Journal of Decision Systems

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjds20

Explainable predictive business process monitoring using gated graph neural networks

Maximilian Harl, Sven Weinzierl, Mathias Stierle & Martin Matzner

To cite this article: Maximilian Harl, Sven Weinzierl, Mathias Stierle & Martin Matzner (2020) Explainable predictive business process monitoring using gated graph neural networks, Journal of Decision Systems, 29:sup1, 312-327, DOI: 10.1080/12460125.2020.1780780

To link to this article: https://doi.org/10.1080/12460125.2020.1780780

Published online: 23 Jun 2020.

JOURNAL OF DECISION SYSTEMS
2020, VOL. 29, NO. S1, 312–327
https://doi.org/10.1080/12460125.2020.1780780

ARTICLE

Explainable predictive business process monitoring using gated graph neural networks

Maximilian Harl, Sven Weinzierl, Mathias Stierle and Martin Matzner

Institute of Information Systems, Friedrich-Alexander-Universität Erlangen-Nürnberg, Nuremberg, Germany

ABSTRACT

Predictive business process monitoring (PBPM) is a class of techniques designed to forecast future behaviour of a running process instance or the value of process-related metrics like times and frequencies. PBPM systems support process workers and process managers in making operational decisions. State-of-the-art PBPM systems apply deep-learning techniques with multiple hidden layers to infer from data which makes it difficult for system users to understand why a prediction was made. However, the user needs to see deeper causes to identify intervention mechanisms that secure process performance. The main contribution of this paper is a technique that makes a prediction more explainable by visualising how much the different activities included in a process impacted the prediction. This work is the first to use gated graph neural networks (GGNNs) to make decisions more explainable, and it's also GGNN's first application to PBPM. We use a process event data set to demonstrate our approach.

ARTICLE HISTORY
Received 23 January 2020
Accepted 5 June 2020

KEYWORDS
Predictive business process monitoring; explainable artificial intelligence; gated graph neural networks; decision support system; deep learning

1. Introduction

Business process management (BPM) discusses decision support systems (DSSs) as a tool that helps decreasing process stakeholders' uncertainties both when driving process improvement on the whole and when reaching decision points during process execution. Business process analysis and monitoring are vital elements of the business process lifecycle (Dumas et al., 2013; Weske et al., 2004) for managing risks by providing transparency (zur Muehlen & Ho, 2006).

Process mining (PM) is a data-driven technology for the analysis of business processes (van der Aalst, 2016). Its initial focus was on discovering process models from historical data, i.e. event logs. Given the shift from descriptive to predictive analytics (Grigori et al., 2004), predictive business process monitoring (PBPM) has emerged as a novel research stream within the BPM domain (Maggi et al., 2014). PBPM deals with predicting a process's future behaviour (e.g. next activities) or the value of process-related performance indicators (e.g. time-, outcome-, workload- or resource-based). To decrease uncertainty in decision-making and, subsequently, the risk of failing the desired process goals (e.g. cost, timeliness or the quality of outputs (Jallow et al., 2007)), high predictive quality is required for PBPM techniques (Pika et al., 2016). As in many other domains (e.g. speech

CONTACT Sven Weinzierl sven.weinzierl@fau.de

recognition, visual object recognition, and object detection (LeCun et al., 2015)), deep-learning (DL) techniques have proven to deliver higher predictive quality than traditional (e.g. probabilistic (Evermann et al., 2017) or transition-system-based (Navarin et al., 2017)) approaches.

DL techniques, however, are considered as *black-box* models for which it is difficult to determine "whether the right decision was made for the right reason" (O'leary, 1988, p. 77).

The DSS research community for long has pointed out the importance of "support-[ing] and enhanc[ing] human judgements" (Csáki et al., 2016, pp. 198-199) for which it is crucial that a DSS considers not only technical aspects such as the predictive quality but also social aspects such as explainability of the system (Phillips-Wren et al., 2015; Power et al., 2019a). In their study, Nunes and Jannach (2017) show that the research field of explanations in decision support and recommender systems has a long history with an increasing trend in the last years. The very recent special issue in *Operational Research* emphasises the on-going importance of "comprehensible visualisation approach to present relevant information to users and decision-makers in an effective way" (Burnay et al., 2019, p. 853). Especially for a DSS that supports high-risk decisions (e.g. life or death decisions in healthcare), explainability needs to be a key design criterion (Power et al., 2019b).

To illustrate the importance of explainability in predictions, let us consider the scenario of a bank that is handling credit loan applications, as depicted in Figure 1. Without the support of a predictive DSS, a process worker handles the loan application based on his own analysis of the application, such as the requested loan amount or the profession of the applicant. By introducing a PBPM technique, the decision of the process worker could be supported by a prediction whether the loan is likely to be

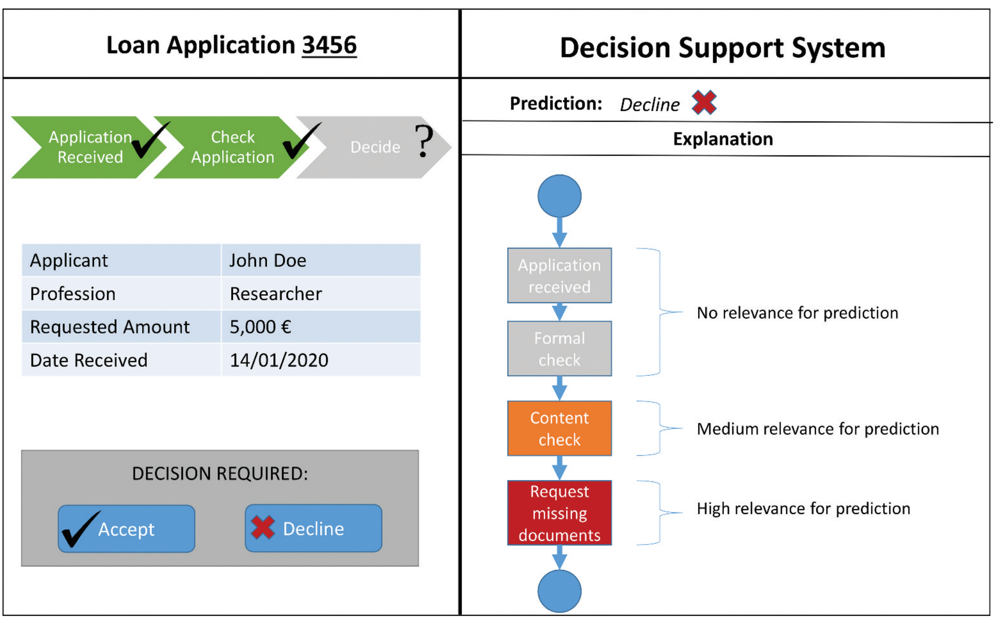

Figure 1. Example of a DSS by using a GGNN-based technique for explainable outcome predictions.

