

Hacia una Mejor Monitorización Predictiva: Combinando Redes Neuronales de Grafos con Redes Neuronales Recurrentes en Minería de Procesos*

Efrén Rama-Maneiro^[0000-0002-0338-4586], Juan Vidal^[0000-0002-8682-6772], and
Manuel Lama^[0000-0001-7195-6155]

Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade
de Santiago de Compostela, Santiago de Compostela, España.
{efren.rama.maneiro,juan.vidal,manuel.lama}@usc.es

Resumen La monitorización predictiva de procesos de negocio, un subcampo clave de la minería de procesos, se centra en predecir características del próximo evento o secuencia de eventos futuros, entre otros aspectos relevantes. A pesar de que se han propuesto numerosas aproximaciones basadas en aprendizaje profundo, como redes neuronales recurrentes y convolucionales, estas no aprovechan de manera óptima la información estructural presente en los modelos de procesos. Este artículo presenta una propuesta innovadora que combina redes neuronales recurrentes con redes convolucionales de grafos, logrando un aprendizaje simultáneo de la información espaciotemporal tanto del registro de eventos como del modelo de proceso. Esta integración permite la identificación de patrones comunes en modelos de procesos, como bucles o paralelos, evitando la pérdida de información durante el aprendizaje del modelo. Un análisis experimental utilizando registros de eventos reales demuestra que nuestra aproximación presenta una mayor consistencia y supera a las técnicas actuales del estado del arte en el ámbito de la monitorización predictiva.

Keywords: Monitorización predictiva · Minería de procesos · Aprendizaje profundo · Redes neuronales recurrentes · Redes neuronales de grafos

* Este trabajo ha recibido apoyo financiero de la Consellería de Educación, Universidade e Formación Profesional (acreditación 2019-2022 ED431G-2019/04), del Fondo Europeo de Desarrollo Regional (FEDER), que reconoce al CiTIUS - Centro Singular de Investigación en Tecnoloxías Intelixentes da Universidade de Santiago de Compostela como Centro de Investigación del Sistema Universitario de Galicia, y del Ministerio de Ciencia e Innovación de España (proyectos PDC2021-121072-C21 y PID2020-112623GB-I00). E. Rama-Maneiro cuenta con el apoyo del Ministerio de Educación de España, bajo el plan nacional FPU (FPU18/05687). Además, los autores también desean agradecer las instalaciones de supercomputación proporcionadas por el CESGA.



1. Introducción

La minería de procesos es una disciplina que busca describir el comportamiento futuro y pasado de un proceso de negocio, dada la información disponible en un registro de eventos [1]. Se pueden realizar múltiples análisis sobre un registro de eventos, como, por ejemplo, descubrir su modelo de proceso subyacente, mejorar un modelo de proceso ya existente utilizando la información del registro de eventos, o comparar un modelo de proceso con un registro de eventos para verificar su grado de conformidad. La monitorización predictiva es un subcampo de la minería de procesos que intenta prever cómo se desarrollará un caso en ejecución en el futuro [2]. Estas predicciones pueden involucrar cuál será la próxima actividad o secuencia de actividades, cuándo ocurrirá el próximo evento o cuánto tiempo queda hasta el final del caso. Se han aplicado muchas técnicas de aprendizaje automático a la monitorización predictiva, aunque los enfoques basados en el aprendizaje profundo son los que presentan un mejor rendimiento [3].

Casi todas las aproximaciones de aprendizaje profundo en monitorización predictiva construyen un modelo predictivo exclusivamente a partir de la información disponible en las trazas de los procesos. Estas aproximaciones ignoran la información estructural implícita disponible en los modelos de proceso. Esto posibilita que, al disponer únicamente de las trazas, las redes neuronales sean incapaces de aprender ciertos patrones de comportamiento como los bucles o los paralelos. Por otro lado, las aproximaciones que si usan los modelos de proceso como entrada obtienen información de la conectividad entre las actividades del modelo. Sin embargo, estas aproximaciones no pueden aprovechar al máximo la información estructural y temporal disponible en el registro de eventos ya que ignoran el orden en el que aparecen los eventos en el prefijo, están sujetos a sobrescribir características si alguna actividad aparece repetida en el prefijo, y la mayoría de ellos se basan en modelos de procesos representados como grafos de seguimiento directo (DFG), que son menos expresivos que las redes de Petri.

