



Predictive End-to-End Enterprise Process Network Monitoring

Felix Oberdorf · Myriam Schaschek · Sven Weinzierl · Nikolai Stein ·
Martin Matzner · Christoph M. Flath

Received: 8 October 2021 / Accepted: 15 September 2022 / Published online: 6 December 2022
© The Author(s) 2022

Abstract Ever-growing data availability combined with rapid progress in analytics has laid the foundation for the emergence of business process analytics. Organizations strive to leverage predictive process analytics to obtain insights. However, current implementations are designed to deal with homogeneous data. Consequently, there is limited practical use in an organization with heterogeneous data sources. The paper proposes a method for predictive end-to-end enterprise process network monitoring leveraging multi-headed deep neural networks to overcome this limitation. A case study performed with a medium-sized German manufacturing company highlights the method's utility for organizations.

Keywords Predictive process analytics · Predictive process monitoring · Deep learning · Machine learning · Neural network · Business process management · Process mining

Accepted after 1 revision by Natalia Kliewer.

F. Oberdorf (✉) · N. Stein · C. M. Flath
Chair of Information Systems and Business Analytics, Julius-Maximilians-University Würzburg, Sanderring 2,
97070 Würzburg, Germany
e-mail: felix.oberdorf@uni-wuerzburg.de

M. Schaschek
Chair of Business Management and Information Systems, Julius-Maximilians-University Würzburg, Sanderring 2,
97070 Würzburg, Germany

S. Weinzierl · M. Matzner
Chair of Digital Industrial Service Systems, Friedrich-Alexander University Erlangen-Nürnberg, Fürther Str. 248,
90429 Nuremberg, Germany

1 Introduction

Business processes are the backbone of organizational value creation (Dumas et al. 2018). The progressing digitalization of business processes results in massive amounts of historical process data (van der Aalst 2016). In parallel, analytics capabilities facilitate the use of this data (Vera-Baquero et al. 2013; Beheshti et al. 2018). Business process analytics refers to a set of approaches, methods, and tools for analyzing process data to provide process participants, decision-makers, and other related stakeholders with insights into the efficiency and effectiveness of organizational processes (Zur Muehlen and Shapiro 2015; Polyvyanyy et al. 2017; Benatallah et al. 2016).

A type of business process analytics aims to predict future process behavior based on business process data (Zur Muehlen and Shapiro 2015). Predictive process analytics is typically realized by a class of information systems, called predictive monitoring systems, which promise to assist decision-makers through predictions based on historical event log data (Schwegmann et al. 2013). As a methodological basis for predictive monitoring systems, predictive process monitoring (PPM) is gaining momentum in business process management. PPM provides a set of methods that allow predicting measures of interest based on event log data (Maggi et al. 2014). By gaining insights into the uncertain future of a process, PPM methods enable decision-makers to prevent undesirable outcomes (van der Aalst et al. 2010; Márquez-Chamorro et al. 2017). For example, in a hypothetical manufacturing company with a production process manifested in a manufacturing execution system, a PPM tool can be used to predict disruptions for running process instances. The predictions allow the company to proactively intervene in the respective process instances to mitigate or prevent disruptions. As disruptions

directly affect productivity, proactive management of process instances enhances value creation. This is typically achieved by providing extended and relevant information at the right time which in turn will lead to time, cost, and workforce savings.

As event log data, PPM typically refers to a single event log documenting a specific process or multiple sub-processes (e.g., Cuzzocrea et al. 2019; Senderovich et al. 2019). Oftentimes, the (process) control flow information is feature-encoded, with one target variable per process instance or prefix (part of the process instance) (e.g., Breuker et al. 2016; Lakshmanan et al. 2015). More sophisticated approaches append (process) context information to control flow information of a single event log to increase the explainability of input variables concerning the target variable (e.g., Yeshchenko et al. 2018; Brunk et al. 2020).

In organizations with a process-oriented design (Eversheim 2013), the departments' organizational alignment supports end-to-end business process execution and management. Departments are connected via the *organization and departments* layer and via the *enterprise process network* layer, connecting departments, processes, and information systems (Fig. 1).¹ More specifically, this layer establishes inter-department and inter-process dependencies, as departments will usually be involved in a multitude of processes (e.g., the production department is responsible for disruptions affecting the shipment process in the logistics department or may influence the sales process in the sales department) and a process will often involve multiple departments (e.g., an order process (red), that spans the sales, logistics, and production department).