accepted or declined. The first hurdle for a successful implementation of such a DSS might be the missing trust of the process worker in the prediction as she/he cannot explain it. However, the even more serious issue would be that she/he cannot act upon the prediction if required. For instance, if the DSS foreshadows the decline of the application, the process worker would need to know whether this is simply due to missing documents or because severe doubts exist regarding the credibility of the customer that were identified when checking the application. For the example shown in Figure 1, the process worker can explain the prediction of a declined loan by the fact that the last step in the process was to request missing documents from the applicant. With this insight, the process worker can now intervene. A possible intervention could be calling the applicant to make him aware of the importance of providing the missing documents.

In the field of PBPM, Breuker et al. (2016) present a technique for next activity prediction with the design objective of *comprehensibility* based on a probabilistic *white-box* approach. However, Márquez-Chamorro et al. (2017) point out that in PBPM "little attention has been given to providing recommendations and explaining the prediction values to the users". Against this background, we identified a research gap in the PBPM literature as a lack of works which deal with explainable business process predictions, especially when using a DL (i.e. black-box) approach.

In our work, we try to overcome the insufficient explainability of DL approaches in PBPM with our *research goal*: providing a gated-graph-neural-networks (GGNNs)-based technique for explainable process outcome predictions. We adopt a design-oriented research methodology to address this goal.

With our technique, we exploit the graph-based layout of GGNNs (Li et al., 2015) to represent the business process data to retrieve a relevance score for each activity of a process instance regarding its impact on the outcome prediction. Similar to van Eck et al. (2019), we subsequently visualise the relevance of the single activities by colouring the related elements of a process model, which is a representation that is familiar to most process stakeholders (Janiesch & Matzner, 2019).

Ultimately, our technique serves as a basis for a DSS for process stakeholders. Such a DSS is likely to decrease uncertainty in decision-making and to foster adoption by process stakeholders striving to make sound decisions under pressure (Nunes & Jannach, 2017).

2. Research method

Our research follows the design-science-research (DSR) paradigm, as proposed by Gregor and Hevner (2013), to design a DSS similar to the approach used by Rybnytska et al. (2018). We address and solve a real-world problem by designing a socio-technical artefact and thus generate new knowledge (Gregor & Hevner, 2013).

As DSR is inherently a problem-solving process, we review the descriptive and prescriptive knowledge in this area (Hevner & March, 2004) by conducting a literature review. As no research has suggested an explainable DL technique for the prediction of business process outcomes so far, our goal is to fill this gap by designing a GGNN-based technique. To ensure the rigour and relevance of our research (Hevner & March, 2004), our work is rooted in the existing knowledge base of BPM and DSS.

The artefact is instantiated, evaluated, and demonstrated. In the evaluation phase, we assess its ability to solve the previously identified problem of explainable business process outcome prediction (Hevner & March, 2004). We evaluate our technique by applying it to a real-life data set from a Dutch financial institute (van Dongen, 2017) to assess our technique's efficacy (Peffers et al., 2012), which in our case is the feasibility of explaining business process outcome predictions.

3. Background

BPM is "a body of methods, techniques and tools to discover, analyse, redesign, execute and monitor business processes" (Dumas et al., 2013, p. 5) to improve business processes and assure consistent outcomes. Decision support is essential for various process stakeholders throughout the BPM life-cycle. van der Aalst (2008) defines four types of decisions in operational processes. Design-time, configuration-time and control-time decisions have to be made by process owners, management and system engineers (Dumas et al., 2013) to improve the process as a whole. Run-time decisions, in contrast, refer to single process instance executions and are important for their efficient and timely handling by process participants.

Various streams of research have tackled decision support in business processes such as business activity monitoring (BAM), business operations management (BOM) or business process intelligence (BPI) (van der Aalst, 2008). Finally, PM has been the established term for the data-driven analysis of business processes from event logs (van der Aalst, 2004). Early works in the field of PM were mainly focused on the ex-post analysis of the process to deliver insights for holistic process improvement to support design-time, configuration-time and control-time decisions. Recently, an increasing amount of work is done on ex-ante applications of PM. PBPM aims "to provide timely information that enables proactive and corrective actions to improve process performance and mitigate risks" (Márquez-Chamorro et al., 2017) and hence is mostly devoted to the support of run-time decisions.

In their survey, Márquez-Chamorro et al. (2017) present a large variety of methods for building predictive methods in the domain of PBPM, ranging from more traditional statistical and data-mining methods to state-of-the-art DL architectures. While most works benchmark their methods regarding predictive quality with measures such as *Accuracy*, some researchers propose explainability of predictions as an essential design and evaluation criteria (Breuker et al., 2016; Verenich et al., 2019) following the emerging research field of XAI (Adadi & Berrada, 2018).

XAI can mainly be split into ante-hoc and post-hoc explanation methods (Du et al., 2019). Post-hoc explanation methods extract explanations and visualisations from a learned model, i.e. they do not explain the entire model. In contrast, ante-hoc explanation methods create an inherent explainability through their architecture. The work of Breuker et al. (2016) is an example of ante-hoc explainability in the field of PBPM. Verenich et al. (2019) present a first attempt based on a DL approach for post-hoc explainability. However, explainable PBPM, especially with deep neural networks, is still mostly unexplored.

