, omdat clustervaliditeit efficiënt kan worden gecontroleerd zonder zware rekenlast tijdens de detectie . Tegelijkertijd bestaat het risico dat anomalieën die niet tot een duidelijk afgebakende cluster behoren over het hoofd worden gezien, wat leidt tot *false positives* wanneer er geen significante cluster rond bepaalde normale punten bestaat[5, 12, 19].

3 Trace Ordering for Anomaly Detection

TOAD is een *unsupervised* clustertechniek, die gebruikt wordt voor het detecteren van collectieve temporele procesafwijkingen. In deze sectie wordt de concrete werking van het TOAD-model besproken. De beschrijving van het TOAD-model is gebaseerd op de toelichting in het onderzoek van Richter et al. [16]. Eerst zal er dieper ingegaan worden op de vereiste input van het model. Daaropvolgend wordt de berekening van temporele relaties tussen *events* en de bepaling van temporele afwijkingen uitgelicht. Tot slot wordt er dieper ingegaan op het ordenen van *traces* en de manier waarop procesafwijkingen door TOAD precies gedetecteerd worden.

3.1 Input

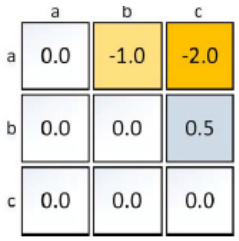
De vereiste input voor het TOAD-model is een *process event log*. Die bevat informatie over de verschillende activiteiten van een proces. Voor elke activiteit die uitgevoerd wordt, wordt er een *event* aangemaakt. Dat *event* bevat informatieattributen over de uitvoering van de activiteit en wordt opgeslagen in de *process event log* [1]. Die *events* moeten voor het TOAD-model minimaal drie specifieke attributen bevatten. Een attribuut dat aangeeft welke procesactiviteit plaatsvond, een attribuut dat de *timestamp* bevat en een attribuut dat de betrokken *case* identificeert. Een *case* is een specifieke uitvoering van het proces en wordt geïdentificeerd met een unieke *case identifier* [15]. Bijkomende attributen zijn toegestaan, maar niet noodzakelijk voor het correct uitvoeren van het TOAD-model. Een *trace* geeft voor één bepaalde *case* de volgorde weer van de verschillende events die plaatsgevonden hebben. De *timestamps* van procesactiviteiten worden door het TOAD-model beschouwd als een volwaardig onderdeel van een *trace*, door de sterke focus van het model op temporele afwijkingen.

3.2 *Temporal Deviation Signatures*

Het TOAD-model start met het extraheren van alle *event*-paren die voorkomen in de *event log*. Een *event*-paar is geldig wanneer beide *events* voorkomen in minstens eenzelfde *trace*. Vervolgens wordt er een set van doorlooptijden aangemaakt voor elk *event*-paar. Die set bevat voor voorkomen van een *event*-paar in de *event log* de doorlooptijd tussen de twee *events*. Op basis van die doorlooptijden set berekent het TOAD-model voor elk *event*-paar de gemiddelde doorlooptijd en de standaardafwijking.

Daarna wordt er een z-*scoring* uitgevoerd. Die z-*scoring* is een standaardisatie, die ervoor zorgt dat *event*-paren met korte en lange doorlooptijden op één schaal geplaatst kunnen worden. *Event*-paren met Z-scores die dicht bij 0 liggen, vertonen normaal procesgedrag. Als de z-score van een *event*-paar echter ver van 0 ligt, is er sprake van een temporele procesafwijking.

Vervolgens wordt er voor elke *case* in de *event log* een *temporal deviation signature* aangemaakt. Een *temporal deviation signature* is een matrix die de z-scores bevat voor alle bestaande *event*-paren in de *event log*. Als een *event*-paar niet voorkomt in de *trace* van een *case*, wordt de score op 0 gezet. Figuur 1, uit [16], toont een illustratief voorbeeld van een eenvoudige *temporal deviation signature*-matrix. Vervolgens worden die matrixen omgevormd tot vectoren, die voor elke *trace* het patroon van alle temporele procesafwijkingen bevat.



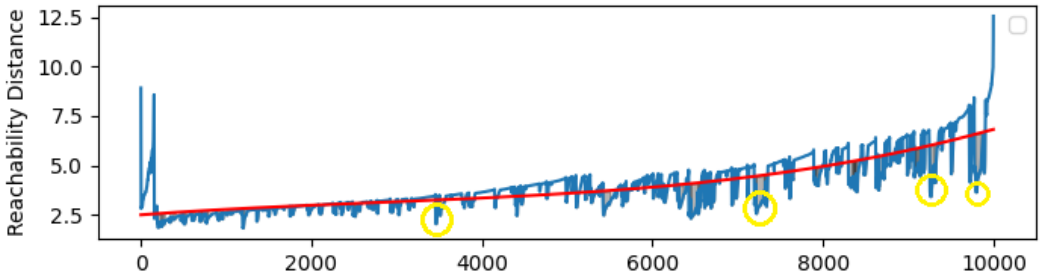
Figuur 1: Vereenvoudigde weergave van een *Temporal Deviation Signature*-matrix [16]

3.3 Ordenen van *traces*

Nadat voor elke *case* een *temporal deviation signature* is aangemaakt en omgezet in een vector, beschouwt het TOAD-model deze *signatures* als punten in een d-dimensionale vectorruimte, waarbij d gelijk is aan de lengte van de vectoren. De volgende stap is om in die datapunten collectieve procesafwijkingen of clusters te identificeren. Daarvoor gebruikt het TOAD-model de dichtheidsgebaseerde clustermethode OPTICS. Dichtheidsgebaseerde clustering is een techniek waarbij de clusters gevormd worden op basis van dichtheid in de data. OPTICS bepaalt die dichtheid aan de hand van de MinPts hyperparameter. MinPts staat voor het minimale aantal buurpunten in een gebied dat nodig is om te spreken van een buurt met hoge dichtheid. Voor elk punt in de vectorruimte berekent OPTICS vervolgens de kernafstand. Die kernafstand is de afstand waarvoor het aantal buurpunten binnen die afstand gelijk is aan MinPts.

Daarna gaat OPTICS verder met het berekenen van de *reachability*-afstanden tussen puntparen. Die *reachability*-afstand kan twee waardes aannemen. De *reachability*-afstand kan gelijk zijn aan de kernafstand van het eerste datapunt of aan de Euclidische afstand tussen de twee datapunten, afhankelijk van welke afstand het grootst is.

Tot slot gaat OPTICS de datapunten ordenen op basis van de berekende *reachability*-afstanden. Het eerste datapunt van de ordening wordt willekeurig gekozen. Vervolgens wordt het punt in de ordening toegevoegd dat de kleinste *reachability*-afstand heeft tot één van de reeds geordende punten. De ordening van de *traces* wordt daarna geplot op de x-as en op de y-as wordt de *reachability*-afstand geplot. Figuur 2 toont een voorbeeld van zo een OPTICS plot. Visueel ontstaan er op de plot dalen doordat verschillende punten erg dicht bij elkaar liggen en de *reachability*-afstand tussen die punten dus klein is. Die dalen representeren de clusters van *traces* met gelijkaardige temporele procesafwijkingen. Op figuur 2 zijn enkele clusterdalen aangeduid.



