Ein Vergleich aktueller Deep-Learning-Architekturen zur Prognose von Prozessverhalten

Kai Heinrich[1], Patrick Zschech[1], Christian Janiesch[1], Markus Bonin[1]

[1] Technische Universität Dresden
{kai.heinrich,patrick.zschech,christian.janiesch,markus.bonin}
@tu-dresden.de

Abstract. Vor dem Hintergrund einer zunehmenden Datenverfügbarkeit sowie der Motivation von Unternehmen, Geschäftsprozesse kontinuierlich zu verbessern, untersucht der Beitrag verschiedene Deep-Learning-Architekturen, die im Kontext der Prozessprognose angewendet werden können, um dadurch kritische Prozessrisiken früher und besser erkennen zu können. Das Forschungsziel ist dabei die Identifikation und Gegenüberstellung aktueller Architekturkonzepte für die Prognose von Folgeereignissen in laufenden Prozessinstanzen. Dazu werden zunächst Ansätze aus dem Bereich Deep Learning identifiziert, welche ein klares Anwendungspotenzial in der Prozessprognose aufweisen. Anschließend wird ein Testdesign entwickelt, anhand dessen die identifizierten Ansätze evaluiert und systematisch gegenübergestellt werden. Die Ergebnisse zeigen, dass neuartige Deep-Learning-Architekturen konkurrenzfähige und teilweise bessere Prognosequalitäten aufweisen als die bisher in der Literatur verwendeten Ansätze.

Keywords: Process Prediction, Process Mining, Process Analytics, Deep Neural Networks, Deep Learning.

1 Einleitung

Die Analyse von Prozessen auf Basis von Prozess- und Prozesskontextdaten nimmt insbesondere aufgrund der fortschreitenden Digitalisierung an Bedeutung zu [1, 2]. Prozessanalysen (PA) können dabei helfen, einen Einblick in die Effektivität und Effizienz von Prozessen zu geben, um diese für die Steigerung der Prozessqualität zu nutzen [3]. Einen Teil dieser Prozessanalysen stellt die Prognose von Prozessschritten als Klassifikations- oder Regressionsproblem dar [4]. Aktuelle Arbeiten beschäftigen sich dabei konkret mit Techniken für die Prognose von nachfolgenden Prozessaktivitäten während der Laufzeit einer Prozessinstanz, um Risiken und Fehlzustände zu erkennen und Handlungsempfehlung zur Steuerung und Kontrolle der Prozesse im Unternehmen frühzeitig ableiten zu können [5]. Gleichzeitig zeigen Techniken aus dem Bereich des Deep Learning (DL), dass diese klassischen Verfahren des maschinellen Lernens, insbesondere bei großen Datenmengen, überlegen sind und neue Potentiale generieren [6].

15th International Conference on Wirtschaftsinformatik,
March 08-11, 2020, Potsdam, Germany

https://doi.org/10.30844/wi_2020_i1-heinrich

Während zwar vereinzelt DL-Methoden im Bereich der PA genutzt werden, fehlt es derzeit an einer allgemeinen Betrachtung der Übertragungsfähigkeit von DL-Architekturen auf den Bereich der Prozessprognose [7, 8]. Demgegenüber führt die Anwendung von breitgefächerten DL-Ansätzen in anderen Problemfeldern wie z. B. der Bild- und Textverarbeitung bereits zu sehr guten Ergebnissen [9, 10]. Ständige Erweiterungen, insbesondere sequentieller Architekturen wie rekurrenter neuronaler Netzwerke (RNN), versprechen verbesserte Prognosequalität und robustere Modelle [11]. Demgegenüber eignen sich sog. Convolutional Neural Networks (CNN) zur Verarbeitung von gitternetzähnlichen Datenstrukturen, wie sie z. B. bei der visuellen Objekterkennung zum Einsatz kommen [9, 12]. Aktuelle Ansätze im Rahmen der CNN-Architekturen zeigen jedoch auch, dass diese ebenfalls im Kontext sequentieller Daten, wie z. B. bei der Verarbeitung natürlicher Sprache, verwendet werden können [13]. Darüber hinaus werden kontinuierlich neue, zum Teil hybride Netzwerkarchitekturen entwickelt, woraus sich einerseits neue Anwendungsfelder ergeben und andererseits bestehende Anwendungsfelder mit neuartigen Ansätzen adressieren lassen.

Als perspektivische Modelle für die Prognose von Prozessverhalten sind in diesem Zusammenhang insbesondere das *Gated Recurrent Unit Network* (GRU), das *Gated Convolutional Neural Network* (GCNN) und das *Key-Value-Predict Attention Network* (KVP) zu nennen (siehe auch Abschnitt 2). Der vorliegende Beitrag beschäftigt sich mit der Übertragbarkeit dieser aktuellen DL-Architekturen auf den Bereich der Prozessprognose. Es ergibt sich damit folgende Forschungsfrage:

Eignen sich die aktuellen Architekturkonzepte des Deep Learning GRU, GCNN und KVP für Prognose von Folgeereignissen in laufenden Prozessinstanzen?

Der Artikel gliedert sich wie folgt: Zunächst werden theoretische Grundlagen der Prozessprognose und verwandte Arbeiten in Abschnitt 2 diskutiert. Darauf aufbauend wird das Forschungs- und Testdesign in Abschnitt 3 vorgestellt. In Abschnitt 4 werden dann die verschiedenen DL-Architekturen auf ihre Performance hin untersucht, wobei neben der Performance der Algorithmen ohne Vorverarbeitung (Abschnitt 4.1) auch der Einfluss verschiedene Vorverarbeitungsstufen des Testdesigns erfasst wird (Abschnitt 4.2). Abschließend findet eine Diskussion (Abschnitt 5) und Zusammenfassung (Abschnitt 6) der Ergebnisse statt.

2 Stand der Forschung

Durch die Verwendung von Prognoseverfahren im Bereich PA können präventive Maßnahmen getroffen, Handlungsempfehlungen ausgesprochen und Fehler und Risiken in kritischen Prozessphasen identifiziert werden. Risiken, die bei kritischen Prozessinstanzen frühzeitig erkannt werden, können finanziellen Schaden oder negative Prozessverläufe verhindern [14]. Neben der Risiken- und Fehlerreduktion kommt der Optimierung von Prozessstrukturen eine besondere Bedeutung zu [15]. Die Prognose kann dabei als Klassifikations- (z. B. nächster Prozessschritt) oder Regressionsproblem (z. B. Restlaufzeit) modelliert werden. Dabei kommen häufig Verfahren des Machine Learning (ML) zum Einsatz, worunter sich auch DL-Ansätze als eine größere

https://doi.org/10.30844/wi_2020_i1-heinrich

Klasse von Verfahren einordnen lassen [16, 17]. Bezüglich des Stands der Forschung lassen sich nun DL-Modelle unterscheiden, welche (i) bereits für die Prozessprognose verwendet wurden und (b) perspektivische Modelle, welche in der Lage sind, sequentielle Daten zu verarbeiten, aber bisher noch nicht für die Prozessprognose verwendet wurden.