4. Gated graph neural networks for explainable outcome prediction in business processes

Intending to achieve explainable process predictions, we propose using process models as a basis for the underlying prediction model. In search for a suitable DL architecture, we found graph neural networks (GNNs), as proposed by Scarselli et al. (2008), which were developed for domains where input data has a graph structure, such as in chemistry, molecular biology or business processes. In particular, we use an extension of GNNs, which are gated graph neural networks (GGNNs) that were shown to be more efficient through the use of gated recurrent units (GRUs) (Cho et al., 2014) to unroll the recurrence over a fixed number of time steps and to apply the backpropagation through time algorithm to compute gradients (Li et al., 2015). By using a GGNN, we can provide explainability of single predictions (post-hoc) as the model provides relevance scores for each node of the neural network, i.e. for each step in the process model. To a certain degree, we also provide ante-hoc explainability as the architecture of the model partially corresponds to the process model.

Our GGNN-based technique for explainable outcome predictions consists of the three steps: (1) *event log to GGNN input transformation*, (2) *GGNN-based prediction model creation* and (3) *single graph instance visualisation*, as seen in Figure 2. We transfer the approach by Li et al. (2015) from the domain of molecular chemistry to the area of PBPM to achieve explainable outcome predictions. Hence, the main contribution of our work is to exchange molecular structures by process models, which is a challenge in regards to data input (1), and data output visualisation (3) as well as using the relevance scores of the prediction model (2) for explaining the predictions.

In step (1), the event log is loaded and the process instances of the event log are transformed into process instance graphs because a GGNN requires input data with a graph-oriented representation. Next (2), the GGNN-based prediction model is created based on the graph instances received as an input from the previous step. The prediction model outputs an outcome prediction for each graph instance and relevance scores on activity-level with respect to the outcome prediction. Finally (3), given the learned prediction model from step (2) and a new process instance from step (1), a visualisation in the

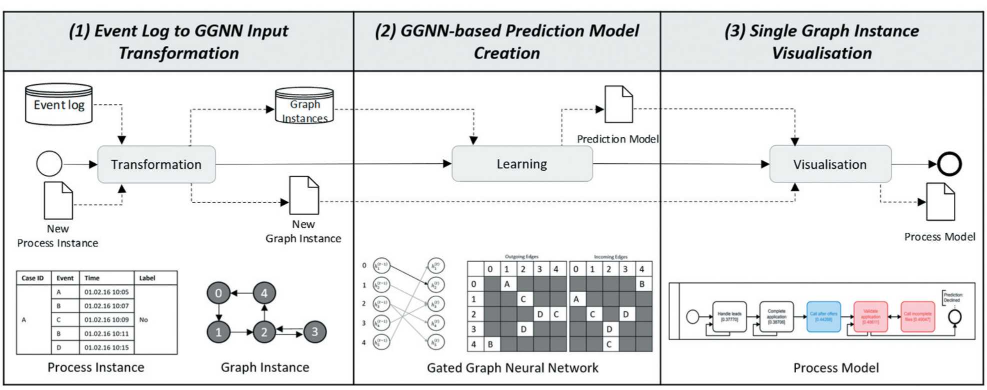

Figure 2. Steps of our GGNN-based technique for explainable outcome predictions.

form of a process model is created, where each activity is coloured depending on its relevance score. In the following, we describe each step of the technique in detail.

4.1. Event log to GGNN Input transformation

The technique transforms the event log's data representation into a format that is geared to the later use by the GGNN. The transformation process consists of the four steps: (1) load the event log, (2) map the event log's process instances to graph instances, (3) add "Start/End" node and (4) encode the numerical values, as depicted in Figure 3.

In the beginning, the technique loads an event log that we define as a multi-set of process instances. A process instance represents a concrete execution of a business process and can be depicted as a sequence of events. An event is a unique tuple consisting of an activity, process instance id, time stamp and one or more additional context attributes. The activity to which an event is assigned represents the type of the event. Further, we assume that an event has a reference to a process instance. In process mining, this assumption has been reported to be typically satisfied in real-world event logs (van der Aalst, 2016). Moreover, an event has assigned a timestamp, and the events within a process instance are ordered by the timestamp. The last element of the event tuple includes context attributes characterising the environment of a process instance (da Cunha Mattos et al., 2014). After introducing the term event, it should be mention that the number of context attributes associated with an event and the number of events per process instance can vary. In the following, we use the process instance, as depicted in Table 1 – which is a credit loan application – as an example to describe our technique. The context attribute *loan accepted* (with its values "Yes" or "No") is assigned to the process instance, which we will consider as the label for the prediction model.

In the second step, the technique maps each process instance of the loaded event log to a graph instance. van Dongen and van der Aalst (2004) propose a method to map a sequence of events (process instance) to a directed graph of events. This method is insufficient for our purpose since we are interested in the relevance of individual activities on the prediction outcome. Therefore, we consider a node of the graph as an activity (not as an event) and map a sequence of events to a graph of activities. After this mapping

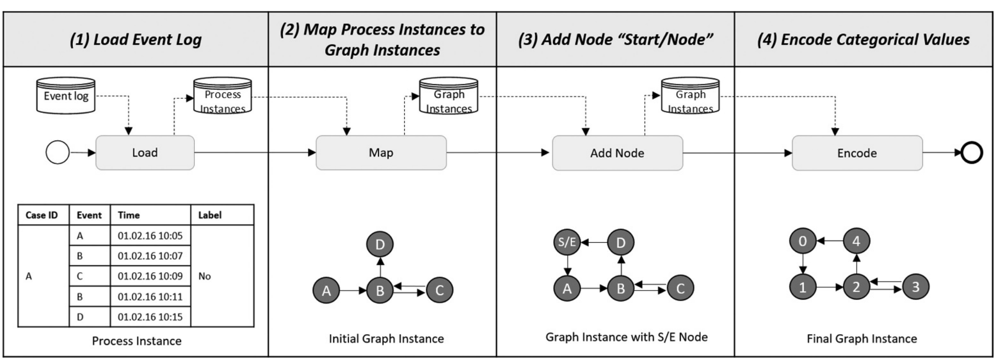

Figure 3. Steps of the event log to GGNN input transformation.