Figuur 2: Voorbeeld van een OPTICS plot met clusteridentificatie

3.4 Anomaliedetectie

Wanneer OPTICS op de klassieke wijze wordt toegepast, bepaalt de gebruiker een vereist dichtheidsniveau. Vervolgens worden de dalen, die onder dat dichtheidsniveau vallen, geïdentificeerd als de clusters. De clusters moeten dus een hogere dichtheid hebben dat het vereiste dichtheidsniveau. Het TOAD-model automatiseert echter die keuze van het dichtheidsniveau. Daardoor richt het model zich niet op clusters met een gelijke dichtheid, maar juist op de kleinere clusters van procesafwijkingen die visueel gerepresenteerd worden door de smalle en diepe dalen.

Het TOAD-model hanteert een automatische detectie van de clusters door het uitvoeren van een baseline vergelijking. Het TOAD-model bouwt op de plot een baseline door een *smoothening* uit te voeren met de Savitzky-Golay filter op de *reachability*-curve. Die filter wordt gebruikt om ruis te onderdrukken zonder de hoge frequentiecomponenten te verwijderen. Het TOAD-model trekt vervolgens de oorspronkelijke *reachability*-afstand af van de baseline waardoor er duidelijke afwijkingen ontstaan. Tot slot past het TOAD model een piekdetectie-algoritme toe om de clusters definitief te identificeren.

4 Methodologie

Om de effectiviteit van TOAD te beoordelen, wordt in dit onderzoek onderzocht in hoeverre het model erin slaagt om collectieve temporele procesafwijkingen te detecteren en of het daarbij een onderscheid kan maken tussen *exceptions* en anomalieën. In deze sectie wordt de methodologie toegelicht die gehanteerd werd om de onderzoeksvragen te beantwoorden. Eerst wordt de experimentele opzet besproken, gevolgd door een beschrijving van de data opzet. Tot slot worden de gebruikte evaluatiemetrieken toegelicht.

4.1 Experimentele opzet

In dit onderzoek werden vijftig afzonderlijke experimenten uitgevoerd, waarbij telkens één unieke synthetisch gegenereerde *event log* uit [11] als input voor het TOAD-model gebruikt werd. Elke *event log* bevatte 10.000 *traces* met tussen de 8 en 15 *events* per *trace*. Voorafgaand aan het toepassen van TOAD werden in elke *event log* aan een variërend aantal *traces* kunstmatige temporele afwijkingen toegevoegd. Dat was noodzakelijk omdat de *event logs* oorspronkelijk geen temporele afwijkingen bevatten. Deze datamanipulatie wordt uitgebreid toegelicht in sectie 4.2.

Bij de configuratie van het TOAD-model krijgt de gebruiker de mogelijkheid om de waarden te bepalen voor verschillende parameters die het model beïnvloeden. In dit onderzoek werd er gekozen

om in de vijftig experimenten dezelfde parameterwaarden te gebruiken. Dit onderzoek tracht namelijk de prestaties van het TOAD-model te evalueren en te achterhalen of het TOAD-model in staat is om onderscheid te maken tussen *exceptions* en anomalieën. De invloed van de parameters op die prestaties en de bijhorende parameteroptimalisatie liggen buiten de scope van dit onderzoek.

De eerste modelparameter is MinPts. Die parameter bepaalt het minimumaantal punten dat rond een potentieel kernpunt moet liggen, om te kunnen spreken van een effectief kernpunt in een gebied met hoge dichtheid. In het TOAD-model wordt die parameter gebruikt door de OPTICS clustermethode om de kernafstanden en *reachability*-afstanden te berekenen. In [16], wordt aangegeven dat er geëxperimenteerd werd met verschillende waarden voor de MinPts parameter. Er werd echter geconcludeerd dat MinPts = 10 de beste resultaten opleverde. Er werd voor dit onderzoek, gebaseerd op die conclusie, gekozen om die parameterwaarde over te nemen en MinPts = 10 te hanteren.

De ε-radius vormt de tweede modelparameter. Die parameter bepaalt de maximale straal rond een kernpunt waarbinnen het MinPts aantal buren moet liggen, om te kunnen spreken van een gebied met hoge dichtheid. Het TOAD-model maakt gebruik van OPTICS dat de kern- en *reachability*-afstand bepaalt met MinPts. De modelparameter ε-radius vormt dus slechts een bovengrens voor OPTICS en heeft verder geen invloed op het TOAD-model of de clustering. Ook voor deze parameter werd gekozen om de waarde die gebruikt werd in de [16] over te nemen. In dit onderzoek was ε-radius = 20.

De derde modelparameter is Prominence. De waarde van deze parameter bepaalt hoe ver een dal onder de baseline van de OPTICS-plot moet gaan om als cluster herkend te worden door het TOAD-model. Een hoge waarde van deze parameter zorgt ervoor dat enkel de zeer uitgesproken diepe clusters herkend worden. Een lage waarde voor de Prominence parameter laat toe dat ook kleinere clusters door het TOAD-model gedetecteerd kunnen worden. Opnieuw werd er gekozen om dezelfde waarde te selecteren als in [16] gebruikt werd. Prominence = 0.3 werd als waarde gebruikt in dit onderzoek.

Tot slot zijn er verschillende andere parameters die vast gecodeerd zijn in de TOAD code. Voorbeelden van zulke parameters zijn de *Savitzky-Golay Window* en *Polynomial Degree*. Aangezien die parameters niet expliciet bovenaan de code vermeld staan als parameters die de gebruiker dient aan te passen, werd er voor gekozen die parameters ongewijzigd te laten.

4.2 Data opzet

Voor de experimenten in dit onderzoek werd gebruik gemaakt van vijftig *event logs* uit [11], die synthetisch gegenereerd werden. De eerste stap in het generatieproces bestond uit het aanmaken van honderd sets met telkens drie hiërarchische declaratieve procesmodellen. Declaratieve procesmodellen leggen een set van regels vast waaraan procesuitvoeringen moeten voldoen. Daardoor wordt er ruimte gelaten voor procesflexibiliteit en alternatieve procesuitvoeringen zolang de opgelegde regels niet geschonden worden [14]. De declaratieve procesmodellen uit [11] werden gegenereerd met Declare MoGeS, een tool die automatische generatie en specialisatie van declaratieve procesmodellen mogelijk maakt. De gegenereerde modellen zijn uitgedrukt in de Declare-taal, een *constraint*-gebaseerde modelleertaal gebaseerd op *Linear Temporal Logic over finite traces* (LTLf). Elk model bestaat uit een verzameling van niet-contradictoire regels tussen activiteiten die impliciet het toegelaten gedrag van het proces definiëren. Specialisaties worden bekomen door bestaande regels te verfijnen of te vervangen door strengere

varianten, waardoor het toegelaten gedrag verder beperkt wordt [9].