Bereits verwendete Modelle. Existierende DL-Modelle zur Prozessprognose basieren vorrangig auf Feed-Forward- und rekurrenten Netzarchitekturen [6, 17].

Klassische *Feed-Forward-Netze* (FNN) stellen hierbei die einfachste Form von DL-Architekturen dar [18]. Diese werden dazu verwendet, formulierte Klassifikationsprobleme mithilfe von vorgegebenen Trainingsbeispielen und den entsprechenden Klassenzugehörigkeiten zu lernen. [17] nutzen diese Architektur in Kombination mit einem vorgeschalteten neuronalen Netz zur Dimensionsreduktion. Durch die extrahierten Sequenzeigenschaften und angereicherten Ereignismerkmale wird das nächste Prozessereignis prognostiziert. Da klassische Feed-Forward-Netze über keinen Mechanismus zur natürlichen Dimensionsreduktion besitzen, existiert auch eine hybride Variante, indem ein vorverarbeitendes neuronales Netz, ein sog. *Stacked Autoencoder* (SAE), vor dem eigentlichen Prognoseschritt wesentliche Merkmale herausfiltert.

Die *Long-short-term-Memory-Architektur* (LSTM) stellt eine Erweiterung der rekurrenten neuronalen Netz-Architektur dar, bei der unterschiedliche Zustände und Zeitpunkte durch Feedback zu vorhergegangenen Schichten des Netzwerkes berücksichtigt werden. Die klassischen RNN-Architekturen weißen allerdings Probleme im Trainingsprozess auf, was zu einer Vernachlässigung einiger Informationen vom Anfang der Sequenz führen kann. LSTM-Architekturen wirken diesem Problem durch Einführung erweiterter Strukturkomponenten in Form eines Langzeitgedächtnisses mit Vernachlässigungsfunktionen für unwichtige und Betonungsfunktionen für wichtige Bestandteile der Sequenz entgegen [19]. Im Rahmen bisheriger PA-Studien wird diese Architektur-Variante insbesondere wegen der Funktion des Langzeitgedächtnisses von [20, 21] verwendet, um nachfolgende Prozessschritte vorauszusagen. Eine leicht erweiterte Variante des LSTM, welche eine bessere Performance in der Prognose von Folgeereignissen aufweist, ist die Architekturform des *Multiplicative LSTM* (mLSTM) [22].

Perspektivische Modelle. Aktuelle Ansätze zur Prognose von Ereignissen auf Basis sequentieller Daten, welche perspektivisch für die Prozessprognose eingesetzt werden können, sind GRU, GCNN und KVP.

Der Zellstruktur des LSTM-Netzwerks steht die GRU-Architektur gegenüber. Ähnlich wie die LSTM-Architektur versucht das GRU dem Problem des verschwindenden Gradienten entgegenzuwirken. Die GRU ist somit eine neue Art von Zellarchitektur, welche in rekurrenten Netzstrukturen verwendet wird. Im Unterschied zu einer LSTM-Zelle hat diese GRU-Zelle eine vereinfachte Zellarchitektur bei gleichbleibender Performance in manchen Anwendungsbereichen [23]. Die realisierten Gates einer GRU-Zelle bestehen aus dem Update-Gate und dem Reset-Gate. Das

https://doi.org/10.30844/wi_2020_i1-heinrich

Update-Gate entscheidet wie intensiv der Zellzustand aktualisiert wird. Das Reset-Gate kontrolliert dabei, welche Zustände vergessen oder beibehalten werden [24].
Beim GCNN nach [25] wird die typischerweise verwendete rekurrente Netzstruktur durch Gated Temporal Convolutions ersetzt. Im Gegensatz zur LSTM-Architektur besitzen Gated Temporal Convolutions eine begrenzte Kontextbreite, die gesetzt werden muss [26]. Darüber hinaus imitieren GCNN die typischen LSTM-Mechanismen und erreichen im Bereich der Sprachverarbeitung sehr gute Evaluationsergebnisse [13, 25].

Mit dem vorgeschlagenen KVP Attention-Layer zeigen [27] eine effektive Erweiterung einer LSTM-Architektur durch eine Separierung der LSTM-Ausgabe von Key, Value und Predict [28]. Die dabei verwendeten, neu eingeführten Attention-Layer können innerhalb einer Eingabesequenz Wirkungszusammenhänge zwischen zwei Sequenzelementen identifizieren, die unterschiedliche Positionen in der betrachteten Sequenz besitzen [29]. Der Kontextvektor enthält dabei einen zusammenfassenden Vektor, welcher die kalkulierten Prognosen vergangener Iterationen enthält. Durch die angesprochene Separierung kann eine verbesserte Performance in der Prognose von nachfolgenden Elementen erreicht werden [27].

3 Forschungsansatz

3.1 Testdesign

Zur Prozessprognose wird eine getrennte Betrachtung der verschiedenen Analyseschritte vorgenommen (siehe Abbildung 1). Um den Einfluss verschiedener Vorverarbeitungsschritte auf die Ergebnisse der Prozessprognose zu prüfen, werden im Testdesign grundsätzlich zwei Szenarien unterschieden: mit Vorverarbeitung (Abschnitt 4.1) und ohne Vorverarbeitung (Abschnitt 4.2). Gegenstand der Analyse sind die Prozessdaten. Es wird zwischen Prozess- und Aktivitätsdaten auf Prozessebene und auf Instanzebene (für Instanzmerkmale) unterschieden. Daten der Instanzebene stehen mit Ereignissen auf Ereignisebene in Zusammenhang. Diese haben Ereignismerkmale [30].

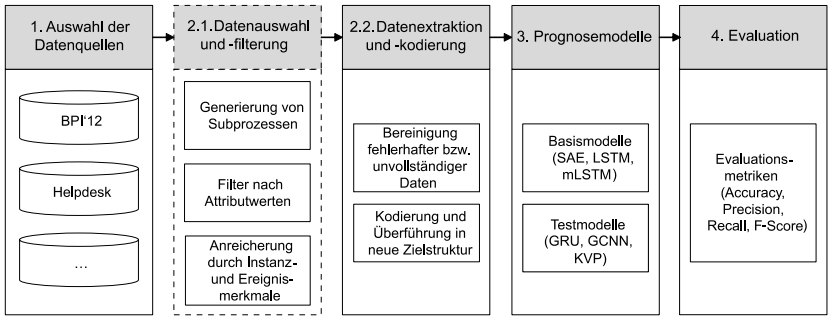

Abbildung 1. Testdesign für die Evaluation von DL-Architekturen zur Prozessprognose

https://doi.org/10.30844/wi_2020_i1-heinrich

Bei der *(1) Auswahl der Datenquellen* (siehe Abschnitt 3.2) werden zwei wesentliche Funktionalitäten realisiert. Zum einen gilt es Ereignislogdaten, falls nicht vorhanden, in das XES-Format zu überführen, welches standardmäßig zur Deklaration von Ereignislogdaten genutzt wird. Zum anderen werden Basiskennzahlen berechnet und als Eingabe für nachfolgende Auswahlschritte als Attributwerte hinzugefügt [31].
Die Eingabeattribute für die in Abschnitt 4.1 durchgeführten Experimente beschränken sich auf die Verwendung der kodierten Sequenzfolgen von Ereignissen [7, 17]. Die vorherzusagende Zielgröße dieser Untersuchung beinhaltet die standardmäßige Definition von Aktivitäten mit der Kennzeichnung sowie deren Lebenszyklusstatus. Mit Festlegung dieser Faktoren werden die einheitliche Bewertung sowie die gleichen Rahmenbedingungen der einzelnen DL-Ansätze innerhalb der zu untersuchenden Szenarien sichergestellt. Es findet keine Vorverarbeitung statt.