Table 1. Example of an event log with a single process instance.

Case ID	Event	Time	Loan accepted
Application_1054	Handle leads	01.02.16 10:05	No
	Handle leads	01.02.16 10:06	
	Complete application	01.02.16 10:06	
	Complete application	01.04.16 18:23	
	Call after offers	01.04.16 18:23	
	Validate application	25.01.16 14:37	
	Call incomplete files	26.01.16 09:42	
	Validate application	27.01.16 11:57	
	Validate application	28.01.16 10:18	
	Call incomplete files	28.01.16 10:18	
	Validation application	28.01.16 12:56	
	Validation application	28.01.16 13:18	

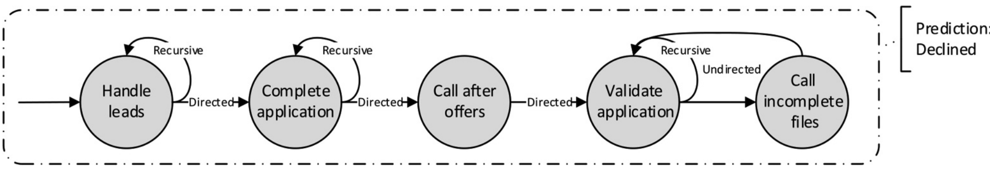

Figure 4. Initial graph instance.

step, the process instance from Table 1 is represented, as shown in Figure 4 with the three different edge types: undirected, directed and recursive.

Further, the technique adds a “Start/End” node to each graph instance since the GGNN requires inputs with a start and an endpoint for the learning phase. The “Start/End” node is connected via an edge of the type start (end) to the first node (last node) of a graph instance. Figure 5 depicts the graph instance with the added “Start/End” node.

Finally, the technique applies a numerical encoding of the categorical values. A GGNN, as used in our technique, requires numerical values to perform forward- and backward propagation. We use an ordinal encoding for the node labels (names); the “Start/End” node gets the value 0 and each other node of a graph instance gets a value between 1 and the number of nodes of the graph instance. As well, the edge label values are replaced by an integer value.

4.2. GGNN-based prediction model creation

For the creation of the prediction model, we use gated graph neural networks (GGNNs), as proposed by Li et al. (2015). The GGNN receives inputs with a graph-oriented

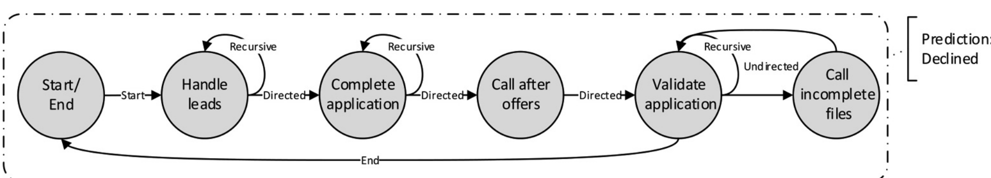

Figure 5. Graph instance with “Start/End” node.

representation, which is why we have to transform process data to graph instances in beforehand. Regarding outcome prediction in PBPM, RNNs can learn predictive models with a high predictive quality (Hinkka et al., 2018). However, state-of-the-art RNNs require sequential data with a vector-oriented representation and do not grasp the possibly existing graph-oriented structure of the underlying (process) data (Scarselli et al., 2008). We use GGNNs to make use of the graph-oriented structure of process data to provide explainable outcome predictions. The GGNN prediction model that maps graph instances to outputs consists of two sub-models – a propagation model and an output model. The propagation model computes the node representations until convergence, depending on the time. Figure 6 visualise the underlying mechanism of how the propagation model of GGNNs deals with a graph instance at one time step.

In the beginning, a graph instance (1) is fed to the propagation model of the GGNN. At a time step, the representation of each node of the graph is calculated based on the representation of the predecessor nodes of the same graph (2). Each node depicts an activity. So, as exemplarily depicted in Figure 6, the representation of activity 1 is calculated based on the representation of activity 0. Note, in a real setting, this process is not limited to one single time step. The connections between the activities are represented by a sparsity matrix (3). The sparsity matrix stores for each node the information about the outgoing as well as the incoming edges. Among other elements (e.g. node labels), the outgoing and incoming edges determine the representation of an activity.

For each activity, the output model maps the activity representation to the output. The output consists of a probability distribution, in which the probability with the highest value represents the prediction, and a relevance score distribution. A relevance score of the distribution refers to the relevance of an activity of a graph instance for the predicted outcome.

4.3. Single graph instance visualisation

Finally, the technique creates based on the relevance score distribution, which is retrieved by applying the prediction model for a new graph instance, a visualisation in the form of a process model, as presented in Figure 7.

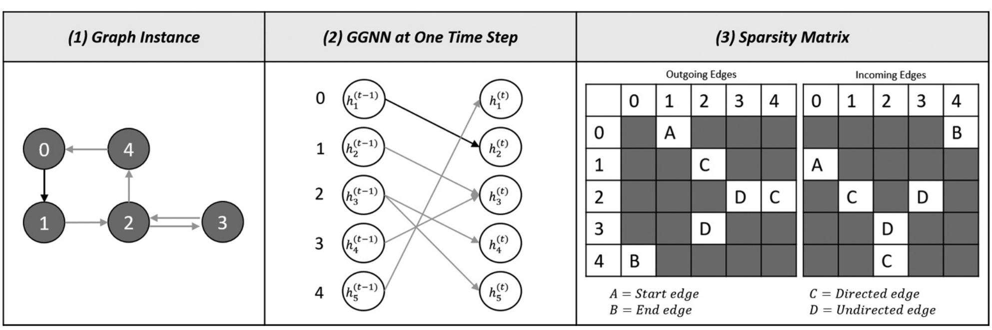

Figure 6. Basic mechanism how GGNNs deal with graph instances at one time step (adapted from Li et al. (2015)).