Het aantal procesregels en het aantal activiteiten in de modellen werd gevarieerd tussen 10 en 26. Binnen één set hebben de drie procesmodellen altijd hetzelfde aantal activiteiten, maar een verschillende graad van striktheid. Het normatieve model is het meest strikte model en bevat expliciet gedefinieerde regels die het gewenste en theoretische procesgedrag beschrijven. Elk procesgedrag dat daar van afwijkt wordt als een procesafwijking beschouwd, ongeacht of die afwijking in de praktijk toelaatbaar is. Het normatieve model wordt veel gebruikt bij *conformance checking* en detecteerd zowel *exceptions* als anomalieën. Het auditor model is een minder strikt model dat een subset van de regels uit het normatieve model bevat, aangevuld met impliciete proceskennis van de auditor. Het model laat bewust bepaalde procesafwijkingen toe die in de praktijk als aanvaardbaar worden beschouwd. Enkel anomalieën worden door het auditor model als procesafwijkingen gedetecteerd. Op die manier fungeert het auditor model als een filter tussen *exceptions* en anomalieën. Het world model is tot slot het meest vrije model en representeert hoe het proces in realiteit verloopt, inclusief alle mogelijke procesafwijkingen ongeacht of die conform de formele regels zijn of niet. Het world model laat dus zowel *exceptions* als anomalieën toe.

Als tweede stap in het generatieproces werden *event logs* gegenereerd met Declare4Py. Dat is een Pythonbibliotheek voor *declarative process mining*, die ondersteuning biedt voor het genereren van *event logs* op basis van de gegenereerde declaratieve procesmodellen met Declare MoGeS [4]. Als basismodel voor de *event logs* werd het world model gekozen. Op basis van dat model werden er *event logs* gegenereerd met 10 000 *traces* die tussen de 8 en 15 *events* bevatten. Met die simulatie werden er 100 *logs* gecreëerd die reëel procesgedrag nabootsen.

De derde stap van het generatieproces bestond uit het uitvoeren van twee *conformance checks* op de 100 gegenereerde *event logs*. In de eerste *conformance check* werden de *event logs* getoetst aan het normatieve model. De *traces* die niet voldeden aan de regels van het normatieve model, werden gelabeld als procesafwijkingen. Vervolgens werd een tweede *conformance check* uitgevoerd waarbij de gelabelde procesafwijkingen uit de eerste *conformance check* getoetst werden aan het auditor-model. De *traces* die wel aan dit model voldeden, werden gelabeld als *exceptions*. De *traces* die ook deze *conformance check* niet haalden, werden gelabeld als anomalieën Door de twee *conformance checks* uit te voeren werden er honderd populaties gecreëerd met gelabelde procesafwijkingen.

Tot slot werd in de vierde stap, de laatste relevante stap voor dit onderzoek, elke gelabelde populatie van procesafwijkingen gebruikt om drie ongebalanceerde datasets te construeren. In elk van die datasets werd het aandeel anomalieën kunstmatig beperkt tot respectievelijk 1%, 2% en 5% van het totale aantal procesafwijkingen. Die keuze weerspiegelt de realiteit waarin het merendeel van de procesafwijkingen *exceptions* zijn en slechts een klein deel werkelijk anomalieën zijn. Op basis van de 100 oorspronkelijke populaties met gelabelde afwijkingen werden er zo 300 ongebalanceerde datasets gegenereerd.

Voor de experimenten in dit onderzoek werden vijftig *event logs* geselecteerd uit de set van honderd synthetisch gegenereerde *logs*. Die selectie gebeurde via *random sampling* om selectie *bias* te vermijden. Voor elk van de geselecteerde *event logs* werd de bijhorende gelabelde populatie van procesafwijkingen met 5% anomalieën gekozen. Het aantal van vijftig *logs* bood een goed evenwicht tussen datavolume en rekentijd, wat essentieel was voor de praktische uitvoerbaarheid van de experimenten.

Omdat de geselecteerde *event logs* geen gelabelde procesafwijkingen bevatten, werd er gekozen om zelf binaire labels toe te voegen aan alle *traces* in de *logs*. De labels werden afgeleid uit de bijhorende populaties met procesafwijkingen. Het doel van de labeling was om een controlemechanisme in te bouwen waarmee de prestaties van het TOAD-model geëvalueerd konden worden. De labels werden gebruikt om de gevormde clusters na het toepassen van TOAD te toetsen aan de *ground truth*. Het TOAD-model kreeg geen toegang tot de labels voor de clustering. Er werden twee labels toegevoegd aan elke *trace*: Deviation en Anomaly. *Traces* die niet voorkwamen in de populatie met procesafwijkingen kregen voor beide labels de waarde 0 en worden als niet afwijkende *traces* beschouwd. *Traces* die wel voorkwamen in de populatie procesafwijkingen, maar daarin niet gelabeld waren als anomalie, kregen voor het Deviation label de waarde 1 en voor het Anomaly label de waarde 0. Die traces worden beschouwd als de *exceptions*. Tot slot kregen de *traces* die als anomalie gelabeld waren in de populatie procesafwijkingen voor beide labels de waarde 1. Die *traces* vormden de kleine groep anomalieën. Door twee labels toe te voegen, kon er na het toepassen van TOAD op de *event log* nagegaan worden of TOAD in staat is om naast het detecteren van procesafwijkingen ook een onderscheid te maken tussen *exceptions* en anomalieën.

Hoewel de geselecteerde *event logs* verrijkt werden met labels die aangeven welke *traces* als *exceptions* en anomalieën beschouwd worden, is het belangrijk te benadrukken dat die labels gebaseerd zijn op afwijkingen tegenover de declaratieve procesmodellen. Het TOAD-model is echter specifiek ontworpen om collectieve temporele procesafwijkingen te detecteren. Zonder verdere aanpassingen aan de gelabelde procesafwijkingen zou het TOAD-model die *traces* niet als procesafwijkingen detecteren, aangezien ze geen temporele afwijkingen bevatten. Om ervoor te zorgen dat TOAD in staat is om de gelabelde procesafwijkingen effectief te detecteren, werd er gekozen een kunstmatige temporele afwijking toe te voegen aan die *traces*. Er werd bewust gekozen om een vertraging toe te voegen omdat dat type afwijking doorgaans als problematischer wordt ervaren dan een versnelling.

Er werd gekozen om in elke geselecteerde *event log* de uitvoering van één specifieke activiteit kunstmatig te vertragen. Die doelactiviteit werd per *log* bepaald op basis van frequentie. De activiteit die in de meeste unieke *traces* voorkwam, werd geselecteerd. Vervolgens werden voor elke *event log* alle tijdsintervallen berekend tussen de doelactiviteit en de direct voorafgaande activiteit. Op basis van die intervallen werd zowel de gemiddelde duur als de standaardafwijking berekend. De standaardafwijking diende als basis voor het bepalen van de grootte van de kunstmatige vertraging. Er werd bewust een onderscheid gemaakt tussen de vertraging voor *exceptions* en anomalieën, zodat onderzocht kon worden of TOAD tijdens het clusteren in staat is om die twee types afwijkingen van elkaar te onderscheiden. *Traces* die als *exception* gelabeld waren (Deviation = 1, Anomaly = 0) kregen een vertraging van één keer de standaardafwijking toegewezen. Anomalieën (Deviation = 1, Anomaly = 1) kregen een grotere vertraging, namelijk drie keer de standaardafwijking. De vertraging werd toegepast door de*timestamp* van de doelactiviteit én van alle daaropvolgende activiteiten in de betreffende *trace* met het berekende aantal seconden te verschuiven. Binnen één *event log* kregen alle *exceptions* dezelfde vertraging en alle anomalieën eveneens een uniforme, maar grotere vertraging.