Für die weitergehenden Analysen in Abschnitt 4.2 werden in der Phase *(2.1) Datenauswahl und -anreicherung* Vorverarbeitungsschritte variiert. Die Auswahl der verschiedenen Kombinationen aus betrachteten Datensätzen, *Subprozessen*, *Attributen* und angereicherten *Instanz-* und *Ereignismerkmalen* orientiert sich zur Gewährleistung der wissenschaftlichen Vergleichbarkeit an vorangegangenen Untersuchungen, wie sie beispielsweise von [5, 17, 20] durchgeführt wurden.

Generierung von Subprozessen. Ein verbreiteter Ansatz zum Setzen von Rahmenbedingungen ist die separate Betrachtung einzelner Subprozesse, welche in einen übergeordneten Prozess eingebettet sind [8]. Als Subprozesse wurden dabei beispielweise *offene Probleme* (BPI'13 Open) und bereits *geschlossene Probleme* (BPI'13 Closed) unterschieden sowie die Subprozesse BPI'12 A und BPI'12 O [20].

Filter nach Attributwerten. Als Filter der Attributwerte wurden nur Ereignisse mit abgeschlossenen Lebenszyklusstatus betrachtet. Dies erfolgt einerseits auf Basis von vollständigen Prozessen (BPI'12 Complete) und andererseits auf Basis von Subprozessen (BPI'12 Sub Complete).

Anreicherung durch Ereignismerkmale. Bisher wurden die herangezogenen Einfluss- sowie Zielvariablen nicht verändert. Im Rahmen der dritten Phase des Testdesigns wurde der Effekt der Anreicherung von weiteren Merkmalen untersucht. Konkret wurden das Ereignismerkmal *verantwortliche Organisationseinheit* (org:group) und die *verwendeten Ressourcen* (org:resource) hinzugefügt.

Anreicherung durch Instanzmerkmale. Nach der isolierten Betrachtung der Reaktion der Architekturvarianten mit Hinzunahme von Ereignismerkmalen folgt die Untersuchung der Verhaltensweisen von Merkmalen auf Prozessinstanzebene. Dabei werden die Instanzmerkmale *Kritikalität eines Zwischenfalls* (Impact) und *betroffene Organisationseinheit* (Org involved) betrachtet.

Nach den wesentlichen Auswahl- bzw. Anreicherungssmaßnahmen der zugrundeliegenden Datenquellen folgt in beiden Fällen die Extraktion und Transformation relevanter Prognosemerkmale und Zielgrößen in der Phase *(2.2) Datenextraktion und Kodierung*. Die im vorherigen Schritt ausgewählten Merkmale werden zunächst von fehlerhaften Daten bereinigt (z. B. Prozessinstanzeinträge ohne Ereignisdaten). Anschließend werden die bereinigten Daten unter Verwendung des XES-Standards in eine geeignete Zielstruktur für den DL-Algorithmus überführt und entsprechend kodiert.

https://doi.org/10.30844/wi_2020_i1-heinrich

Nach der Implementierung der ausgewählten *(3) Prognosemodelle* kommen für die Phase *(4) Evaluation* etablierte Metriken zur Beurteilung der Klassifikationsgüte zum Einsatz. Diese Testmodelle werden zusammen mit den in Abschnitt 2 genannten, bereits bekannten Basismodellen evaluiert. Zur Evaluation wird die Methodik der 10-fachen Kreuzvalidierung genutzt. Neben der *Accuracy* in Trainings- (Tr) und Validierungsdaten (Va) werden auch die Metriken *Precision*, *Recall* und *F-Score* erhoben [32, 33]. Es werden dabei folgende Varianten berücksichtigt: i) ein *Macro Averaging* (ma) auf Basis des Durchschnitts ohne Berücksichtigung der Klassengrößen, ii) ein *Micro Averaging* (mi), welches demgegenüber die Klassengrößen berücksichtigt und iii) ein *Weighted Averaging* (w) auf Basis des gewichteten Klassendurchschnitts [8].

3.2 Datensätze

Für die Durchführung der Evaluationsstudie wurden verschiedene Datensätze herangezogen. Hierbei wurde einerseits darauf geachtet, dass es sich um weit verbreitete Standarddatensätze handelt, welche im Feld der Prozessprognose häufig für Benchmark-Zwecke eingesetzt werden und andererseits, dass eine hohe Breite an Anwendungsszenarien abgedeckt wird. Die Eigenschaften der ausgewählten Datensätze sind in Tabelle 1 zusammengefasst. Dabei werden die PA-Kennzahlen *Anzahl der Prozessinstanzen* (#P), *Anzahl der Ereignisse* (#E), *Anzahl der vorkommenden Aktivitäten* (#A) und das *Verhältnis der Aktivitäten* zu den vorhandenen Prozessinstanzen (Sparsity) für jeden Prozesskontext angegeben [5, 34].

Tabelle 1. Verwendete Datensätze der Evaluationsstudie

Datensatz	Prozesskontext	#PI	#E	#A	Sparsity
BPI 11 [35]	Krankenhausmanagement	1143	150291	624	0,5459
BPI 12 [36]	Kreditvergabe	13087	262000	36	0,0028
BPI 13 [37]	Problem- und Incident-Management	7554	65533	13	0,0017
HELPDESK [38]	Software-Support	3804	13710	9	0,0024
EnvLog [39]	Genehmigungsauftrag	787	34848	331	0,4206

3.3 Implementierung

Zur Implementierung des Testdesigns wurden verschiedene Softwaremodule herangezogen. Dabei werden die Daten zunächst mit einem CSV to XES Converter in das XES-Format umgewandelt, um die Vorverarbeitung zu ermöglichen. Die auf Python basierende Analyseumgebung operiert unter Benutzung der Pakete *OpenXES* zur Datentransformation, *NLTK* zur Bereitstellung von Funktionalitäten zur Verarbeitung natürlicher Sprache und die ML-Frameworks *Tensorflow* und *SciKit-Learn* zur Implementierung der eigentlichen Modelle. Zusätzlich werden für Datenoperationen und

https://doi.org/10.30844/wi_2020_i1-heinrich

Evaluation einige weitere Pakete wie *Pandas* oder *Seaborn* genutzt. Die Modelle selbst wurden auf einem High-Performance-Computing-Cluster[1] trainiert.