En este artículo, planteamos la hipótesis de que el rendimiento de los modelos predictivos puede mejorarse combinando la información de las trazas y el modelo de proceso y aprovechando la información sobre cómo se comporta la ejecución del modelo de proceso a lo largo del tiempo. Proponemos capturar esta ejecución utilizando redes neuronales de grafo (GNN) y redes neuronales recurrentes (RNN), teniendo en cuenta la secuencia completa de estados y no solo el estado de la repetición anterior a la predicción, es decir, las activaciones de eventos previos al que se está reproduciendo se mantienen en la matriz de características. Esto debería facilitar la detección de algunos patrones de comportamiento comunes en los modelos de proceso, como bucles y paralelos, además de los patrones de comportamiento relacionados con el tiempo. Hemos validado nuestra propuesta con las principales aproximaciones del estado del arte utilizando diez conjuntos de datos reales públicamente disponibles, demostrando que nuestra aproximación mejora la precisión de las predicciones en casi todos los casos.

2. Aproximación

Esta sección presenta nuestra propuesta, “**P**redictor de **P**rocesos con **C**onvolución de **G**rafos **R**ecurrentes” (PPCGR), cuyo objetivo es predecir la siguiente actividad de un caso en ejecución utilizando la información contenida tanto en el modelo de proceso como en el registro de eventos.

Nuestro modelo predictivo está basado en Redes Neuronales Recurrentes de Grafo (GRNN), que son una combinación de redes convolucionales de grafos (GCN) [15] y RNN. Las GCN son un tipo de GNN que operan sobre grafos, capturando dependencias entre los nodos del mismo. Para tener en cuenta la direccionalidad en las redes de Petri en un contexto de monitorización predictiva, se modifica el operador GCN utilizando la matriz laplaciana normalizada con paseo aleatorio en lugar de la matriz laplaciana [16]. La GRNN integra las capacidades de las GCN y las RNN para aprender dependencias espacio-temporales en la entrada, siendo más efectiva que usar GCN y RNN [17]. En nuestro modelo, utilizamos como RNN la capa “Unidad Recurrente con Compuertas” (GRU). Nótese que la utilización de la recurrencia en combinación a la red convolucional de grafos permite capturar información del prefijo completo, independientemente de que la información se sobrescriba en la fase de codificación debido a la presencia de bucles.

La Figura 1 ilustra el flujo de datos de nuestra aproximación. Nuestro punto de partida es un registro de eventos y un prefijo, sobre el cual se quiere realizar la predicción de la siguiente actividad. Dado el registro de eventos, aplicamos el Split Miner [18] con varios hiperparámetros para generar un conjunto de modelos de procesos, cada uno correspondiente a una configuración específica de hiperparámetros. A continuación, se selecciona el mejor modelo de proceso, basado en su ajuste en el conjunto de validación. Este modelo se convierte en un grafo de lugares que encapsula las relaciones entre los lugares de la red de Petri. El grafo se representa como una matriz de adyacencia y se transforma en una matriz laplaciana normalizada de paseo aleatorio que sirve como una de las entradas para nuestro modelo de Red Neuronal Recurrente de Grafo (GRNN). Como el modelo de proceso es común a todos los prefijos del registro, la extracción del modelo se realiza solo una vez, justo antes de comenzar a reproducir cualquier prefijo sobre el modelo.