4.3 Evaluatiemetrieken

De prestaties van het TOAD-model werden in dit onderzoek beoordeeld aan de hand van verschillende evaluatiemetrieken. De eerste evaluatiemetriek die gemeten werd, is de verdeling van het aantal gecreëerde niet-*noise* clusters. Die metriek biedt inzicht in de granulariteit van de clustering. Indien er een extreem laag aantal clusters gevormd zou worden, kan dat een teken zijn van over-aggregatie van de *traces* waardoor er geen duidelijke clusterstructuren gevonden konden worden in de data. Een te groot aantal clusters kan echter wijzen op *overfitting* van het TOAD-model, waarbij er in een extreem geval voor elke *trace* een aparte cluster gevormd zou kunnen worden [17].

Precision is de tweede evaluatiemetriek die voor zowel de *exceptions*, de anomalieën als de gehele groep procesafwijkingen berekend werd. De set procesafwijkingen bestaat uit de *exceptions* en de anomalieën samen. De metriek biedt inzicht in de mate waarin de *traces* die door het TOAD-model in een niet-*noise* cluster geplaatst werden ook effectief als procesafwijking gelabeld waren. Er worden twee variabelen gebruikt om de *Precision* te berekenen. De eerste variabele staat voor de *True Positive traces* (TP). Dat zijn de *traces* die als procesafwijking gelabeld zijn en ook door het TOAD-model in een niet-*noise* cluster geplaatst werden. De tweede variabele staat voor de *False Positive traces* (FP). Dat zijn de *traces* die niet als procesafwijking gelabeld zijn, maar toch door het TOAD-model in een niet-*noise* cluster geplaatst werden. De *Precision* wordt berekend door het aantal TP-*traces* te delen door de som van de TP- en FP-*traces* en resulteert in een waarde tussen 0.0 en 1.0. Indien de *Precision* laag is, bestaan de gevormde niet-*noise* clusters voornamelijk uit *traces* die niet als procesafwijking gelabeld werden. Indien de *Precision* hoog is, is het TOAD-model er in geslaagd om de clusters voornamelijk te vormen uit afwijkende *traces* [18].

De derde evaluatiemetriek die voor de *exceptions*, de anomalieën en de gehele groep procesafwijkingen berekend werd, is *Recall*. Die metriek geeft inzicht in de mate waarin afwijkende *traces* door het TOAD-model herkend werden en in een niet-*noise* cluster geplaatst werden. Ook *Recall* wordt berekend aan de hand van twee variabelen. De eerste variabele is opnieuw de *True Positive traces*. De tweede variabele bestaat uit de *False Negative traces*. Dat zijn de *traces* die als procesafwijking gelabeld zijn, maar toch in de *noise* cluster geplaatst werden en dus niet als procesafwijking gedetecteerd werden. De *Recall* wordt berekend door het aantal TP-*traces* te delen door de som van de TP- en FN-*traces* en resulteert ook in een waarde tussen 0.0 en 1.0. Indien de *Recall* laag is, werden er weinig afwijkende *traces* gevonden door TOAD. Indien *Recall* hoog is, is het model er in geslaagd om een groot deel van de procesafwijkingen te detecteren [18].

De F1-score is de vierde evaluatiemetriek en wordt berekend aan de hand van de *Precision* en de *Recall*. Concreet wordt de F1-score berekend door het product van de *Precision* en *Recall* te vermenigvuldigen met twee en dat resultaat te delen door de som van de *Precision* en de *Recall*. Die bewerking resulteert in een samengevoegd gemiddelde met een waarde tussen 0.0 en 1.0. Indien de F1-score hoog is, moeten de *Precision* en *Recall* ook hoog zijn en kan geconcludeerd worden dat het TOAD-model redelijk goed gepresteerd heeft. Indien de F1-score laag is, betekent dat dat beide metrieken laag waren en het TOAD-model dus minder goed gepresteerd heeft [18].

Tot slot werd de cluster-*purity* berekend om de homogeniteit van de gevormde niet-*noise* clusters te

beoordelen. Dat werd zowel in relatie tot de *exceptions*, de anomalieën als de gehele set procesafwijkingen berekend. De cluster-*purity* geeft aan in welke mate er homogeniteit is binnen een niet-*noise* cluster op vlak van het type *trace*. De gemiddelde cluster-*purity* geeft het gemiddelde weer voor de verzameling van clusters die gevormd werden op basis van een *event log*. Indien de gemiddelde cluster-*purity* hoog is, wil dat zeggen dat er vrij homogene clusters gevormd werden. Als dat gemiddelde laag is, werden er voornamelijk heterogene clusters gevormd. Daarnaast wordt ook de standaardafwijking van de gemiddelde cluster-*purity* berekend. Als die standaardafwijking laag is, wil dat zeggen dat de clusters gelijkend van aard zijn. Indien er een hoge standaardafwijking is, wil dat zeggen dat er veel verschil is in cluster-*purity* [6].

5 Resultaten

In de volgende secties worden de resultaten besproken voor elk van de gehanteerde evaluatiemetrieken. Deze resultaten zijn het gevolg van een reeks van vijftig experimenten, waarbij in elk experiment het TOAD-model werd toegepast op één afzonderlijke event log. Op die manier werd de prestatie van het model op consistente en herhaalbare wijze geëvalueerd over vijftig unieke cases.

5.1 Gecreëerde niet-*noise* clusters

Tabel 1 geeft een overzicht van het aantal gecreëerde niet-*noise* clusters over de vijftig uitgevoerde experimenten. Het kleinste aantal niet-*noise* clusters dat door het TOAD-model werd gevormd bedraagt 32. Gemiddeld werden er 93,96 niet-*noise* clusters gegenereerd, met een standaardafwijking van 24,47. Het hoogste aantal clusters dat in een enkel experiment werd vastgesteld, bedraagt 138. Deze spreiding wijst op een aanzienlijke variabiliteit in het clusteringsgedrag van TOAD over de verschillende *event logs*.

Minimum	Gemiddelde	Standaardafwijking	Maximum
32	93.96	24.47	138

Tabel 1: Beschrijvende statistieken van het aantal gecreëerde niet-*noise* clusters

5.2 *Precision*

In tabel 2 worden de resultaten voor de *precision*-metriek getoond, opgesplitst per type procesafwijking. Voor de *exceptions* varieerde de gemeten *precision* tussen 0,0 en 0,8729. De gemiddelde waarde bedroeg 0,4384 met een standaardafwijking van 0,2770. De *precision*-scores voor anomalieën lagen beduidend lager. Ook hier was de minimale waarde gelijk aan 0,0, maar de maximale *precision* bedroeg slechts 0,0864. Het gemiddelde kwam uit op 0,0257, met een standaardafwijking van 0,0198. Voor de volledige set procesafwijkingen, zowel *exceptions* als anomalieën, lag de *precision* tussen 0,0 en 0,9346. De gemiddelde waarde voor deze categorie bedroeg 0,4641 met een standaardafwijking van 0,2940. De resultaten wijzen op aanzienlijke verschillen in *precision*, afhankelijk van het type procesafwijking.