Die in Python erstellten Modelle mit ihren Architekturen, Hyperparametern und Trainingsinputs sind in Tabelle 2 dargestellt. Die Hyperparameter wurden im Grid-Search-Verfahren auf Basis der in Tabelle 2 angegebenen Intervalle optimiert. Konstante Hyperparameter wurden aus den entsprechenden Quellen übernommen.

Tabelle 2. Architektur und Hyperparameter der verwendeten DL-Modelle

Modell	Architektur	Hyperparameter
LSTM	[20]	BS=[10;30], LR=[0,8;1], LRD=1,0, ES=[256;612]
GRU	[11]	BS=[10;30], LR=[0,8;1], LRD=1,0, ES=[256;612]
mLSTM	[34]	siehe [34]
SAE	[17]	BS=[20;100], LR=[0,001;1], LRD=[0.80;0.99]
GCNN	[25]	BS=[20;300], LR=[0,001;1], LRD=[0.80;0.99], ES=[128;256]
KVP	[27]	BS=[10;30], LR=[0,8;1], LRD=1,0, ES=[256;612], AW=[2;6]

Legende: BS = Batch Size, LR = Learning Rate, LRD = Learning Rate Decay, ES = Embedding Size, AW = Attention Window

4 Vergleich von DL-Architekturen für die Prozessprognose

4.1 Analyseergebnisse ohne Vorverarbeitung

Tabelle 3 zeigt die detaillierten Ergebnisse ohne jegliche Berücksichtigung von vorverarbeitenden Maßnahmen. Das Training des mLSTM-Modells für den BPI‘11 Datensatz wurde aufgrund des Überschreitens einer annehmbaren Trainingszeit (>24h) abgebrochen, so dass hier keine Ergebnisse vorliegen.

Sowohl BPI’11 als auch EnvLog zeigen eine erhöhte Komplexität der Ereignislogs. Bei der Untersuchung der Metriken in Bezug auf die einzelnen Architekturvarianten zeigt sich, dass insbesondere das trainierte Prognosemodell KVP im Vergleich zu den anderen Architekturen das beste Abschneiden hinsichtlich der Evaluationsmetriken auf komplexeren Datengrundlagen besitzt. KVP erreicht bei den Datensätzen EnvLog (Sparsity = 0,4206) und BPI’11 (Sparsity = 0,5459) mit einem F1-Scorce$_{micro}$ von 0,78 und 0,69 eine bessere Performance.

Über alle Evaluationsmetriken hinweg kann eine konstant bessere Effektivität in der Prognose der nächsten Aktivität identifiziert werden. Besonders bei der Untersuchung der Macro-Mittelungen der Metriken ist, dass KVP Aktivitäten mit weniger Instanzen effektiver identifiziert werden und somit mit geringer Empfindlichkeit auf kleine Klassengröße reagieren. Mit einem F1-Score$_{micro}$ von 0,52 auf dem EnvLog

[1] 64 Knoten, jeder mit 2x Intel(R) Xeon(R) CPU E5-E5-2680 v3 (12 Cores) @2.50GHz, kein Multi-Threading, 64 GB RAM (2.67 GB per Kern), 128 GB SSD, 4x NVIDIA Tesla K80 (12 GB GDDR RAM).

https://doi.org/10.30844/wi_2020_i1-heinrich

und 0,41 auf dem BPI'11 Datensatz zeigt SAE demgegenüber die schlechteste Performance. Der Effekt der erhöhten Komplexität der Ereignislogs fällt bei der Architektur KVP eher geringer aus als den anderen Architekturvarianten. Über alle betrachteten Architekturvarianten kann beim Datensatz EnvLog ein starkes Overfitting identifiziert werden. Besonders zeigen GCNN und mLSTM mit einer Differenz von 0,26 bzw. 0,3 zwischen $Accuracy_{tr}$ und $Accuracy_{va}$ auf dem EnvLog Datensatz und SAE mit 0,31 auf dem BPI'11 Datensatz Probleme in der Generalisierbarkeit der trainierten Klassifikatoren. Es zeichnet sich eine Tendenz ab, dass eine fallende Prognoseperformance mit steigender Sparsity zusammenhängt. Alle betrachteten DL-Architekturen zeigen eine Reaktion auf die Entwicklung der zugrundeliegenden Komplexität. Der Unterschied liegt jedoch in der Intensität der fallenden Performance, welche im Vergleich zu den weiteren Ansätzen bei KPV geringer ausfällt.

Tabelle 3. Ergebnisse (Ø) der durchgeführten Experimente ohne Vorverarbeitung[2]

		Accuracy		**Precision**			**F1-Score**			**Recall**		
Datensatz	**Modell**	*Tr*	*Va*	*w*	*mi*	*ma*	*w*	*mi*	*ma*	*w*	*mi*	*ma*
BPI'11	LSTM	0,61	0,62	**0,78**	0,62	0,39	0,67	0,62	0,37	0,62	0,62	0,39
	GRU	0,60	0,60	0,66	0,60	0,21	0,62	0,60	0,21	0,60	0,60	0,24
	mLSTM	-	-	-	-	-	-	-	-	-	-	-
	SAE	0,72	0,41	0,43	0,41	0,06	0,37	0,41	0,04	0,41	0,41	0,05
	GCNN	0,70	0,57	0,56	0,55	0,21	0,53	0,55	0,17	0,55	0,55	0,17
	KVP	**0,80**	**0,69**	0,73	**0,69**	**0,42**	**0,70**	**0,69**	**0,43**	**0,69**	**0,69**	**0,50**
BPI'12	LSTM	0,88	0,84	0,90	0,84	0,69	0,86	0,84	0,70	0,84	0,84	0,76
	GRU	0,56	0,73	0,83	0,73	0,47	0,76	0,73	0,47	0,73	0,73	0,53
	mLSTM	0,83	0,80	0,78	0,80	0,71	0,77	0,80	0,63	0,80	0,80	0,63
	SAE	**0,91**	0,69	0,74	0,68	0,66	0,67	0,68	0,56	0,68	0,68	0,56
	GCNN	0,87	**0,87**	0,86	**0,87**	**0,79**	0,84	**0,87**	0,69	**0,87**	**0,87**	0,69
	KVP	0,79	0,86	**0,93**	0,86	0,73	**0,88**	0,86	**0,72**	0,86	0,86	**0,76**
BPI'13	LSTM	**0,72**	**0,70**	0,79	**0,70**	**0,38**	**0,73**	**0,70**	**0,36**	**0,70**	**0,70**	**0,37**
	GRU	0,62	0,67	**0,83**	0,67	0,29	0,73	0,67	0,27	0,67	0,67	0,32
	mLSTM	0,67	0,57	0,56	0,57	0,26	0,51	0,57	0,23	0,57	0,57	0,28
	SAE	0,59	0,44	0,60	0,44	0,23	0,46	0,44	0,16	0,44	0,44	0,16
	GCNN	0,70	0,68	0,59	0,68	0,32	0,61	0,68	0,29	0,68	0,68	0,29
	KVP	0,70	0,67	0,80	0,67	0,37	0,71	0,67	0,33	0,67	0,67	0,34
Helpdesk	LSTM	0,84	0,85	**0,91**	0,85	0,44	**0,88**	0,85	0,43	0,85	0,85	0,42
	GRU	0,73	0,85	0,91	0,85	0,44	0,88	0,85	0,43	0,85	0,85	0,43
	mLSTM	0,82	0,82	0,77	0,82	0,46	0,78	0,82	0,40	0,82	0,82	0,40
	SAE	0,74	0,74	0,66	0,74	0,31	0,69	0,74	0,24	0,74	0,74	0,25
	GCNN	0,85	**0,86**	0,85	**0,86**	**0,52**	0,85	**0,86**	**0,52**	**0,86**	**0,86**	**0,52**
	KVP	**0,86**	0,85	0,91	0,85	0,44	0,88	0,85	0,43	0,85	0,85	0,42
EnvLog	LSTM	0,67	0,58	0,69	0,58	0,36	0,61	0,58	0,35	0,58	0,58	0,40
	GRU	0,85	0,67	0,71	0,67	0,46	0,68	0,67	0,46	0,67	0,67	0,48
	mLSTM	0,83	0,53	0,55	0,54	0,39	0,53	0,54	0,38	0,54	0,54	0,39
	SAE	0,67	0,52	0,71	0,52	0,47	0,58	0,52	0,36	0,52	0,52	0,32
	GCNN	0,88	0,62	0,64	0,62	0,45	0,60	0,62	0,39	0,62	0,62	0,39
	KVP	**0,89**	**0,78**	**0,81**	**0,78**	**0,71**	**0,79**	**0,78**	**0,71**	**0,78**	**0,78**	**0,73**