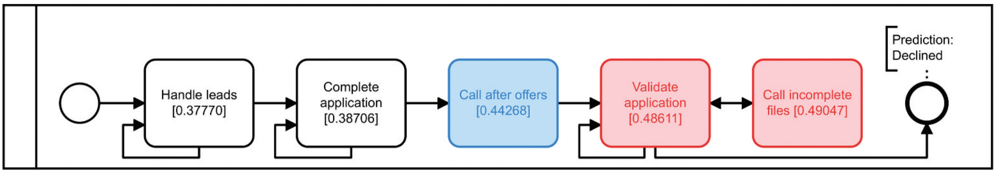

Figure 7. Process model for explainable outcome predictions.

The process model has a start and an end activity and, therefore, not only a single element that represents the start and the end position like the graph instance (see Figure 5). Further, each activity of the process model is coloured depending on its relevance score. If the colour of the activity is black, the relevance score referring to the predicted outcome is low. A red-coloured activity depicts a high relevance score, whereas a blue-coloured activity means an average relevance score. Ultimately, to convey a more profound understanding, the calculated relevance scores are presented for each activity of the process model.

5. Evaluation

In this section, we demonstrate the usefulness of our technique. For that, we present the used data, the technical setup, our results and a discussion.

5.1. Data

For the evaluation, we used a data set from a Dutch financial institute describing their loan application process. The business process consists of 8 distinct steps that are executed by human workers, as seen in Figure 8: *Handle leads, Complete application, Call after offer, Validate application, Call incomplete files, Asses potential fraud, Personal loan collection* and *Shortened completion*.

In this work, we use a sub-set of the event log attributes, which includes: *case ID, event ID, timestamp* and *acceptance status*. The *case ID* is the identifier for one process instance and *event ID* the identifier for a certain event. The attribute *timestamp* defines when an event occurs, and the attribute *acceptance status* has either the value 0 or 1, depending on the loan being granted or not. The attribute *acceptance status* is also the classification target. Therefore, only control-flow attributes of the event log are used for the classification.

5.2. Technical setup

To train the model, 27,490 graph instances were generated from the described event log. These were then class balanced so that the GGNN can learn from equal amounts of 0 and 1 labelled graph instances that signify that an application was rejected or accepted respectively. This class balancing step reduced the overall number of input graphs to 16,544. The problem of *imbalanced classes* is well-known in the data-mining literature and it is common

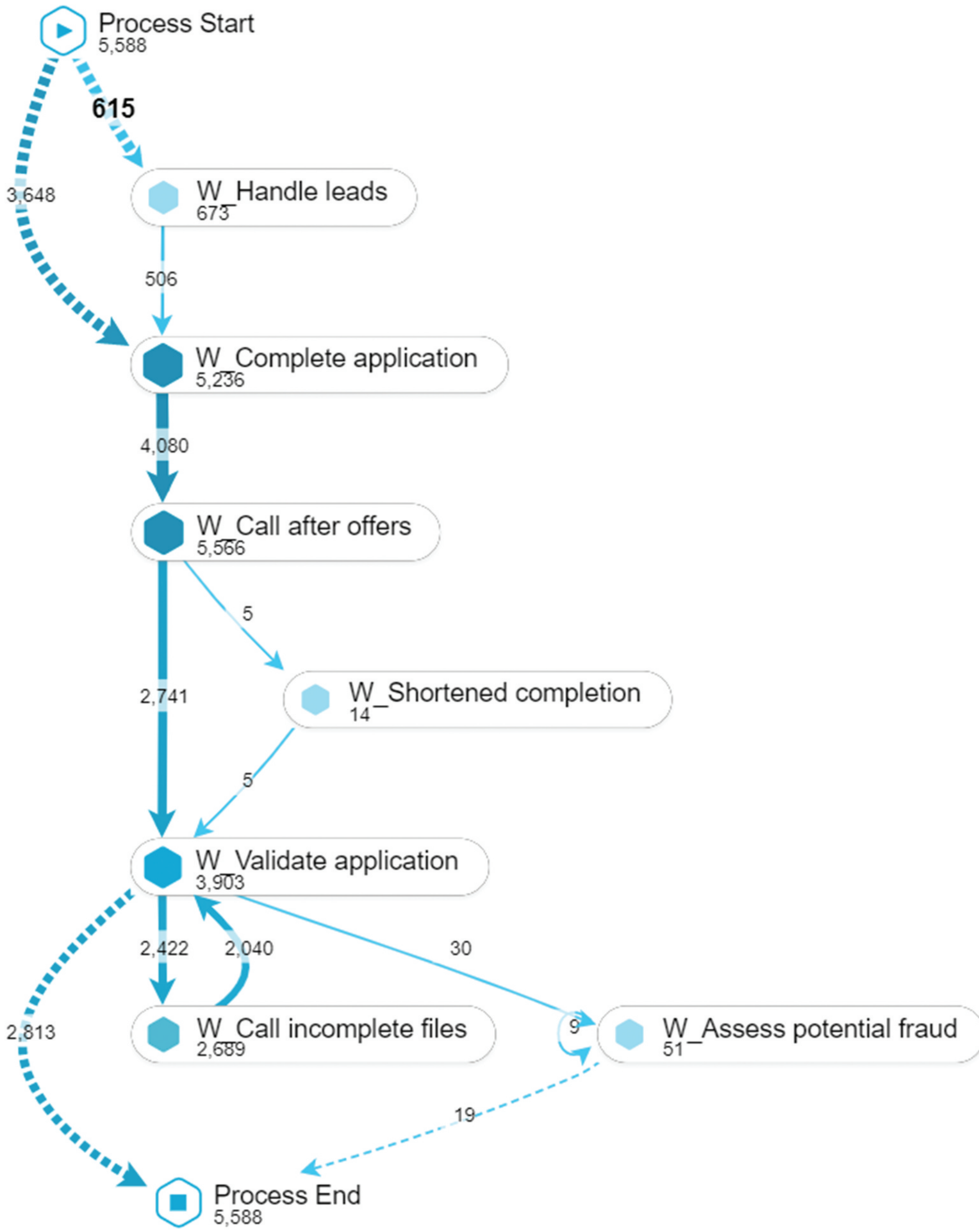

Figure 8. Process model of loan application process created with Celonis[1].

practice to deal with it (López et al., 2013). Afterwards, we applied a 90/10 split validation for the remaining graph instances.