	Minimum	Gemiddelde	Standaardafwijking	Maximum
Exceptions	0.0	0.4384	0.2770	0.8729
Anomalieën	0.0	0.0257	0.0198	0.0864
Procesafwijkingen	0.0	0.4641	0.2940	0.9346

Tabel 2: Beschrijvende statistieken van de *precision*-scores

5.3 *Recall*

Tabel 3 presenteert de resultaten voor de *recall*-metriek, afzonderlijk berekend voor *exceptions*, anomalieën en de volledige set procesafwijkingen. Voor de *exceptions* werd een gemiddelde *recall* van 0,1418 vastgesteld, met een standaardafwijking van 0,0595. De hoogste *recall*-waarde binnen deze categorie bedroeg 0,2584, terwijl de laagste gelijk was aan 0. Bij de anomalieën varieerde de *recall* tussen 0 en 0,3333. De gemiddelde waarde voor de anomalieën bedroeg 0,1415 met een standaardafwijking van 0,0721. Voor de volledige set procesafwijkingen werd een gemiddelde *recall* van 0,1420 gemeten. De minimumwaarde bedroeg hier 0,0 en het maximum 0,2546 en de standaardafwijking was 0,0580. Deze resultaten wijzen op vrij lage *recall*-waarden voor alle types procesafwijkingen, wat impliceert dat het TOAD-model slechts een beperkt aandeel van de effectief aanwezige afwijkingen correct identificeert.

	Minimum	Gemiddelde	Standaardafwijking	Maximum
Exceptions	0.0	0.1418	0.0595	0.2584
Anomalieën	0.0	0.1415	0.0721	0.3333
Procesafwijkingen	0.0	0.1420	0.0580	0.2546

Tabel 3: Beschrijvende statistieken van de *recall*-scores

5.4 F1-score

Tabel 4 toont een overzicht van de F1-scores die het TOAD-model behaalde bij het detecteren van respectievelijk *exceptions*, anomalieën en de volledige set procesafwijkingen. Voor de *exceptions* varieerde de F1-score van 0,0 tot maximaal 0,3825. De gemiddelde score bedroeg 0,1846 met een standaardafwijking van 0,0991. Voor de anomalieën bleven de prestaties beduidend lager. De F1-score varieerde tussen 0,0 en 0,1049 met een gemiddelde van 0,0410 en een standaardafwijking van 0,0286. Wanneer beide afwijkingstypes samen worden genomen, werd een gemiddelde F1-score van 0,1874 geregistreerd met een minimumwaarde van 0,0 en een maximum van 0,3818. De standaardafwijking voor de procesafwijkingen bedroeg 0,0989. De cijfers bevestigen het algemene beeld dat TOAD slechts in beperkte mate in staat is om afwijkende *traces* accuraat te identificeren tijdens de clustering.

	Minimum	Gemiddelde	Standaardafwijking	Maximum
Exceptions	0.0	0.1846	0.0991	0.3825
Anomalieën	0.0	0.0410	0.0286	0.1049
Procesafwijkingen	0.0	0.1874	0.0989	0.3818

Tabel 4: Beschrijvende statistieken van de F1-scores

5.5 Cluster-*Purity*

In tabel 5 wordt een overzicht gegeven van de behaalde cluster-*purity* per afwijkingstype. Voor de *exceptions* varieerde de cluster-*purity* tussen 0,0 en 0,8790 met een gemiddelde waarde van 0,4332 en een standaardafwijking van 0,2732. Dat wijst op een gematigd niveau van homogeniteit binnen de clusters waarin *exceptions* werden ondergebracht. De resultaten voor de anomalieën tonen echter een aanzienlijk lagere cluster-*purity*. De minimumwaarde bedroeg ook hier 0,0, terwijl het maximum beperkt bleef tot 0,0783. De gemiddelde waarde bedroeg slechts 0,0258 met een standaardafwijking van 0,0205 wat aangeeft dat anomalieën zelden in homogene clusters worden gegroepeerd. Wanneer zowel *exceptions* als anomalieën gezamenlijk worden beschouwd als procesafwijkingen, werd een gemiddelde cluster-*purity* van 0,4590 vastgesteld. De standaardafwijking in deze categorie bedroeg 0,2905 met een minimale waarde van 0,0 en een maximale waarde van 0,9403. Die spreiding suggereert een hoge variabiliteit in de homogeniteit van de gevormde clusters over de verschillende *event logs* heen.

	Minimum	Gemiddelde	Standaardafwijking	Maximum
Exceptions	0.0	0.4332	0.2732	0.8790
Anomalieën	0.0	0.0258	0.0205	0.0783
Procesafwijkingen	0.0	0.4590	0.2905	0.9403

Tabel 5: Verdeling van de cluster-*purity* voor *Exceptions* & Anomalieën en procesafwijkingen

6 Discussie & Beperkingen

Om te evalueren hoe goed het TOAD-model collectieve temporele procesafwijkingen kan detecteren en of het model daarbij onderscheid kan maken tussen *exceptions* en anomalieën, werden vijftig afzonderlijke experimenten uitgevoerd. In elk experiment werd TOAD toegepast op een unieke synthetische gegenereerde *event log*. De prestaties van het model werden beoordeeld aan de hand van vijf evaluatiemetrieken: het aantal gevormde niet-*noise* clusters, *precision*, *recall*, F1-score en cluster-*purity*.

In een ideaal scenario zou TOAD per *event log* exact drie clusters vormen: één cluster met anomalieën, één met *exceptions* en één overkoepelende noise-cluster voor de niet-afwijkende *traces*. De realiteit blijkt echter complexer. Uit tabel 1 blijkt dat het aantal niet-*noise* clusters varieerde van 32 tot 138, met een gemiddelde van 93,96. Dat is ruim 90 clusters boven het theoretische optimale aantal en wijst op fragmentatie van de data. TOAD groepeert de gedetecteerde afwijkende *traces* dus in veel kleine clusters. Er is echter geen sprake van *overfitting* waarbij elke *trace* zijn eigen cluster zou krijgen. Het hoge aantal clusters suggereert vooral dat TOAD kleine variaties in de vertragingen lijkt te detecteren en daarom veel clusters aanmaakt. Dat is opvallend, aangezien binnen elke *event log* alle *exceptions* en anomalieën telkens een uniforme vertraging kregen.