[2] Ergebnisse mit Vorverarbeitung unter https://www.researchgate.net/publication/337243603.

https://doi.org/10.30844/wi_2020_i1-heinrich

4.2 Zusammenfassung der Analyseergebnisse mit Vorverarbeitungsschritten

Evaluation für Generierung von Subprozessen. Bei den durchgeführten Experimenten ist zu erkennen, dass die Ergebnisse über alle Datensätze hinweg ein besseres Gesamtergebnis zeigen. Für alle Datensätze können jedoch bei der isolierten Analyse der maximalen Werte keine beobachtbar steigenden Ergebnismetriken für die Micro-Mittelungen identifiziert werden. Durch die Verringerung der Aktivitätsklassen zeigen die Experimente im Rahmen dieser Phase des Testdesigns stabilere Kennzahlen durch die Verwendung der Macro-Kalkulation der Metriken sowie der Differenz beider Kalkulationsansätze. Weiter zeigt sich, dass auf dem Datensatz BPI'13 Closed, wie bereits in der ursprünglichen BPI'13 Datengrundlage, die LSTM-Architektur den besten F1-$Score_{micro}$ mit 0,68 aufweist. Es fällt allerdings auf, dass der GCNN-Ansatz ein besseres Abschneiden mit einem F1-$Score_{macro}$ von 0,49 hinsichtlich der Prognose kleinerer Aktivitätsklassen in Gegenüberstellung des LSTM zeigt (F1-$Score_{macro}$ = 0,42). Ein ähnliches Muster in den Evaluationsmetriken lässt sich am BPI'13 Open Datensatz feststellen, in dem die GCNN-Architektur eine deutlich bessere Macro-Mittelung der F1-Scores besitzt und auch mit einem F1-$Scores_{micro}$ von 0,65 den marginalen Bestwert enthält. Homogener verteilen sich die maximalen Metriken in den Subprozessen des BPI'12-Datensatzes. Hier weist GCNN bezüglich der Subprozesse BPI'12 A in allen Evaluationsmetriken den Maximalwert auf. Hinsichtlich dessen zeigt sich diese Architekturalternative als beste Variante, wobei sie mit einem Unterschied des F1-$Scores_{micro}$ von 0,05 zur nächstbesten Alternative mit KVP den größten Abstand besitzt. Deutlich knapper prägen sich die Evaluationsmetriken hinsichtlich der Subprozesse BPI'12 O aus. Die Performance bei der Betrachtung aller vorgenommenen Klassifikationen wird jedoch im Allgemeinen vom GCNN dominiert (F1-$Score_{micro}$ = 0,86).

Evaluation für Filtern nach Attributwerten. Die Ergebnisse zeigen, dass die allgemeine Performance der Evaluationsmetriken ausgehend von dem Ereignislog BPI'12 mit einer maximalen $Accuracy_{va}$ von 0,87 auffallend auf eine $Accuracy_{va}$ von 0,81 fällt. Das gleiche Muster wie in den vorherigen Untersuchungen zeigt sich in der Beobachtung, dass die GCNN Architektur in Aktivitätsklassen mit weniger Instanzen minimal bessere Ergebnisse zeigen. Die isolierte Bewertung der GCNN Architektur auf dem BPI'12 Complete Datensatz zeigt mit den maximalen Werten der Macro-(0,61) sowie Micro-Mittelungen (0,81) ein robustes Prognosemodell. Auf dem BPI'12 Sub Complete Datensatz hingegen erweist sich die SAE Architektur als dominierend beste Alternative mit einem F1-$Score_{macro}$ von 0,6 und einem F1-$Score_{micro}$ von 0,81.

Evaluation für Anreicherung durch Ereignismerkmale. Das Anreichern der Prognoseingaben durch Ereignismerkmale lässt vermuten, dass die erhobenen Evaluationsmetriken durch die Bereitstellung zusätzlicher Informationen eine verbesserte Performance zeigen [17, 20]. Mit der Beobachtung der $Accuracy_{tr}$ der vorliegenden Architekturen kann dies im Rahmen dieses Testdesigns untersucht werden. Das vermutete Verhalten kann jedoch nicht hinsichtlich aller betrachteten Datengrundlagen

https://doi.org/10.30844/wi_2020_i1-heinrich

beobachtet werden. Eine auffallende Steigerung der $Accuracy_{tr}$ zeigt GCNN mit Hinzunahme der gewählten Einflussmerkmale. Besonders intensiv lässt sich dieser Unterschied auf den Datensätzen BPI'13 mit 0,17 bemerken. Die geringste Intensität zeigt der Effekt der beschriebenen Maßnahme auf der Architektur KVP. Hier lassen sich minimale positive wie negative Unterschiede in der $Accuracy_{tr}$ identifizieren. Die Evaluationsmetriken auf den Validierungsdatensätzen weisen ebenfalls keine Stetigkeit auf. Jedoch kann in gewissen Datensätzen eine verbesserte Performance nachgewiesen werden. Von allen Evaluatuationsmetriken, sowohl Macro- als auch Micro-Mittelungen, kann die GCNN-Architektur als deutlich beste Alternative gewertet werden. Mit einem $F1\text{-}Score_{micro}$ von 0,93 wird die höchste dokumentierte Metrik identifiziert. Auffällig ist, dass sich die neu hinzugefügte Architektur KVP, mit Ausnahme des BPI'12 O Datensatzes, konstant als zweitbeste Alternative beweist. Auf dem EnvLog Datensatz zeigen alle Ansätze ein Abfallen der Validierungsergebnisse. Einen besonders auffallenden Verlust mit dem zusätzlichen Einflussmerkmal zeigt KVP mit einem Delta von 0,06.