Considering hyperparameters, the model was trained with a maximum number of epochs of 1,000 and a patience of 250. We used the *Adadelta* (Zeiler, 2012) optimisation algorithm with a starting learning rate of 1.0. The size of the hidden layers of the neural network was set to 200 neurons with five hidden layers and two time steps on the first, second and fourth layer, as recommended by Li et al. (2015).

Finally, we implemented our technique in Python[2]. We used the DL framework TensorFlow[3] to facilitate the process of building DL models. The model was developed based on the implementation of Li et al. (2015). The graphs of the artefact were generated using Graphviz[4]. The instantiation of the artefact and the data can be found at GitHub[5]. We conducted all experiments on a workstation with 12 CPUs, 128 GB RAM and a single GPU RTX 6000.

5.3. Results

Under the previously described setup, we reached an *Accuracy* (number of correct classified graph instances divided by the total number of classifications) of 77.00% after 548 epochs. Given that the data set was balanced to a 50:50 ratio of accepted and declined loans, the results of our technique are patently better than a random guess. Li et al. (2015) show that the predictive quality of the GGNN typically increases with the amount of data used for training. As such, we can expect our approach to even improve in practical scenarios where process data is usually present in larger amounts. While a reasonable predictive quality presents a crucial foundation for our technique, this work is not primarily concerned with the *Accuracy* of the underlying model but rather with its application to the PBPM domain to obtain an explainable outcome prediction. So, the achieved *Accuracy* is acceptable for our purpose since it is in line with that of other contemporary approaches in the domain of PBPM (Evermann et al., 2017; Tax et al., 2017).

To better understand how our DSS could be used in an organisation and why the explainability of the prediction is required for process stakeholders, we describe an exemplary scenario at the Dutch financial institute that supplied the data set. The business process at hand deals with loan applications submitted by customers through an online system. The business objective of the process is to increase the number of successful loan applications (i.e. the loan is issued) while minimising the risk of default (i.e. the loan is not paid back). To proactively manage the business success, in particular, predictions of declined loans are of interest to intervene in the process. For a precise intervention, the employee needs to understand which information triggers the system to reject a loan probably. In the following, we present two exemplary loan applications from the data set, and we demonstrate how the explanation delivered by our technique supports the employee in decision making.

The application (see Figure 9) for the loan was completed and the details were discussed over the phone afterwards. The application was iteratively improved (this explains the recursive edge on the activity *Complete application*). Since the completion of the application and the call after offers have a relatively low relevance, they do not be

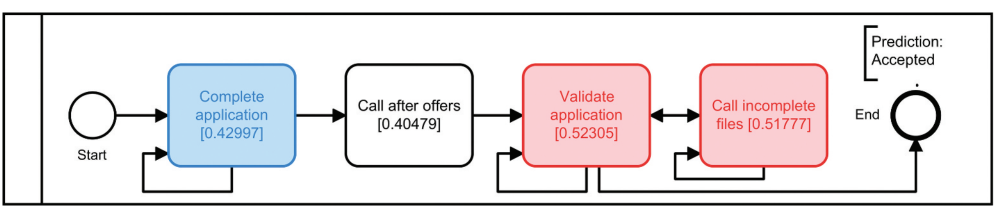

Figure 9. Approved loan application.

weighted heavily in the further reasoning. After the discussion on the phone, where the previous part of the application was possibly validated, additional files were required. This step had to be repeated multiple times because the financial institute, e.g. needed further proof for the securities and an event *Validate application* was generated for every file (this again explains the recursive edges on the activities *Validate application* and *Call incomplete files*). Both of these steps show high relevance, which is obvious since the absence of one corresponding event for the other would likely lead to a denial of the application. After the last required files were validated, the application process ended on the activity *Validate application* and the loan was approved.

After, as seen in Figure 10, the lead has been repeatedly handled (recursive edge), i.e. possibly multiple leads were investigated, and the application has been completed, no incomplete files were found and no application validation took place. The first two steps in themselves hold a very low relevance because all other steps in the process still need to be conducted, which decide on approval or denial. The loan might, therefore, have been rejected, because necessary additional files that are required after lead handling were not provided or refused to be provided in the call after offers. Thus, the *Call after offers* activity bears a high relevance, because e.g. no validation of the application follows afterwards. To now still bring the loan application to a success, an analyst could intervene by procuring the probably necessary documents for the completion of the loan application.

5.4. Discussion

Attributed to the fact that the used data set was balanced to a 50:50 ratio of accepted and declined loans, the quality of our technique for the outcome prediction is evidently better than a random guess. Additionally, Li et al. (2015) show that the predictive quality of the GGNN typically increases with the amount of data used for training. So, we expect a further improvement in practical scenarios where process data could be available in much larger sizes (van der Aalst, 2015).

Even though a reasonable predictive quality presents a crucial prerequisite for the utilisation of our technique, the focus of our work is on the explainability of the predictions. Moreover, we specialise in the relevance of specific activities for the classification outcome. Concurrently, we create a certain degree of explainability for the underlying graph-oriented DL model. Each process activity is internally handled as a graph node, which is subsequently mapped to a relevance score and the whole graph to the classification outcome (i.e. the process outcome). On the way from input to output, the process data is run, among others, through multiple GRU cells organised in layers, leading to

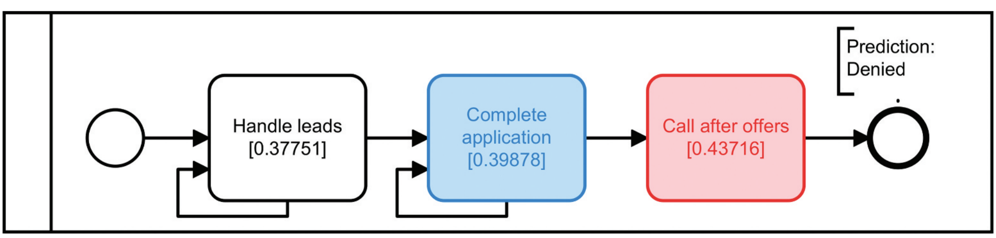

Figure 10. Denied loan application.

a blurring of the original process information. Therefore, the mapping between the input and the output is hard to comprehend by just looking at the parameters of the model, which is why extracting relevance scores from the model for the individual process activities, visualised in a process model, leads to higher interpretability.