Al obtener un modelo de proceso, reproducimos el prefijo, generando una matriz de características de nodo que representa las activaciones de los lugares y transiciones para ese prefijo. Además, creamos una matriz de características de atributo que contiene información adicional, como las actividades ejecutadas, datos temporales relacionados con eventos y otros atributos. Como resultado, las entradas de nuestro modelo GRNN consisten en (i) la matriz de adyacencia y (ii) la matriz de características de nodo (secuencia de activaciones en la red de Petri después de repetir el prefijo). Sobre la salida de la operación GRNN, se aplica una operación de *readout* que combina las características aprendidas de todos los nodos y genera una salida única que representa el estado completo del grafo para todo el prefijo. Este resumen se concatena a la matriz de características de atributos y se alimenta a una red recurrente de “memoria a largo-corto plazo”

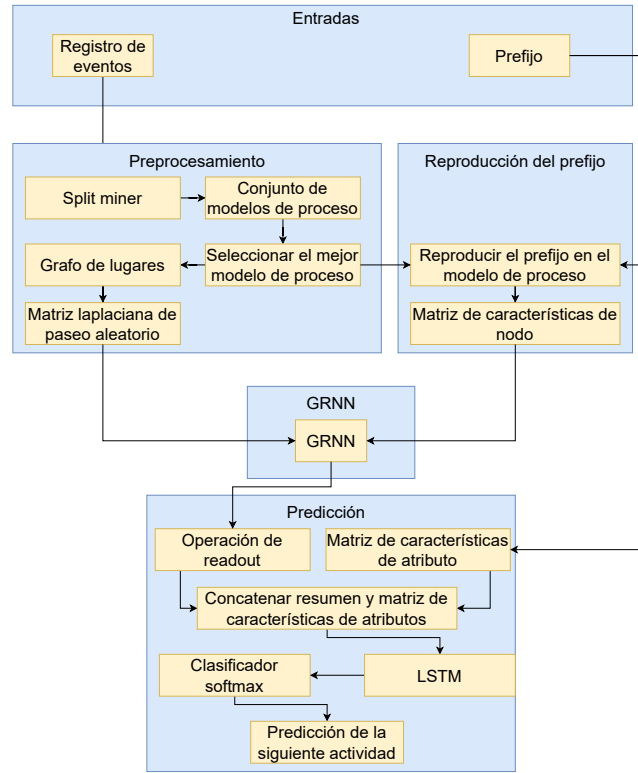


Figura 1. Diagrama del flujo de datos en PPCGR.

(LSTM), cuya salida se pasan posteriormente a un clasificador softmax para realizar la predicción de la siguiente actividad.

2.1. Codificación

En este artículo, distinguimos entre características específicas de cada nodo del grafo de lugares y características inherentes a cada evento del prefijo. Las características específicas del nodo pertenecen a una matriz de características del nodo, mientras que las características inherentes al evento pertenecen a una matriz de características de atributos.

Una matriz de características del nodo se define como una matriz de dimensiones $\mathbb{R}^{L \times |P| \times |Q|}$, donde L es la longitud máxima de traza del registro de eventos, P es el número de lugares del modelo de proceso y Q es el número de características consideradas para cada nodo. Esta matriz captura las interacciones de todos los eventos del prefijo con el modelo de proceso de tal manera que cada matriz $\mathbb{R}^{|P| \times |Q|}$ representa las interacciones de un evento con el modelo de proceso. Si el número de eventos en el prefijo es menor que L , se aplica un relleno previo (*padding*) (rellenado en el lado izquierdo de la matriz). Por otro

lado, PPCGR también utiliza una matriz de características de atributos, que codifica información sobre todo el prefijo y tiene dimensiones $\mathbb{R}^{L \times |R|}$, donde $|R|$ es el número de atributos considerados en el modelo predictivo.

La Figura 2 es un ejemplo de codificación de un prefijo en un proceso de entrega de pedidos de comida a domicilio. En este proceso, primero se realiza la recepción del pedido por un agente telefónico, después se realiza la preparación de la comida del pedido, se embala y se realiza la entrega al cliente.