Evaluation für Anreicherung durch Instanzmerkmale. Im Vergleich zur Anreicherung mit Ereignismerkmalen führen die zusätzlichen Instanzmerkmale bei den Modellen GCNN, GRU, mLSTM und SAE zu generellen Verbesserungen. Im Gegensatz dazu sind jedoch bei den Architekturen KVP und LSTM auch Verschlechterungen bezüglich der Prognosequalität festzustellen. Trotz der Verschlechterung des LSTM-Netzes im Vergleich zu den Experimenten mit Ereignismerkmalen auf dem Datensatz BPI'13 liefert das Modell zweimal mit BPI'13 {*Impact*} ($F1\text{-}Score_{micro}$= 0,69) und BPI'13 {*Org involved*; *Impact*} ($F1\text{-}Score_{micro}$ = 0,70) die besten Prognoseergebnisse. Den größten Verlust durch die Hinzunahme von Instanzmerkmalen im Vergleich zu Ereignismerkmalen zeigt die KVP-Architektur. GCNN zeigt demgegenüber zwar Verbesserungen mit Instanzmerkmalen, diese führen jedoch nicht dazu, dass eine beobachtbare Steigerung der Prognosequalität zustande kommt. Mit der einzelnen Betrachtung der neu selektierten Prognosemodelle kann mit der GRU-Netzstruktur auf dem BPI'13 {*Org involved*} der größte positive Unterschied des F1-Scores ($F1\text{-}Score_{micro}$ = 0,71) beobachtet werden. Auffällig ist bei der detaillierteren Sichtung der Macro- und Micro-Mittelungen, dass mit einem $F1\text{-}Score_{macro}$ von 0,36 aufgrund des erhöhten $Recall_{macro}$ von 0,43 die beste Identifikation von Klassen mit geringer Instanzanzahl gelingt. Mit allen verwendeten Instanzmerkmalen auf dem BPI'13 Datensatz kann eine Verbesserung der Trainings- sowie Validierungsmetriken identifiziert werden.

5 Diskussion

Durch die Modifikationen der Einflussvariablen veränderten sich auch die zugrundeliegenden Prozesscharakteristiken der Ereignisdaten. Zur Veranschaulichung der Auswirkungen ist in Abbildung 2 der Einfluss der Prozesscharakteristiken auf die Prognoseperformance dargestellt.

https://doi.org/10.30844/wi_2020_i1-heinrich

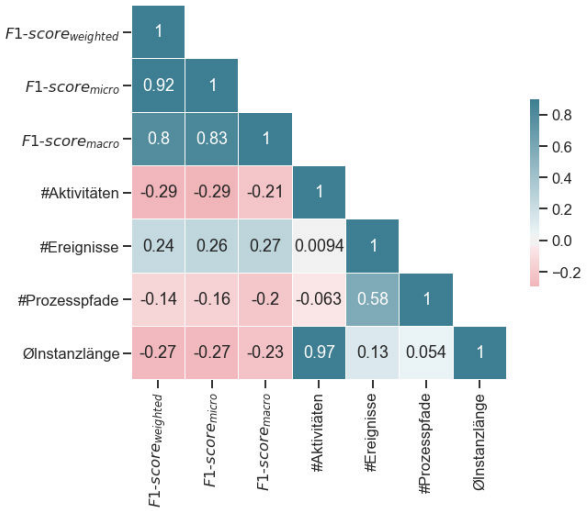

Abbildung 2. Korrelationsmatrix der Ereignislogcharakteristika aller erhobenen F1-Scores

Die Abbildung zeigt, dass der größte negative Einfluss der berücksichtigten Eigenschaften in der Anzahl der designierten Aktivitätsklassen mit -0,29 die stärkste negative Korrelation mit den F1-Scores besitzt. Weiter zeigt sich die durchschnittliche Instanzlänge als weitere Eigenschaft, welche mit steigendem Wert die Performance der Prognosemodelle tendenziell negativ beeinflusst. Mit erhöhter Anzahl an dokumentierten Ereignissen kann eine positive Entwicklung mit einem Korrelationskoeffizienten von 0,26 identifiziert werden. Grundsätzlich zeigen die Ergebnisse der Korrelationsanalyse eher schwache Zusammenhänge zwischen den Charakteristika der Ereignislogs und den Auswertungsmetriken, so dass hier auf eine Ableitung von Korrelationshypothesen verzichtet wird.

Zudem lassen sich mit der Betrachtung der relativen Ränge der F1-Scores aus Abbildung 3 die Prognoseergebnisse der einzelnen Architekturen auf aggregierter Ebene gegenüberstellen. Hieraus kann die allgemeine Performance aller betrachteten Experimente berücksichtigt werden. Allerdings gilt es zu bemerken, dass diese von der Wahl und Anzahl der verschiedenen Szenarien beeinflusst werden und sich somit nur bedingt allgemeingültige Aussagen aufstellen lassen. Zu entnehmen ist jedoch eine grobe Gegenüberstellung, wo beispielsweise beobachtet werden kann, dass der GCNN-Ansatz beim direkten Vergleich über beide Kalkulationsvarianten hinweg prinzipiell allen anderen Architekturen überlegen ist.

https://doi.org/10.30844/wi_2020_i1-heinrich

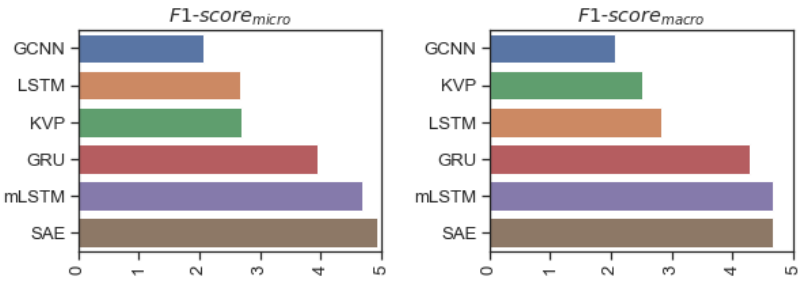

Abbildung 3. Relativer Rang über alle Datensätze hinsichtlich der F1-Scores

Abbildung 4 zeigt beispielhaft die Entwicklungen der Architekturen über die Untersuchungsaspekte Datenquelle, Ereignismerkmal, Instanzmerkmal sowie die kombinierte Verwendung. Festzuhalten ist, dass SAE mit steigender Anzahl an Merkmalen eine deutliche Steigerung der Prognosequalität zeigt. Der Schwäche der fehlenden Robustheit durch niedrige Macro-Mittelungen wird mit zusätzlicher Aufnahme von Merkmalen entgegengewirkt. Überraschend zeigt dieser Ansatz im Vergleich zu anderen durchgeführten Experimenten hohe F1-Scores$_{macro}$. Die GCNN Architektur zeigt demgegenüber nur minimale Reaktionen auf die Hinzunahme von weiteren Merkmalen auf Instanz- und Ereignisebene auf dem BPI'12 Datensatz.