Op basis van de *precision*-scores toont TOAD een duidelijk asymmetrisch prestatiebeeld voor de *exceptions* en anomalieën. *Precision* meet het aandeel correct geïdentificeerde afwijkende *traces* ten opzichte van het totaal aantal *traces* dat TOAD als afwijkend classificeert. Voor *exceptions* varieert de *precision* van 0,0 tot 0,8729 met een gemiddelde van 0,4384. Voor anomalieën ligt dat veel lager. De gemiddelde *precision*-score is slechts 0,0257 met een maximum van 0,0864. Wanneer beide types pro-

cesafwijkingen samen worden beschouwd, stijgt de gemiddelde *precision* naar 0,4641. Uit die cijfers kan afgeleid worden dat TOAD in sommige experimenten in staat was om *exceptions* redelijk goed te detecteren. De piekwaarde van 0,8729 toont aan dat de vertraging bij *exceptions* in die gevallen voldoende onderscheidend was om duidelijke clusters te vormen. Voor anomalieën geldt het tegenovergestelde. De toegevoegde vertragingen lijken te weinig onderscheidend. Slechts één à twee procent van de door TOAD als anomalie gelabelde *traces* is effectief een echte anomalie. Zelfs in het beste geval blijft de zuiverheid onder de 9 procent, wat aantoont dat anomalieën zelden als afzonderlijke groep worden herkend. In de gecombineerde resultaten voor de gehele set van procesafwijkingen wordt de slechte detectie van anomalieën deels gecompenseerd door de grotere aantallen correct geïdentificeerde *exceptions*, waardoor de *precision* stijgt. De *precision*-scores tonen dus dat TOAD duidelijk beter presteert in het detecteren van *exceptions* dan in het detecteren van anomalieën. De algemene detectie van procesafwijkingen is echter ondermaats.

De *recall*-scores bevestigen dat TOAD er in de uitgevoerde experimenten niet in geslaagd is om afwijkende *traces* systematisch te detecteren. Voor *exceptions* bedraagt de gemiddelde *recall* 0,1418 met een maximum van 0,2584. Voor anomalieën ligt de gemiddelde *recall* nagenoeg gelijk op 0,1415 met een iets hoger maximum van 0,3333. De gecombineerde *recall* over alle procesafwijkingen bedraagt 0,1420. Dat betekent dat TOAD in de vijftig experimenten gemiddeld slechts veertien procent van de afwijkende *traces* herkent als procesafwijkingen. Opvallend is dat de *recall*-waarden voor *exceptions* en anomalieën zo goed als gelijk zijn, ondanks dat *exceptions* slechts één keer de standaardafwijking als vertraging kregen, terwijl anomalieën drie keer zoveel vertraging kregen. Dat suggereert dat de z-scores van beide afwijkingstypes vaak onder de detectiedrempel van TOAD bleven. Als gevolg daarvan belanden veel afwijkende *traces*, zowel *exceptions* als anomalieën, in de *noise*-cluster en worden ze niet als afwijking gedetecteerd.

Omdat *precision* en *recall* elk slechts een deelaspect van de modelprestaties weergeven, is het zinvol om ook naar de F1-scores te kijken. Die metriek combineert beide metingen in één getal en drukt zo de balans uit tussen zuiverheid en volledigheid. Voor *exceptions* bedraagt de gemiddelde F1-score 0,1846 met een maximum van 0,3825. Bij anomalieën ligt de gemiddelde F1-score veel lager. Het gemiddelde is 0,0410 met een maximum van 0,1049. Voor de totale groep procesafwijkingen is de gemiddelde F1-score 0,1874. De resultaten tonen aan dat TOAD, ondanks het feit dat er soms wel relatief zuivere clusters van *exceptions* gevormd worden, er niet in slaagt om tegelijk veel procesafwijkingen correct te detecteren en het aantal foutieve detecties laag te houden. Bij anomalieën is de F1-score zo laag dat TOAD in de praktijk weinig bruikbaar lijkt voor het onderscheiden van dat type procesafwijkingen. Wanneer anomalieën in een cluster terechtkomen, worden ze vrijwel altijd omringd door foutief geclassificeerde *traces*, waardoor de zuiverheid van die clusters en dus ook de F1-score laag blijft.

Ook de cluster-purity bevestigt dat TOAD moeite heeft met het vormen van kwalitatieve clusters. Deze metriek meet de homogeniteit binnen een niet-*noise* cluster, oftewel het aandeel *traces* in de cluster die effectief een procesafwijking zijn. Voor *exceptions* varieert de cluster-*purity* van 0,0 tot 0,8790 met een gemiddelde van 0,4332. Voor anomalieën zijn de scores aanzienlijk lager met een maximum van 0,0783 en een gemiddelde van slechts 0,0258. Voor de volledige groep procesafwijkingen bedraagt de gemiddelde cluster-*purity* 0,4590. De resultaten tonen aan dat TOAD er doorgaans niet in slaagt om

clusters te vormen die homogeen zijn op vlak van het type procesafwijking. TOAD slaagt er dus niet in om clusters te vormen die uit één type procesafwijking bestaan. De hoge standaardafwijking suggereert bovendien dat de prestaties sterk variëren per experiment. In sommige gevallen wist TOAD wel een relatief zuivere cluster te vormen, maar meestal bestaan de clusters uit een mengeling van *exceptions*, anomalieën en niet afwijkende *traces*. Ook in de algemene detectie van procesafwijkingen blijkt TOAD beperkt succesvol. Minder dan de helft van de *traces* in een niet-*noise* cluster is effectief een procesafwijking.

Gezamenlijk laten de vijf evaluatiemetrieken zien dat TOAD in de uitgevoerde experimenten ondermaats presteerde. De fragmentatie van de data laat zien dat TOAD veel clusters creëert, maar dat die clustering niet leidt tot betere detectie van anomalieën. Hoewel TOAD in sommige *logs* redelijk zuivere clusters kan vormen, detecteert het model gemiddeld slechts 14% van de *exceptions* en 14% van de anomalieën. Ten slotte tonen de cluster-purity aan dat de clusters die TOAD vormt vaak heterogeen zijn en dat anomalieën vrijwel altijd verdwijnen tussen de *exceptions* en de niet afwijkende *traces*. Toad slaagt er dus niet in om een consequent onderscheid te maken tussen *exceptions* en anomalieën. Wanneer gekeken wordt naar de prestaties van het TOAD model op vlak van de detectie van de gehele set van procesafwijkingen, zijn ook die resulaten ondermaats.

Hoewel dit onderzoek waardevolle inzichten biedt in de prestaties van het TOAD-model bij het detecteren van collectieve temporele procesafwijkingen en het onderscheiden van *exceptions* en anomalieën zijn er enkele beperkingen die de resultaten en de generaliseerbaarheid ervan kunnen beïnvloeden.

Een eerste belangrijke beperking betreft de grootte van de kunstmatig toegevoegde vertragingen. In elk experiment werd aan de afwijkende *traces* een vertraging toegevoegd om een detecteerbare temporele afwijking te simuleren. Die vertraging werd proportioneel berekend op basis van de standaardafwijking van de tijdsintervallen tussen een geselecteerde doelactiviteit, die vertraagd zou worden en de voorgaande activiteit. Concreet werd gekozen voor een vertraging van één keer de standaardafwijking voor *exceptions* en drie keer de standaardafwijking voor anomalieën. Hoewel deze aanpak rationeel onderbouwd is, kan het zijn dat de gekozen schaal van de vertragingen te beperkt was. Indien de toegevoegde afwijkingen te klein zijn in verhouding tot de variatie in normaal procesgedrag, blijven ze mogelijk onder de detectiedrempel van het TOAD-model. Dat kan geleid hebben tot lage *recall*- en *precision*-scores, zoals eerder besproken.