Consequently, the enterprise process network extends the scope from the process level to the process network level. The primary data source in the enterprise process networks are event logs documenting the control flow information of a process. This logged control flow is often combined with additional event-log-related context information directly related to the process. The primary log data is supplemented by additional data sources which are related to the process, e.g., sensor data (temperature, humidity, vibration), or measurements. Complex manufacturing business process environments encompass many heterogeneous data sources. We refer to these as different types of data, i.e., measuring differently scaled data or collecting data at varying frequency (Canizo et al. 2019).

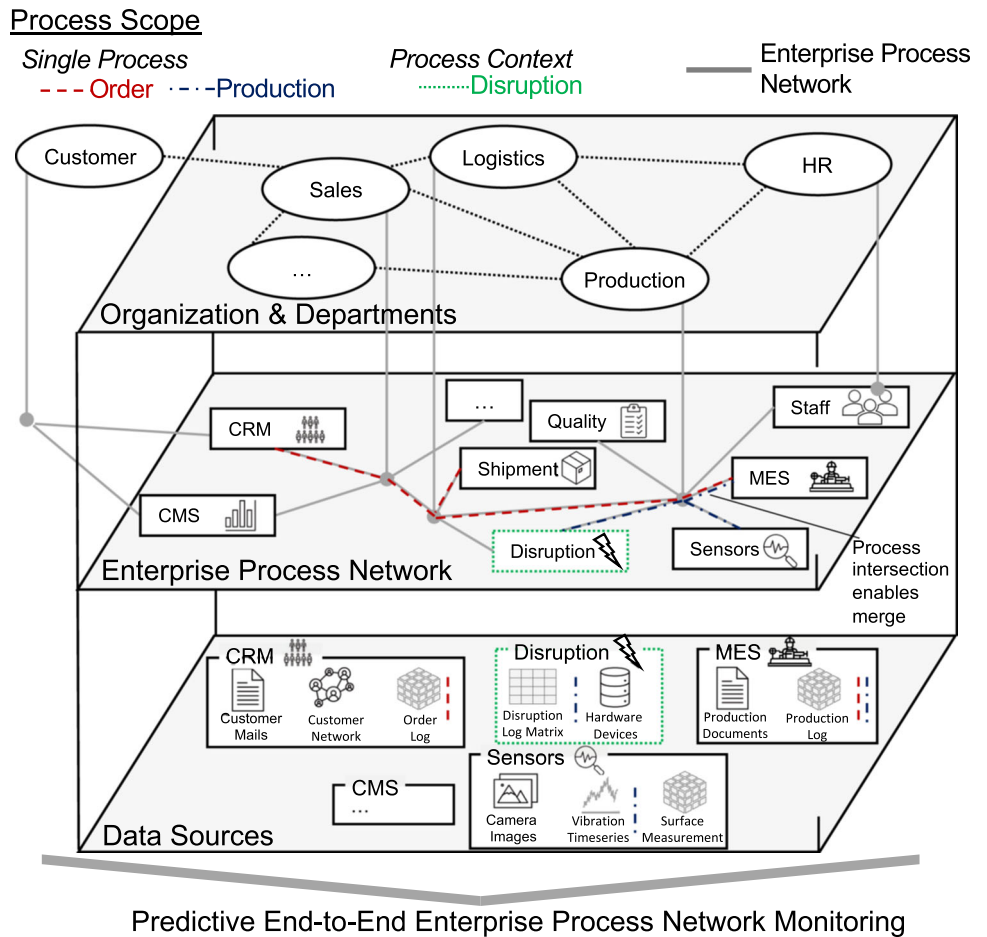
¹ We consider the enterprise process network as the intra-organizational process network, based on the control view concept of the ARIS framework (Scheer 2013) for the architecture of integrated information systems. Furthermore, we adopt the "business process trends pyramid" of vom Brocke and Rosemann (2014, p. 54) with its distinct layers for enterprise (organization and departments), business processes, and implementation (both enterprise process network).