Im Gegenzug hierzu zeigt LSTM eine größere Empfindlichkeit in Bezug auf die Merkmalsauswahl. So können Sprünge je nach definiertem Merkmal erhoben werden. Mit dem Datensatz BPI'13 kann im Gegensatz zum BPI'12 eine Entwicklung der Prognosequalität der Ansätze LSTM und GCNN mit der Hinzunahme von weiteren Merkmalen registriert werden.

https://doi.org/10.30844/wi_2020_i1-heinrich

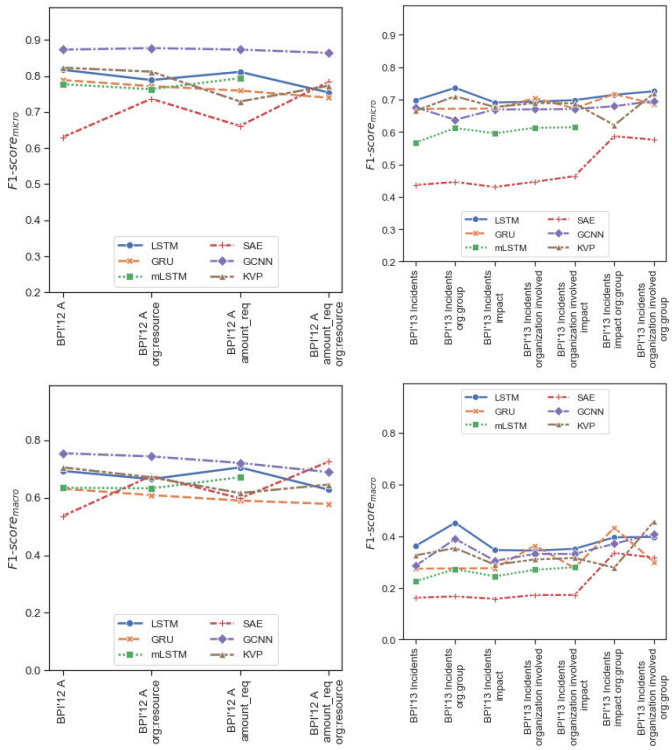

Abbildung 4. Verläufe der F1-Scores bei unterschiedlichen Einflussmerkmalen

Zusammenfassend lässt sich festhalten, dass es keine dominierende DL-Architektur gibt, die über alle Experimente hinweg die beste Performance besitzt. Jedoch verdeutlichen die Ergebnisse, dass je nach Prozesskontext, Datenverfügbarkeit und favorisierter Metrik die passende Architektur mit hoher Vorhersagegenauigkeit zu wählen ist. Im Vergleich mit bisher bekannten Ansätzen wie LSTM und mLSTM zeigen insbesondere die beiden perspektivischen Architekturen GCNN und KVP grundlegend konkurrenzfähige Prognosequalitäten. Bei der Betrachtung der GRU ist zu sehen, dass diese einerseits eine eher verschlechterte Performance aufweist, andererseits in speziellen Anwendungsfällen die beste Prognosequalität bietet.

Gleichzeitig können auf Basis der Ergebnisse nur bedingt verallgemeinerbare Heuristiken abgeleitet werden, wie z. B. dass der Attention-Layer-basierte Ansatz bei komplexeren Datensätzen geringere Performanceeinbußen aufweist. Für weitere Implikationen streuen die Evaluationsmetriken der verschiedenen Architekturen jedoch zum Teil zu stark über die einzelnen Teilstudien hinweg, ohne dass sich unmittelbar systematische Effekte erkennen lassen. Eine Interpretation der Performanceschwankungen gestaltet sich insbesondere aufgrund der Black-Box-Eigenschaften sämtlicher DL-Architekturen als schwierig, da nicht komplett nachvollzogen werden kann, welche konkreten Architekturkomponenten und Hyperparameter aus einer Vielzahl an festzulegenden Stellschrauben für die Performanceveränderungen verantwortlich sind.

https://doi.org/10.30844/wi_2020_i1-heinrich

Dies ist auch der Grund, weshalb die Ergebnisse angrenzender Studien, z. B. auf Basis von LSTM-Architekturen [20, 34], nur bedingt rekonstruiert werden konnten. So zeigte sich während der Trainingsprozeduren, dass bei verschiedenen Datengrundlagen bereits geringfügige Modifikationen von Hyperparametern zu starken Performanceänderungen führten. Umso wichtiger gestaltet sich die Durchführung breitangelegter, systematischer Evaluationsstudien unter möglichst gleichbleibenden Vergleichsbedingungen, so wie sie durch den vorliegenden Beitrag mithilfe eines strukturierten Testdesigns motiviert wurden. Hierzu bestand in den aktuellen Evaluationsstudien der Fokus darin, möglichst optimale Architekturspezifikationen mittels Grid-Search-Verfahren zu ermitteln, um die verschiedenen DL-Netzwerkestrukturen auf einer hohen Abstraktionsebene gegenüberzustellen. In anschließenden Arbeiten ist es jedoch notwendig, stärker ins Detail zu gehen, um auf Basis von Sensitivitätsanalysen und weiteren Explainable-AI-Methoden [40] ein Verständnis darüber zu erlangen, welchen Einfluss die verschiedenen Architekturkomponenten, Verarbeitungsschichten und Modellparameter je nach Prozesskontext und Datengrundlagen auf die Prognosegenauigkeit haben.

6 Zusammenfassung und Ausblick

Im vorliegenden Beitrag haben wir mit GRU, GCNN und KVP aktuelle Architekturkonzepte des DL auf ihre Eignung für die Prognose von Folgeereignissen in laufenden Prozessinstanzen gegenüber bereits etablierten Architekturen des DL überprüft. Die Ergebnisse mit den erhobenen Evaluationsmetriken legen nahe, dass GCNN und KVP über alle Analysen hinweg sehr gute Ergebnisse liefern, GRU aber nur in Einzelfällen die anderen dominiert. Die hier betrachteten perspektivischen DL-Architekturen eignen sich damit grundsätzlich alle zur Prozessprognose.

Ersichtlich wird aber, dass eine bestmögliche Wahl der zu untersuchenden Prognosemodelle nur schwer zu treffen ist. Annahmen zur Architekturauswahl können nur durch die Evaluation im Problemkontext validiert werden. Es könnten entsprechend noch weitere Architekturvarianten für die Anwendung im Prozesskontext in Frage kommen. Eine Entscheidung sollte daher immer auf Basis der Prozessdaten, der Laufzeit und auch des Implementierungsaufwands erfolgen, da bspw. nicht immer Standard-Packages zur Verfügung stehen. Es gilt auch zu berücksichtigen, dass Parameter nicht beliebig granular angepasst und Modelle durchtrainiert werden können, da es zum Teil mehrere Stunden dauert, bis Architekturen mit einer Vielzahl an Parametern auf Basis komplexer Datensätze trainiert sind. Wir verstehen die erhobenen aktuellen DL-Netzstrukturen sowie Verarbeitungsschichten somit als Grundlage für weitere Analysen.