Een tweede beperking betreft het gebruik van vaste parameterinstellingen. In dit onderzoek werden de standaardinstellingen uit de originele TOAD-publicatie [16] overgenomen, zonder bijkomende optimalisatie. Er werd besloten dat de invloed van de parameters buiten de *scope* van dit onderzoek valt, maar dat sluit niet uit dat alternatieve parametercombinaties betere prestaties hadden kunnen opleveren op de specifieke *event logs* die in dit onderzoek gebruikt werden.

Een derde beperking heeft betrekking op de aard van de gebruikte temporele afwijkingen. De vertragingen die aan de afwijkende *traces* toegevoegd werden, zijn artificieel gegenereerd. Hoewel synthetische manipulaties vaak gebruikt worden in experimenteel onderzoek, om controle te behouden over de datakarakteristieken, vertegenwoordigen ze niet noodzakelijk realistische procesverstoringen zoals die zich in echte bedrijfsomgevingen voordoen. Het is mogelijk dat echte anomalieën in *event logs* subtieler of complexer zijn dan een uniforme vertraging, wat kan verklaren waarom TOAD in eerder onderzoek

betere resultaten behaalde dan in deze experimentele opzet.

Daarnaast is ook het volume en de verhouding van afwijkende *traces* in de gebruikte *event logs* een aandachtspunt. In dit onderzoek werden vijftig synthetisch gegenereerde logs gebruikt, elk bestaande uit 10.000 *traces*. De gelabelde procesafwijkingen bevatten respectievelijk 5% anomalieën, met het totaal aantal afwijkende *traces* variërend van enkele tientallen tot enkele duizenden. Wanneer afwijkende *traces* in grote aantallen aanwezig zijn in een dataset, kan dat ertoe leiden dat die afwijkingen door een detectiemodel als normaal gedrag geïnterpreteerd worden. Omgekeerd geldt dat in datasets met veel normale *traces* de afwijkingen sterker kunnen opvallen, omdat het normale gedrag beter gedefinieerd is. De verhouding tussen normaal en afwijkend gedrag beïnvloedt dus mogelijks de detectiecapaciteit van TOAD.

Tot slot is er een beperking in de manier waarop TOAD omgaat met ontbrekende informatie. In veel *event logs* zijn niet alle mogelijke activiteitenparen aanwezig in elke *trace*. Daardoor ontbreken er voor bepaalde paren inter-event-tijden die nodig zijn voor de berekening van z-scores. In de huidige versie van TOAD wordt dat opgelost door een standaardwaarde van nul toe te kennen aan de ontbrekende z-scores. Die nulwaarde impliceert dat het gedrag normaal is, terwijl er in werkelijkheid geen informatie beschikbaar is. Dat kan leiden tot een kunstmatig hoge dichtheid van vectorrepresentaties rond de oorsprong in de vectorruimte, wat het clustergedrag van het model kan beïnvloeden.

7 Conclusie

Procesafwijkingen komen voor bij het merendeel van de processen. Die procesafwijkingen kunnen in uitzonderlijke gevallen indicaties zijn van fraude, defecte machines, cyberaanvallen of het systematisch verkeerd uitvoeren van bepaalde procesactiviteiten. Wanneer een procesafwijking het gevolg is van zo een ernstige verstoring van het proces, wordt die procesafwijking een anomalie genoemd. Die anomalieën dienen verder onderzocht te worden om de oorzaak aan te kunnen pakken. Procesafwijkingen die niet voortkomen uit ernstige oorzaken worden *exceptions* genoemd. Die procesafwijkingen moeten niet verder onderzocht worden. Desondanks is het essentieel om alle procesafwijkingen te detecteren. Het niet detecteren van ernstige anomalieën kan ernstige gevolgen hebben.

In dit onderzoek werd onderzocht in welke mate het TOAD-model in staat is om collectieve temporele procesafwijkingen te detecteren binnen *event logs* en werd voor het eerst onderzocht of het model een onderscheid kan maken tussen *exceptions* en anomalieën. Collectieve temporele procesafwijkingen zijn groepen van procesafwijkingen waarbij een gelijkaardige versnelling of vertraging van de procesuitvoering plaatsvond. Door te focussen op dit type procesafwijkingen hanteert het TOAD-model een nieuwe en unieke invalshoek voor anomaliedetectie. Aan de hand van vijftig experimenten met synthetisch gegenereerde *event logs*, die elk een variërend aantal procesafwijkingen bevatten, werd het TOAD-model in dit onderzoek geëvalueerd op basis van vijf metrieken: het aantal gevormde niet-*noise* clusters, *precision*, *recall*, F1-score en cluster-*purity*.

De resultaten van dit onderzoek tonen aan dat het TOAD-model in de onderzochte context ondermaats presteert in het detecteren van procesafwijkingen. De *precision*- en *recall*-scores zijn laag, wat betekent dat slechts een beperkt deel van de afwijkende *traces* correct gedetecteerd wordt door TOAD,

terwijl er ook veel foutieve detecties plaatsvinden. Vooral bij anomalieën zijn de resultaten bijzonder laag, wat suggereert dat TOAD moeite heeft met het herkennen van zeldzame afwijkingen. De cluster-*purity* toont aan dat de gevormde clusters vaak heterogeen zijn en dus bestaan uit een mix van normale en afwijkende *traces*. Daarnaast blijkt uit de resultaten dat TOAD er niet in slaagt om een structureel onderscheid te maken tussen *exceptions* en anomalieën, hoewel in de data bewust een verschil in mate van vertraging werd aangebracht om dat onderscheid te faciliteren. Dat suggereert dat het model in zijn huidige vorm onvoldoende gevoelig is om dat onderscheid op basis van temporele afwijkingen waar te nemen.

Hoewel deze bevindingen waardevolle inzichten opleveren over de beperkingen van TOAD, zijn er ook verschillende interessante pistes voor toekomstig onderzoek. Zo kan de invloed van de modelparameters verder worden onderzocht, aangezien in dit onderzoek enkel de aanbevolen parameterinstellingen uit de originele TOAD-publicatie gebruikt werden. Het is mogelijk dat alternatieve parametercombinaties betere prestaties van het TOAD-model opleveren. Daarnaast zou het zinvol zijn om na te gaan hoe gevoelig het model is voor de grootte van de temporele vertragingen en of grotere afwijkingen ook effectief duidelijker herkend worden.

Verder kan er geëxperimenteerd worden met complexere vormen van procesafwijkingen, waarbij niet slechts één activiteit vertraagd wordt, maar meerdere afwijkingen gelijktijdig voorkomen. Dit sluit beter aan bij realistische scenario's waarin procesafwijkingen vaak niet tot één geïsoleerde gebeurtenis beperkt blijven. Daarnaast is het waardevol om TOAD te vergelijken met andere technieken voor anomaliedetectie, zoals *autoencoders*, *isolation forests*, *active learning*-benaderingen of andere detectiemethoden. Hoewel TOAD een *unsupervised* model is en dus moeilijk direct vergeleken kan worden met *supervised* methodes, kunnen gelabelde datasets toch toelaten om de prestaties van verschillende technieken objectief te beoordelen.