En el caso de la matriz de atributos, se registra no solo la tarea que se está realizando en cada evento (F_3), sino también la persona encargada de realizar la tarea (F_4). Cabe destacar que no se registran medidas temporales directamente, sino que se aplica discretización basada en cuantiles en función de los intervalos de diferencias de tiempo obtenidos del conjunto de entrenamiento del registro de eventos, tanto para los tiempos entre eventos (F_1) como para los tiempos desde un evento hasta el principio del caso (F_2). La matriz de la Figura 2 se corresponde con la matriz resultante tras la reproducción de todo el prefijo completo, por lo que nuestro modelo predictivo utiliza toda la secuencia de matrices intermedias resultantes de la reproducción. En lo que respecta a la matriz de nodos, se registran los lugares activados (F_1) para cada transición ejecutada (F_2 y F_3) tras realizar la reproducción del prefijo sobre el modelo de proceso descubierto. Nótese que no solo se registran las transiciones observables, correspondientes a cada una de las actividades del proceso, sino también las transiciones ocultas ($HT1$ y $HT2$) que se corresponden a transiciones generadas por el algoritmo de descubrimiento de modelos para mejorar el flujo de control del modelo.

Actividades: Recepción de pedido (RP) → Preparación de pedido (PP) → Empaque y etiquetado (EE) → Entrega (EN)	Matriz de nodos $\begin{pmatrix} F_1 & F_2 & F_3 \\ P1 & RP & 0 \\ P2 & HT1 & PP \\ P3 & HT2 & EE \\ P4 & EN & 0 \end{pmatrix}$
Tiempo (discretizado) entre actividades: 1 → 3 → 2 → 1	Matriz de atributos $\begin{pmatrix} F_1 & F_2 & F_3 & F_4 \\ 1 & 1 & RP & LEON \\ 3 & 3 & PP & CLAIRE \\ 2 & 4 & EE & JILL \\ 1 & 5 & EN & CHRIS \end{pmatrix}$
Tiempo (discretizado) desde el principio: 1 → 3 → 4 → 5	
Recursos: Leon → Claire → Jill → Chris	
Transiciones: RP → HT1 → PP → HT2 → EE → EN	
Plazas: $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$	

Figura 2. Ejemplo de una matriz de características de nodos y una matriz de características de atributos.

La codificación de las matrices anteriormente mencionadas posee algunas particularidades. En el caso de la matriz de nodos, es preciso destacar que se registran todas las activaciones de la reproducción del prefijo, y no solo las últi-



mas. También se registran todas las transiciones ejecutadas entre cada par de transiciones, añadiendo tantas columnas adicionales como máximo de transiciones ejecutables entre cada par de transiciones observables del modelo de proceso. Por último, nótese que en sendas matrices no se utiliza una codificación one-hot, sino *embeddings*, ya que permiten mantener la dimensionalidad de las matrices independiente del número de atributos y actividades que haya en el registro de eventos. Esto resulta beneficioso para mantener la complejidad computacional y facilitar el entrenamiento del modelo neuronal.

3. Evaluación

3.1. Configuración experimental

Para llevar a cabo la evaluación de PPCGR nos basamos en la experimentación de [4] bajo las mismas condiciones: se realiza una validación cruzada de 5 pliegues, dividiendo cada pliegue de entrenamiento en una distribución de trazas del 80 %-20 % para obtener un conjunto de validación. El conjunto de validación se utiliza para encontrar el modelo con mejor rendimiento, en términos de la época con la precisión de validación más alta. Además, utilizamos los mismos atributos y comparamos las aproximaciones utilizando la métrica de “precisión”, como se informa en [4].

Además, también utilizamos los mismos registros de eventos, excepto “Sepsis” y “Nasa” porque no se pueden aplicar a [11] ya que no contienen los recursos que realizan los eventos. Las características de estos registros de eventos se muestran en la Tabla 1, donde la mayoría de ellos tienen una alta variabilidad tanto en las características temporales de la ejecución del proceso -duración de eventos y trazas- como en la longitud de las trazas. Hay dos modos de ejecutar el enfoque de [13], por lo que se utilizan las dos configuraciones. Además, hemos ampliado la experimentación de [4] agregando los enfoques de [14,12] probados en las mismas condiciones. Utilizamos la variante “ponderada” de [14] debido a que obtuvo los resultados más consistentes en todos los conjuntos de datos.