Given this data scope definition, Fig. 1 distinguishes data sources such as an order event log (red-dashed), a production event log (blue-dash-dotted), both with control flow and process-related context information, as well as disruption context information (green-dotted).² In this exemplary enterprise process network, a disruption prediction may benefit from additional information from the logistics process. By considering the interplay between the different processes, the predictive power may increase, as more data potentially results in additional relevant features. Higher predictive power enhances the organization's value creation. By contrast, existing PPM approaches do not adopt such a process network perspective (Borkowski et al. 2019). This may limit their practical use as seamless combination of heterogeneous data sources relating to multiple processes is very difficult. By focusing on enterprise process network monitoring, we address this limitation and introduce a predictive end-to-end method. The main contribution of our research is threefold:

1. We present a method for predictive enterprise process network monitoring in the business process management (BPM) domain. The method establishes an end-to-end perspective on predictive process network monitoring in an organizational context. In doing so, it facilitates the combination of heterogeneous data sources for predictive tasks and guides the problem specification as well as the design and application of a multi-headed neural network (MH-NN) model.
2. Our novel multi-headed deep neural network (DNN) model integrates multiple data sources from an enterprise process network, such as the color-highlighted process logs or context information in Fig. 1. With this deep learning (DL) architecture, the heterogeneous data are processed in dedicated neural network (NN) input heads and concatenated for prediction, based on cross-department information.
3. The results from a case study conducted with a medium-sized German manufacturing company shed light on the practical relevance. We evaluate our method against traditional machine learning (ML) and state-of-the-art DL approaches in terms of predictive power and runtime performance based on real-world data. While the DL model constructed with our method exhibits somewhat higher computational costs, its predictive power is significantly higher than the considered baselines.

² To be able to merge individual process event logs and context information, we need a common point of reference, such as a timestamp.

Fig. 1 Overview of process scope in the organizational context



2 Background and Related Work

We first review recent advances in PPM with a special focus on predictive models. In doing so, we highlight the research gap and position our methodological contributions.

2.1 Prediction Methods in Predictive Process Monitoring

Process mining (PM) is an established process analysis method in BPM that involves data-driven (process model) discovery, conformance checking, and enhancement of processes (van der Aalst et al. 2011a). PM’s general idea is to gain process transparency from event log data. It is thus an approach for process analytics, particularly focusing on ex-post process diagnostics. With the advent of predictive analytics, new potentials of gaining insights from event log data have been unlocked (Breuker et al. 2016). Using these methods, PPM has emerged as a new subfield of PM (Márquez-Chamorro et al. 2017). PPM provides a set of techniques to predict the properties of operational processes, which can be arranged into two general groups

(Mehdiyev et al. 2020). The first group of techniques addresses *regression* tasks and refers to the prediction of continuous target variables, such as the completion time of a process instance (e.g., van der Aalst et al. 2011b; Wahid et al. 2019). In contrast, the second group tackles *classification* tasks and refers to the prediction of discrete target variables, such as the next activity (e.g., Mehdiyev et al. 2017; Breuker et al. 2016), process violations (e.g., Di Francescomarino et al. 2016), or process-related outcomes (e.g., Flath and Stein 2018; Kratsch et al. 2020). A branch of early PPM approaches augment discovered process models with predictive capabilities but require certain model structures to support prediction tasks. Thereby, the process model is transformed into a predictive model. For example, van der Aalst et al. (2011b) introduce a technique that uses an annotated transition system with the capability to predict process completion time based on historical event log data. Another example is Rogge-Solti et al. (2013), who mine a stochastic Petri net with arbitrary delay distribution from event log data. These approaches can be described as *process-aware* because they utilize “(...) an explicit representation of the process model to make predictions” (Márquez-Chamorro et al. 2017, p. 4).

However, real-world processes are usually more complex than the discovered process models (van der Aalst 2011). The process-model-dependence limits the predictive power (Senderovich et al. 2019). To overcome this restriction, another, more recent branch of PPM approaches proposes to encode sequences of process steps as feature vectors for the straightforward use of ML models. This transforms the event log's sequential process information into a predictive model without discovering a process model. Leveraging the generalization power of ML models, sequence-encoding approaches often outperform predictive models built on top of discovered process models (Senderovich et al. 2017).