References

1. Hilbert, A., Zschech, P.: Process Analytics. WISU. 942–948 (2016).

https://doi.org/10.30844/wi_2020_i1-heinrich

2. van der Aalst, W.M.P., Zhao, J.L., Wang, H.J.: Business Process Intelligence: Connecting Data and Processes. ACM Transactions on Management Information Systems. 5,1-7 (2015).
3. zur Muehlen, M., Shapiro, R.: Business process analytics. In: Handbook on Business Process Management 2. pp. 137–157. Springer (2010).
4. Beheshti, S.M.R., Benatallah, B., Sakr, S., Grigori, D., Motahari-Nezhad, H.R., Barukh, M.C., Gater, A., Ryu, S.H.: Process Analytics: Concepts and Techniques for Querying and Analyzing Process Data. Springer International Publishing (2016).
5. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Petrucci, G., Yeshchenko, A.: An Eye into the Future: Leveraging A-priori Knowledge in Predictive Business Process Monitoring. In: International Conference on Business Process Management. pp. 252–268 (2017).
6. Di Francescomarino, C., Dumas, M., Federici, M., Ghidini, C., Maggi, F.M., Williams, R., Simonetto, L.: Genetic algorithms for hyperparameter optimization in predictive business process monitoring. Information Systems. 74, 67–83 (2018).
7. Evermann, J., Rehse, J.-R., Fettke, P.: XES Tensorflow - Process Prediction using the Tensorflow Deep-Learning Framework. CoRR. abs/1705.01507, (2017).
8. Mehdiyev, N., Lahann, J., Emrich, A., Enke, D., Fettke, P., Loos, P.: Time Series Classification using Deep Learning for Process Planning: A Case from the Process Industry. Procedia Computer Science. 114, 242–249 (2017).
9. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature. 521, 436–444 (2015).
10. Heinrich, K., Zschech, P., Skouti, T., Griebenow, J., Riechert, S.: Demystifying the Black Box: A Classification Scheme for Interpretation and Visualization of Deep Intelligent Systems. Americas Conference on Information Systems, Cancún, Mexico (2019).
11. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734 (2014).
12. Heinrich, K., Zschech, P., Möller, B., Breithaupt, L., Maresch, J.: Objekterkennung im Weinanbau – Eine Fallstudie zur Unterstützung von Winzertätigkeiten mithilfe von Deep Learning. HMD Praxis der Wirtschaftsinformatik. (2019).
13. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition. 77, 354–377 (2018).
14. Conforti, R., De Leoni, M., La Rosa, M., Van Der Aalst, W.M.: Supporting risk-informed decisions during business process execution. In: International Conference on Advanced Information Systems Engineering. pp. 116–132. Springer (2013).
15. Del-RiO-Ortega, A., Resinas, M., Cabanillas, C., Ruiz-Cortes, A.: On the definition and design-time analysis of process performance indicators. Information Systems. 38, 470–490 (2013).
16. Evermann, J., Rehse, J.-R., Fettke, P.: A Deep Learning Approach for Predicting Process Behaviour at Runtime. In: Business Process Management Workshops, LNBIP (2016).
17. Mehdiyev, N., Evermann, J., Fettke, P.: A Multi-stage Deep Learning Approach for Business Process Event Prediction. In: IEEE 19th Conference on Business Informatics (CBI). pp. 119–128 (2017).

https://doi.org/10.30844/wi_2020_i1-heinrich

18. da Silva, I.N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L.H.B., dos Reis Alves, S.F.: Artificial Neural Network Architectures and Training Processes. In: Artificial Neural Networks : A Practical Course. pp. 21–28. Springer International Publishing, Cham (2017).
19. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation. 9, 1735–1780 (1997).
20. Evermann, J., Rehse, J.-R., Fettke, P.: Predicting process behaviour using deep learning. Decision Support Systems. 100, 129–140 (2017).
21. Mehdiyev, N., Evermann, J., Fettke, P.: A Novel Business Process Prediction Model Using a Deep Learning Method. Business and Information Systems Engineering. (2018).
22. Krause, B., Lu, L., Murray, I., Renals, S.: Multiplicative LSTM for sequence modelling. arXiv:1609.07959 [cs, stat]. (2016).
23. Bansal, T., Belanger, D., McCallum, A.: Ask the GRU: Multi-task Learning for Deep Text Recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 107–114. ACM, New York, NY, USA (2016).
24. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning. pp. 2342–2350 (2015).
25. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language Modeling with Gated Convolutional Networks. In: International Conference on Machine Learning. pp. 933–941 (2017).
26. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in neural information processing systems. pp. 2042–2050 (2014).
27. Daniluk, M., Rocktäschel, T., Welbl, J., Riedel, S.: Frustratingly short attention spans in neural language modeling. arXiv preprint arXiv:1702.04521. (2017).
28. Brown, A., Tuor, A., Hutchinson, B., Nichols, N.: Recurrent Neural Network Attention Mechanisms for Interpretable System Log Anomaly Detection. In: Proceedings of the First Workshop on Machine Learning for Computing Systems (2018).
29. Werlen, L.M., Pappas, N., Ram, D., Popescu-Belis, A.: Self-Attentive Residual Decoder for Neural Machine Translation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 1366–1379 (2018).
30. Aalst, W.M.P. van der: Process mining: data science in action. Springer (2016).
31. Verbeek, H.E., Bose, R.J.C.: Prom 6 tutorial. Technical report, Tech. Rep. (2010).
32. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing & Management. 45, 427–437 (2009).
33. Shepperd, M., Bowes, D., Hall, T.: Researcher bias: The use of machine learning in software defect prediction. IEEE Transactions on Software Engineering. 40, 603–616 (2014).
34. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: International Conference on Advanced Information Systems Engineering. pp. 477–492. Springer (2017).
35. Van Dongen, B.F.: Real-life event logs - Hospital log. Eindhoven University of Technology (2011).
36. Van Dongen, B.F.: BPI Challenge 2012. Eindhoven University of Technology (2012).
37. Steeman, W.: BPI Challenge 2013. Ghent University (2013).

https://doi.org/10.30844/wi_2020_i1-heinrich

38. Verenich, I.: Helpdesk, https://data.mendeley.com/datasets/39bp3vv62t/1, last accessed 2018/05/25.
39. Buijs, J.C.A.M.: Environmental permit application process ('WABO'), CoSeLoG project – Municipality 4. Eindhoven University of Technology (2014).
40. Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access. 6, 52138–52160 (2018).

https://doi.org/10.30844/wi_2020_i1-heinrich