Configuramos nuestro modelo con los mismos hiperparámetros para cada registro de eventos. Utilizamos 256 unidades ocultas tanto para la LSTM como para GRNN y el optimizador Adam con una tasa de aprendizaje de $1e-3$ sobre un planificador de tasa de aprendizaje basado en reinicio en caliente y enfriamiento por coseno. También hemos utilizado diez categorías discretas para discretizar las características de tiempo y una dimensión de *embedding* de tamaño 32. Entrenamos durante 100 épocas utilizando un criterio de detención temprana en la precisión de validación: el entrenamiento se detiene si la precisión de validación no mejora durante 10 épocas después de alcanzar el último máximo. No se realizó una búsqueda de hiperparámetros ya que nuestra aproximación es muy resistente a una amplia variedad de hiperparámetros diferentes.

3.2. Resultados

La tabla 2 muestra los resultados de la experimentación. PPCGR obtiene el mejor resultado en 7 de los registros de eventos probados, el segundo mejor

Event log	Trazas	Actividades	Eventos	Longitud media caso	Longitud máxima caso	Duración media evento	Duración máxima caso	Duración media caso	Duración máxima caso	Variantes
BPI-2012	13087	36	262200	20.04	175	0.45	102.85	8.62	137.22	4366
BPI-2012-A	13087	10	60849	4.65	8	2.21	89.55	8.08	91.46	17
BPI-2012-C	13087	23	164506	12.57	96	0.74	30.92	8.61	91.46	4336
BPI-2012-O	5015	7	31244	6.23	30	3.28	69.93	17.18	89.55	168
BPI-2012-W	9658	19	170107	17.61	156	0.7	102.85	11.69	137.22	2621
BPI-2012-W-C	9658	6	72413	7.5	74	1.75	30.92	11.4	91.04	2263
BPI-2013-C-P	1487	7	6660	4.48	35	51.42	2254.84	178.88	2254.85	327
BPI-2013-I	7554	13	65533	8.68	123	1.57	722.25	12.08	771.35	2278
Env-permit	1434	27	8577	5.98	25	1.09	268.97	5.41	275.84	116
Helpdesk	4580	14	21348	4.66	15	11.16	59.92	40.86	59.99	226

Tabla 1. Estadísticas de los registros de eventos utilizados para la evaluación. Las medidas relacionadas con el tiempo se muestran en días.

resultado en uno y el tercero mejor en dos. En particular, las mayores diferencias con el mejor enfoque para cada registro de eventos se logran en *BPI-2013-I*, *BPI-2013-C-P* y en *BPI-2012-W*. En otros registros, como *BPI-2012*, *BPI-2012-C* y *Env-permit*, la diferencia es menos notable. Por otro lado, PPCGR tiene un rendimiento inferior en los registros de eventos *BPI-2012-A*, *BPI-2012-O* y en *BPI-2012-W-C*.

Para facilitar la interpretación de los resultados, se aplicó una técnica de análisis bayesiano basada en el modelo de Plackett-Luce [19] el cual permite comparar y clasificar diferentes aproximaciones basándose en sus respectivas probabilidades de ser el mejor en función de los datos observados. Al emplear este modelo, se pueden obtener clasificaciones de las aproximaciones de manera robusta. La figura 3 muestra los resultados del modelo bayesiano de Plackett-Luce para clasificar los algoritmos. Nuestra aproximación obtiene la mejor clasificación en general y la probabilidad más alta de ser la mejor aproximación, con una diferencia de 24,8 puntos porcentuales sobre el segundo. La figura 4 muestra los intervalos de credibilidad (cuantiles 5% y 95%) y la probabilidad esperada de ganar para cada aproximación evaluada. Nótese que si un par de aproximaciones no tienen sus intervalos superpuestos, se puede afirmar que son estadísticamente distintas desde un punto de vista de la clasificación. El intervalo de credibilidad de PPCGR no se superpone con ninguno otro, lo que muestra que nuestra propuesta es estadísticamente diferente en términos de posición en el *ranking* de los que no se superponen.