Tot slot vormt ook de huidige omgang van TOAD met ontbrekende informatie een mogelijk verbeterpunt. In de huidige implementatie krijgen activiteitenparen waarvoor geen doorlooptijd beschikbaar is een z-score van nul toegewezen, wat onterecht suggereert dat er sprake is van normaal gedrag. Alternatieve strategieën voor het omgaan met *missing values*, kunnen onderzocht worden om de vertekening van de vectorruimte te beperken.

De besproken toekomstige onderzoekspistes tonen aan dat er nog veel ruimte is voor verbetering en verdere validatie van TOAD. De methode biedt een interessante en unieke invalshoek om met enkel temporele informatie anomalieën te detecteren, maar in de huidige vorm blijkt de robuustheid en nauwkeurigheid ervan beperkt. Verdere verfijning is nodig om TOAD toepasbaar te maken in realistische *process mining*-omgevingen.

Referenties

[1] Massimiliano De Leoni, Suriadi Suriadi, Arthur H. M. Ter Hofstede, and Wil M. P. Van Der Aalst. Turning event logs into process movies: animating what has really happened. *Software & Systems Modeling*, 15(3):707–732, July 2016. ISSN 1619-1366, 1619-1374. doi: 10.1007/s10270-014-0432-2. URL http://link.springer.com/10.1007/s10270-014-0432-2.

[2] Johannes De Smedt and Pnina Soffer, editors. *Process Mining Workshops: ICPM 2023 International Workshops, Rome, Italy, October 23–27, 2023, Revised Selected Papers*, volume 503 of *Lecture Notes in Business Information Processing*. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-56106-1 978-3-031-56107-8. doi: 10.1007/978-3-031-56107-8. URL https://link.springer.com/10.1007/978-3-031-56107-8.

[3] Benoît Depaire, Jo Swinnen, Mieke Jans, and Koen Vanhoof. A Process Deviation Analysis Framework. In Wil Van Der Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, Clemens Szyperski, Marcello La Rosa, and Pnina Soffer, editors, *Business Process Management Workshops*, volume 132, pages 701–706. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-36284-2 978-3-642-36285-9. doi: 10.1007/978-3-642-36285-9_69. URL http://link.springer.com/10.1007/978-3-642-36285-9_69. Series Title: Lecture Notes in Business Information Processing.

[4] Ivan Donadello and Aladdin Shikhizada. Declare4Py: A Python Library for Declarative Process Mining.

[5] Nebrase Elmrabit, Feixiang Zhou, Fengyin Li, and Huiyu Zhou. Evaluation of Machine Learning Algorithms for Anomaly Detection. In *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–8, Dublin, Ireland, June 2020. IEEE. ISBN 978-1-72816-428-1. doi: 10.1109/CyberSecurity49315.2020.9138871. URL https://ieeexplore.ieee.org/document/9138871/.

[6] Peter Grabusts. Evaluation of Clustering Results in the Aspect of Information Theory. In *2020 61st International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, pages 1–5, Riga, Latvia, October 2020. IEEE. ISBN 978-1-72819-105-8. doi: 10.1109/ITMS51158.2020.9259227. URL https://ieeexplore.ieee.org/document/9259227/.

[7] Harjinder Kaur, Gurpreet Singh, and Jaspreet Minhas. A Review of Machine Learning based Anomaly Detection Techniques. *International Journal of Computer Applications Technology and Research*, 2(2):185–187, March 2013. ISSN 23198656. doi: 10.7753/IJCATR0202.1020. URL http://www.ijcat.com/archives/volume2/issue2/ijcatr02021020.pdf.

[8] Jonghyeon Ko and Marco Comuzzi. A Systematic Review of Anomaly Detection for Business Process Event Logs. *Business & Information Systems Engineering*, 65(4):441–462, Au-

gust 2023. ISSN 2363-7005, 1867-0202. doi: 10.1007/s12599-023-00794-y. URL https://link.springer.com/10.1007/s12599-023-00794-y.

[9] Manal Laghmouch. Declare MoGeS: Model Generator and Specializer.

[10] Manal Laghmouch, Mieke Jans, and Benoit Depaire. Classifying process deviations with weak supervision. In *2020 2nd International Conference on Process Mining (ICPM)*, pages 89–96, Padua, Italy, October 2020. IEEE. ISBN 978-1-72819-832-3. doi: 10.1109/ICPM49681.2020.00023. URL https://ieeexplore.ieee.org/document/9229959/.

[11] Manal Laghmouch, Benoît Depaire, and Mieke Jans. Towards Full Population Testing in Auditing: How Many Process Deviations Should Be Labeled? In *2024 6th International Conference on Process Mining (ICPM)*, pages 49–56, Kgs. Lyngby, Denmark, October 2024. IEEE. ISBN 9798350365030. doi: 10.1109/ICPM63005.2024.10680672. URL https://ieeexplore.ieee.org/document/10680672/.

[12] Ali Bou Nassif, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. Machine Learning for Anomaly Detection: A Systematic Review. *IEEE Access*, 9:78658–78700, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3083060. URL https://ieeexplore.ieee.org/document/9439459/.

[13] Dan Pelleg and Andrew W Moore. Active Learning for Anomaly and Rare-Category Detection.

[14] Maja Pesic and Helen Schonenberg. DECLARE: Full Support for Loosely-Structured Processes.

[15] Florian Richter. Exploring Anomalies in Time - About Temporal Deviations in Processes.

[16] Florian Richter, Yifeng Lu, Ludwig Zellner, Janina Sontheim, and Thomas Seidl. TOAD: Trace Ordering for Anomaly Detection. In *2020 2nd International Conference on Process Mining (ICPM)*, pages 169–176, Padua, Italy, October 2020. IEEE. ISBN 978-1-72819-832-3. doi: 10.1109/ICPM49681.2020.00033. URL https://ieeexplore.ieee.org/document/9230026/.

[17] Federico Sabbatini and Roberta Calegari. ExACT Explainable Clustering: Unravelling the Intricacies of Cluster Formation.

[18] Radek Silhavy and Petr Silhavy, editors. *Artificial Intelligence Application in Networks and Systems: Proceedings of 12th Computer Science On-line Conference 2023, Volume 3*, volume 724 of *Lecture Notes in Networks and Systems*. Springer International Publishing, Cham, 2023. ISBN 978-3-031-35313-0 978-3-031-35314-7. doi: 10.1007/978-3-031-35314-7. URL https://link.springer.com/10.1007/978-3-031-35314-7.

[19] Akanksha Toshniwal, Kavi Mahesh, and Jayashree R. Overview of Anomaly Detection techniques in Machine Learning. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 808–815, Palladam, India, October 2020. IEEE. ISBN 978-1-72815-464-0. doi: 10.1109/I-SMAC49090.2020.9243329. URL https://ieeexplore.ieee.org/document/9243329/.

[20] Ludwig Zellner. Fusing Rule and Process Mining.