6. Conclusion

The research presented in this paper focuses on the usage of DSS for risk management in business processes by decreasing uncertainty for process stakeholders. We fill the research gap of explainable business process outcome prediction using a DL approach. The presented technique is based on GGNNs that are most suitable for representing the sequential flow of a business process.

The idea of this paper was to provide a basis for a decision support system for process stakeholders that explains business process predictions with a familiar representation, i.e. process models. The instantiation of the technique was demonstrated and evaluated with a real-life data set exhibiting the technique's potential for explaining the predictions made by the neural network. Thereby, considering the whole of this work, we showed the fulfilment of our research goal, primarily through the provision of an easily accessible and understandable visualisation of the most relevant activities in a business process.

Our work did not investigate the impact of wrongly classified process outcomes towards misleading explainability. Du et al. (2019) point out the importance of assessing the faithfulness of an explanation, i.e. if a user can trust an explanation, which should be considered in future works. Furthermore, the evaluation was executed with a single data set which limits conclusions on the generalisability of our artefact.

Mainly, there are two different directions for future research. First, the prediction model currently only relies on the information contained in the control-flow of the process instance. The process outcome, however, is undoubtedly also influenced by contextual attributes (Fernandez et al., 2019) of the process instance (Márquez-Chamorro et al., 2017) such as the loan amount (see Figure 1) (Wang et al., 2019) or the resources involved (Heinrich et al., 2018). Second, Heavin and Power (2018) state the importance of DSS to support decision-makers with operational tasks but also tactical and strategic tasks. The majority of techniques in the PBPM domain focus on the prediction on process instance level to support operational process tasks. However, the prediction of business processes on a process model level would better support process managers (Park & Song, 2019). Therefore, we plan to extend the presented technique to visualise the relevance of process steps not only for single process instances but to aggregate the resulting relevance scores to provide a holistic picture of the process in regards to a defined process outcome.

Notes

1. https://www.celonis.com
2. www.python.org
3. www.tensorflow.org
4. www.graphviz.org

5. https://github.com/fau-is/xop

Acknowledgments

This project is funded by the German Federal Ministry of Education and Research (BMBF) within the framework programme *Software Campus* (https://softwarecampus.de) under the number 01IS17045.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Maximilian Harl http://orcid.org/0000-0001-7522-5296
Sven Weinzierl http://orcid.org/0000-0003-2268-7352
Mathias Stierle http://orcid.org/0000-0001-5938-1125
Martin Matzner http://orcid.org/0000-0001-5244-3928

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, (6), 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

van Dongen, B.F. (2017). BPI challenge 2017. https://doi.org/doi:10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b

Breuker, D., Matzner, M., Delfmann, P., & Becker, J. (2016). Comprehensible predictive models for business processes. *MIS Quarterly*, *40*(4), 1009–1034. https://doi.org/10.25300/MISQ/2016/40.4.10

Burnay, C., Dargam, F., & Zarate, P. (2019). Special issue: Data visualization for decision making: An important issue. *Operational Research*, *19*(4), 853–855. https://doi.org/10.1007/s12351-019-00530-z

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint*. *arXiv:1406.1078* .

Csáki, C., Meredith, R., O'Donnell, P., & Adam, F. (2016). Understanding the scientific contribution of an international community of researchers: The case of the IFIP Wg 8.3 Conferences on DSS (1982–2014). *Journal of Decision Systems*, *25*(sup1), 198–215. https://doi.org/10.1080/12460125.2016.1187426

da Cunha Mattos, T., Santoro, F., Revoredo, K., & Nunes, V. (2014). A formal representation for context-aware business processes. *Computers in Industry*, *65*(8), 1193–1214. https://doi.org/10.1016/j.compind.2014.07.005

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, *63*(1), 68–77. https://doi.org/10.1145/3359786

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H.A. (2013). *Fundamentals of business process management* (1st ed.). Springer.

Evermann, J., Rehse, J.-R., & Fettke, P. (2017). Predicting process behaviour using deep learning. *Decision Support Systems*, *100*, 129–140. https://doi.org/10.1016/j.dss.2017.04.003

Fernandez, C., Provost, F., & Han, X. (2019). Counterfactual explanations for data-driven decisions. In *Proceedings of the 40th International Conference on Information Systems (ICIS2019)*. AISeL.

Gregor, S., & Hevner, A.R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), 337–355. https://doi.org/10.25300/MISQ/2013/37.2.01

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., & Shan, M.-C. (2004). Business process intelligence. *Computers in Industry*, *53*(3), 321–343. https://doi.org/10.1016/j.compind.2003.10.007

Heavin, C., & Power, D.J. (2018). Challenges for digital transformation – Towards a conceptual decision support guide for managers. *Journal of Decision Systems*, *27*(sup1), 38–45. https://doi.org/10.1080/12460125.2018.1468697

Heinrich, B., Schiller, A., & Schön, D. (2018). The cooperation of multiple actors within process models: An automated planning approach. *Journal of Decision Systems*, *27*(4), 238–274. https://doi.org/10.1080/12460125.2019.1600894

Hevner, A., & March, S.T. (2004). Design science research in information systems. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Hinkka, M., Lehto, T., Heljanko, K., & Jung, A. (2018). Classifying process instances using recurrent neural networks. In *Proceedings of the 16th International Conference on Business Process Management (BPM2018)* (pp. 313–324). Springer.