	BPI-2012	BPI-2012-A	BPI-2012-C	BPI-2012-O	BPI-2012-W	BPI-2012-W-C	BPI-2013-C-P	BPI-2013-I	Env-Permit	Helpdesk
Khan [8]	42.9	74.9	47.37	66.08	60.15	52.22	43.58	51.91	83.59	79.97
Venugopal [14]	54.69	54.88	63.75	67.87	53.9	64.8	48.44	49.62	69.58	78.7
Mauro [5]	84.66	79.76	80.06	82.74	85.98	68.64	24.94	36.67	53.59	31.79
Evermann [9]	59.33	75.82	62.38	79.42	75.37	67.53	58.83	66.78	76.19	83.66
Pasquadibisceglie [6]	83.25	74.12	74.6	78.88	81.19	68.34	47.45	46.03	86.69	83.93
Theis (w/ attr.) [13]	80.96	65.67	75.75	76.89	86.86	83.84	54.65	51.5	85.12	79.69
Theis (w/o attr.) [13]	82.89	65.5	75.26	78.38	86.22	80.06	59.48	59.41	86.29	78.77
Camargo [11]	83.28	75.98	77.93	81.35	76.4	68.95	54.67	66.68	85.78	82.93
Zararah [12]	85.31	79.87	77.43	81.15	85.19	68.19	63.29	68.94	86.06	83.96
Hinkka [7]	86.65	81.19	80.64	87.23	84.78	70.54	63.47	74.69	84.43	83.08
Tax [10]	85.46	79.53	80.38	82.29	85.35	69.79	64.01	70.09	85.71	84.19
PPCGR	87.08	79.78	80.85	82.95	88.34	74.2	67.53	77.72	87.66	85.2

Tabla 2. Precisión media de la validación cruzada de 5 pliegues. Los mejores, segundos mejores y terceros mejores enfoques se destacan en cian, naranja y amarillo, respectivamente.

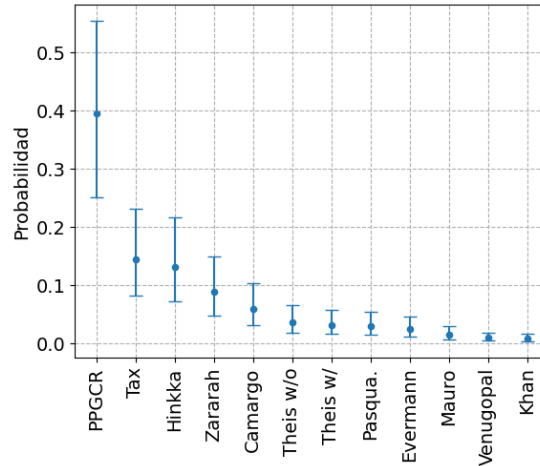


Figura 4. Intervalos de credibilidad (cuantiles 5 % y 95 %) y la probabilidad esperada de ganar para las aproximaciones evaluadas.

En general, PPCGR supera a las otras aproximaciones excepto en aquellos registros de eventos más simples como el *BPI-2012-A* o el *BPI-2012-O*. En estos casos, un modelo basado en redes recurrentes es suficiente para capturar las dependencias entre los eventos. Estos resultados confirman que la aproximación

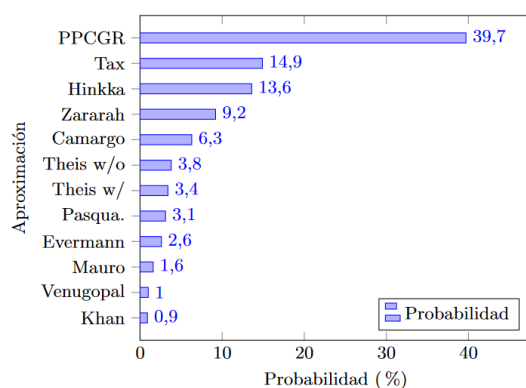


Figura 3. Probabilidades del posterior de cada aproximación.

basada en grafos es más adecuada para capturar patrones de comportamiento más complejos que generalmente no se identifican cuando el aprendizaje se realiza mediante modelos más simples como LSTMs. Esto mejora la predicción de la próxima actividad cuando hay bucles o paralelos presentes en el modelo de proceso.

4. Conclusiones y trabajo futuro

En este artículo, hemos presentado PPCGR, un enfoque novedoso que resuelve la predicción de la próxima actividad utilizando la información de los registros de eventos y del modelo del proceso. Nuestra arquitectura combina RNNs con GNNs, permitiendo el procesamiento simultáneo del estado de la red de Petri para cada evento del prefijo, lo que facilita el aprendizaje bidimensional de los prefijos en términos espaciales y temporales. Evaluamos PPCGR en 10 registros de eventos reales, comparándolo con 10 aproximaciones del estado del arte, mostrando una mayor consistencia y mejores resultados en general. Esto demuestra la adecuación de las GRNNs en la monitorización predictiva de procesos al aprovechar eficientemente la información del modelo del proceso. En trabajos futuros, exploraremos diferentes operadores de convolución de grafos, extendemos el enfoque a otros modelos de proceso y abordaremos tareas predictivas más complejas.

Referencias

1. W. M. P. van der Aalst et al., “Process mining manifesto,” in *Proceedings of the 9th International Business Process Management Workshops (BPM 2011)*, ser. Lecture Notes in Business Information Processing, vol. 99. Springer, 2011, pp. 169–194.

2. F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini, “Predictive monitoring of business processes,” in *Advanced Information Systems Engineering*. Springer International Publishing, 2014, pp. 457–472.
3. N. Tax, I. Teinmaa, and S. J. van Zelst, “An interdisciplinary comparison of sequence modeling methods for next-element prediction,” *Software and Systems Modeling*, 2020.
4. E. Rama-Maneiro, J. Vidal, and M. Lama, “Deep learning for predictive business process monitoring: Review and benchmark,” *IEEE Transactions on Services Computing*, pp. 1–1, 2022.
5. N. D. Mauro, A. Appice, and T. M. A. Basile, “Activity prediction of business process instances with inception CNN models,” in *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 348–361.
6. V. Pasquadibisceglie, A. Appice, G. Castellano, and D. Malerba, “Using convolutional neural networks for predictive process analytics,” in *2019 International Conference on Process Mining (ICPM)*. IEEE, jun 2019.
7. M. Hinkka, T. Lehto, and K. Heljanko, “Exploiting event log event attributes in RNN based prediction,” in *Communications in Computer and Information Science*. Springer International Publishing, 2019, pp. 405–416.
8. A. Khan, H. Le, K. Do, T. Tran, A. Ghose, H. Dam, and R. Sindhgatta, “Memory-augmented neural networks for predictive process analytics,” Feb. 2018.
9. J. Evermann, J.-R. Rehse, and P. Fettke, “Predicting process behaviour using deep learning,” *Decision Support Systems*, vol. 100, pp. 129–140, aug 2017.
10. N. Tax, I. Verenich, M. L. Rosa, and M. Dumas, “Predictive business process monitoring with LSTM neural networks,” in *Advanced Information Systems Engineering*. Springer International Publishing, 2017, pp. 477–492.
11. M. Camargo, M. Dumas, and O. González-Rojas, “Learning accurate LSTM models of business processes,” in *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 286–302.
12. Z. A. Bukhsh, A. Saeed, and R. M. Dijkman, “Processtransformer: Predictive business process monitoring with transformer network,” 2021.
13. J. Theis and H. Darabi, “Decay replay mining to predict next process events,” *IEEE Access*, vol. 7, pp. 119 787–119 803, 2019.
14. I. Venugopal, J. Tollich, M. Fairbank, and A. Scherp, “A comparison of deep-learning methods for analysing and predicting business processes.” IEEE, 2021.
15. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *5th International Conference on Learning Representations, ICLR 2017*, 2017.
16. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in *The Semantic Web*. Springer International Publishing, 2018, pp. 593–607.
17. L. Ruiz, F. Gama, and A. Ribeiro, “Gated graph recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 6303–6318, 2020.
18. A. Augusto, R. Conforti, M. Dumas, M. L. Rosa, and A. Polyvyanyy, “Split miner: automated discovery of accurate and simple business process models from event logs,” *Knowledge and Information Systems*, vol. 59, no. 2, pp. 251–284, may 2018.
19. B. Calvo, J. Ceberio, and J. A. Lozano, “Bayesian inference for algorithm ranking analysis,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, jul 2018.