Jallow, A., Majeed, B., Vergidis, K., Tiwari, A., & Roy, R. (2007). Operational risk analysis in business processes. *BT Technology Journal*, *25*(1), 168–177. https://doi.org/10.1007/s10550-007-0018-4

Janiesch, C., & Matzner, M. (2019). BAMN: A modeling method for business activity monitoring systems. *Journal of Decision Systems*, *28*(3), 185–223. https://doi.org/10.1080/12460125.2019.1631682

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436. https://doi.org/10.1038/nature14539

Li, Y., Tarlow, D., Brockschmidt, M., & Zemel, R. (2015). Gated graph sequence neural networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR2016)*. Citeseer.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, *250*, 113–141. https://doi.org/10.1016/j.ins.2013.07.007

Maggi, F.M., Di Francescomarino, C., Dumas, M., & Ghidini, C. (2014). Predictive monitoring of business processes. In *Proceedings of the 26th International Conference on Advanced Information Systems Engineering (CAiSE2014)* (pp. 457–472). Springer.

Márquez-Chamorro, A.E., Resinas, M., & Ruiz-Cortés, A. (2017). Predictive monitoring of business processes: A survey. *IEEE Transactions on Services Computing*, *11*(6), 962–977. https://doi.org/10.1109/TSC.2017.2772256

zur Muehlen, M., & Ho, D.T.-Y. (2006). Risk management in the BPM lifecycle. In *Proceedings of the 4th International Conference on Business Process Management (BPM2004)* (pp. 454–466). Springer.

Navarin, N., Vincenzi, B., Polato, M., & Sperduti, A. (2017). LSTM networks for data-aware remaining time prediction of business process instances. In *Proceedings of the Symposium Series on Computational Intelligence (SSCI2017)* (pp. 1–7). IEEE.

Nunes, I., & Jannach, D. (2017). A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-adapted Interaction*, *27*(3), 393–444. https://doi.org/10.1007/s11257-017-9195-0

O'leary, D.E. (1988). Methods of validating expert systems. *Interfaces*, *18*(6), 72–79. https://doi.org/10.1287/inte.18.6.72

Park, G., & Song, M. (2019). *Predicting performances in business processes using deep neural networks*. *Decision Support Systems*, *129*, 113191. https://doi.org/10.1016/j.dss.2019.113191

Peffers, K., Rothenberger, M., Tuunanen, T., & Vaezi, R. (2012). Design science research evaluation. In *Proceedings of the 7th International Conference on Design Science Research in Information Systems (DESRIST2012)* (pp. 398–410). Springer.

Phillips-Wren, G., Iyer, L.S., Kulkarni, U., & Ariyachandra, T. (2015). Business analytics in the context of big data: A roadmap for research. *Communications of the Association for Information Systems*, *37(23)*, 448–472. https://doi.org/10.17705/1CAIS.03723

Pika, A., van der Aalst, W.M.P., Wynn, M.T., Fidge, C.J., & ter Hofstede, A.H.M. (2016). Evaluating and predicting overall process risk using event logs. *Information Sciences*, *352-353*, 98–120. https://doi.org/10.1016/j.ins.2016.03.003

Power, D.J., Cyphert, D., & Roth, R.M. (2019a). Analytics, bias, and evidence: The quest for rational decision making. *Journal of Decision Systems*, *28*(2), 120–137. https://doi.org/10.1080/12460125.2019.1623534

Power, D.J., Heavin, C., & Keenan, P. (2019b). Decision systems redux. *Journal of Decision Systems*, *28*(1), 1–18. https://doi.org/10.1080/12460125.2019.1631683

Rybnytska, O., Burstein, F., Rybin, A.V., & Zaslavsky, A. (2018). Decision support for optimizing waste management. *Journal of Decision Systems*, *27*(sup1), 68–78. https://doi.org/10.1080/12460125.2018.1464312

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605

Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017). Predictive business process monitoring with LSTM neural networks. In *Proceedings of the 29th International Conference on Advanced Information Systems Engineering (CAiSE2017)* (pp. 477–492). Springer.

van der Aalst, W.M.P. (2004). *Business process management demystified: A tutorial on models, systems and standards for workflow management*. In *Proceedings of the 4th Advanced Course on Petri Nets (ACPN)* (pp. 1–65). Springer.

van der Aalst, W.M.P. (2008). Decision support based on process mining. In *Handbook on decision support systems 1* (pp. 637–657). Springer.

van der Aalst, W.M.P. (2015). Extracting event data from databases to unleash process mining. In *BPM - Driving innovation in a digital world* (pp. 105–128). Springer.

van der Aalst, W.M.P. (2016). *Process mining: Data science in action* (2nd ed.). Springer.

van Dongen, B.F., & van der Aalst, W.M.P. (2004). Multi-phase process mining: Building instance graphs. In *Proceedings of the 23rd International Conference on Conceptual Modeling (ER2004)* (pp. 362–376). Springer.

van Eck, M.L., Sidorova, N., & van der Aalst, W.M.P. (2019). Guided interaction exploration and performance analysis in artifact-centric process models. *Business & Information Systems Engineering*, *61*(6), 649–663. https://doi.org/10.1007/s12599-018-0546-0

Verenich, I., Dumas, M., La Rosa, M., & Nguyen, H. (2019). Predicting process performance: A white-box approach based on process models. *Journal of Software: Evolution and Process*, *31*(6), 1–26. https://doi.org/10.1002/smr.2170

Wang, J., Yu, D., Liu, C., & Sun, X. (2019). Outcome-oriented predictive process monitoring with attention-based bidirectional LSTM neural networks. In *Proceedings of the 17th International Conference on Web Services (ICWS2019)* (pp. 360–367). IEEE.

Weske, M., van der Aalst, W.M.P., & Verbeek, H.M.W. (2004). Advances in business process management. *Data & Knowledge Engineering*, *50*(1), 1–8. https://doi.org/10.1016/j.datak.2004.01.001

Zeiler, M.D. (2012). Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.