The multi-layer perceptron (MLP) is a classic NN architecture (from the class of feed-forward DNN, Goodfellow et al. 2016), that has been leveraged for PPM. The MLP does not explicitly model temporality. As a workaround, sequential data has a two-dimensional data structure. For example, Theis and Darabi (2019) used MLPs to predict the next activities. DNNs have been applied to PPM, due to the conceptual similarities between next event prediction and natural language processing tasks (Evermann et al. 2016). DNNs can outperform statistical (e.g., Verenich et al. 2019) and traditional ML approaches (e.g., Kratsch et al. 2020; Mehdiyev et al. 2020; Evermann et al. 2016). DNNs perform multirepresentation learning, which "(...) focuses on extracting the multiple representations from the single view of data" (Zhu et al. 2019, p. 3) and are good at unveiling intricate structures in data (LeCun et al. 2015). A popular sub-class of DNNs are recurrent neural network (RNN) approaches (Rama-Maneiro et al. 2021), including LSTM and gated recurrent unit (GRU) neural networks, providing the capability to capture temporal dependencies within sequences (Rumelhart et al. 1985). Another DNN architecture, which allows the processing of temporal patterns across short time horizon (local temporal neighborhood), is the convolutional neural network (CNN) (Zhao et al. 2017). To leverage the potential of CNN for PPM, a preprocessing of sequences from temporal to spatial structure is needed. Pasquadibisceglie et al. (2019) show the validity of such a sequence preprocessing for predicting the next process activity using the *helpdesk event log* and *BPI challenge 2012* data. Graph neural networks (GNNs) have recently been used in PPM because the process control flow follows a graph structure (e.g., Stierle et al. 2021) and can directly be processed through GNNs. Beyond the four general architectural types MLPs, RNNs, CNNs, and GNNs, extensions (e.g., transformer networks with dense layers like MLPs; Moon et al. 2021) or combinations (e.g., long-term recurrent convolutional networks; Park and Song 2020) were proposed for PPM.

2.2 Data Scope vs. Prediction Methods in Predictive Process Monitoring

Statistical approaches in PPM (e.g., van der Aalst et al. 2011b; Rogge-Solti et al. 2013) start with the control flow information of event log data. This type of information is key for process predictions, as the control flow of processes describes their structure.

By using ML, the scope of data is extended and PPM techniques can encode further event log information in feature vectors (e.g., Folino et al. 2012). This additional information is called *process context information*. It characterizes the environment in which the process is performed (Da Cunha Mattos et al. 2014; Rosemann et al. 2008), and represents, for example, information about the resource that performs an activity.

In recent years, PPM research has suggested DL architectures that integrate context information to improve prediction results (Rama-Maneiro et al. 2021). Current PPM approaches receive single event logs as input and do not leverage information from multiple data sources. Thereby, an event log can also contain several subprocesses, such as in the event log shared at the BPI Challenge 2012.³

Currently, there are no PPM techniques using multiple data sources to perform end-to-end enterprise process network predictions. Figure 2 differentiates published PPM techniques based on two dimensions, namely *data scope* and *prediction method*, to extract the research gap within scientific literature concerning end-to-end PPM.

New time series forecasting techniques (e.g., Canizo et al. 2019; Mo et al. 2020; Wan et al. 2019) offer a promising way to realize such predictions through multi-headed NN. These networks process data from each input head (e.g., from a machine sensor) individually and merge the heads' outcomes subsequently. Motivated by this idea, we set out to adapt this method for end-to-end enterprise process networks.

3 Predictive End-To-End Enterprise Process Network Monitoring

We propose PPNM, a five-phase method for predictive end-to-end enterprise process network monitoring (Fig. 3). We develop our PPNM method based on the *method engineering research framework for information systems development methods and tools* proposed by Brinkkemper (1996). Methods describe systematic procedures "to perform a systems development project, based on a specific

³ <https://www.win.tue.nl/bpi/doku.php?id=2012:challenge> &redirect=lid=2012/challenge.

Fig. 2 Classification of exemplary PPM techniques by *data scope* and *prediction method* with highlighted research gap and our proposed end-to-end enterprise process network monitoring (PPNM) method

| | | | | |
|--------------------|-----------------------|--------------------------------|---|---|
| Data scope | Multiple data sources | CF _s + CI + add. CI | PPNM method | |
| | Multiple data sources | CF _s + CI | | Research gap |
| | Single data source | CF + Context information (CI) | Folino et al. 2012 Senderovich et al. 2018 Yeshchenko et al. 2018 | Brunk et al. 2020 Schönig et al. 2018 Evermann et al. 2017 Camargo et al. 2019 |
| | Single data source | Control flow (CF) | Rogge-Solti et al. 2013 van der Aalst et al. 2011 | Tax et al. 2017 |
| | | Pure statistics | Machine learning | Deep learning |
| Prediction methods | | | | |

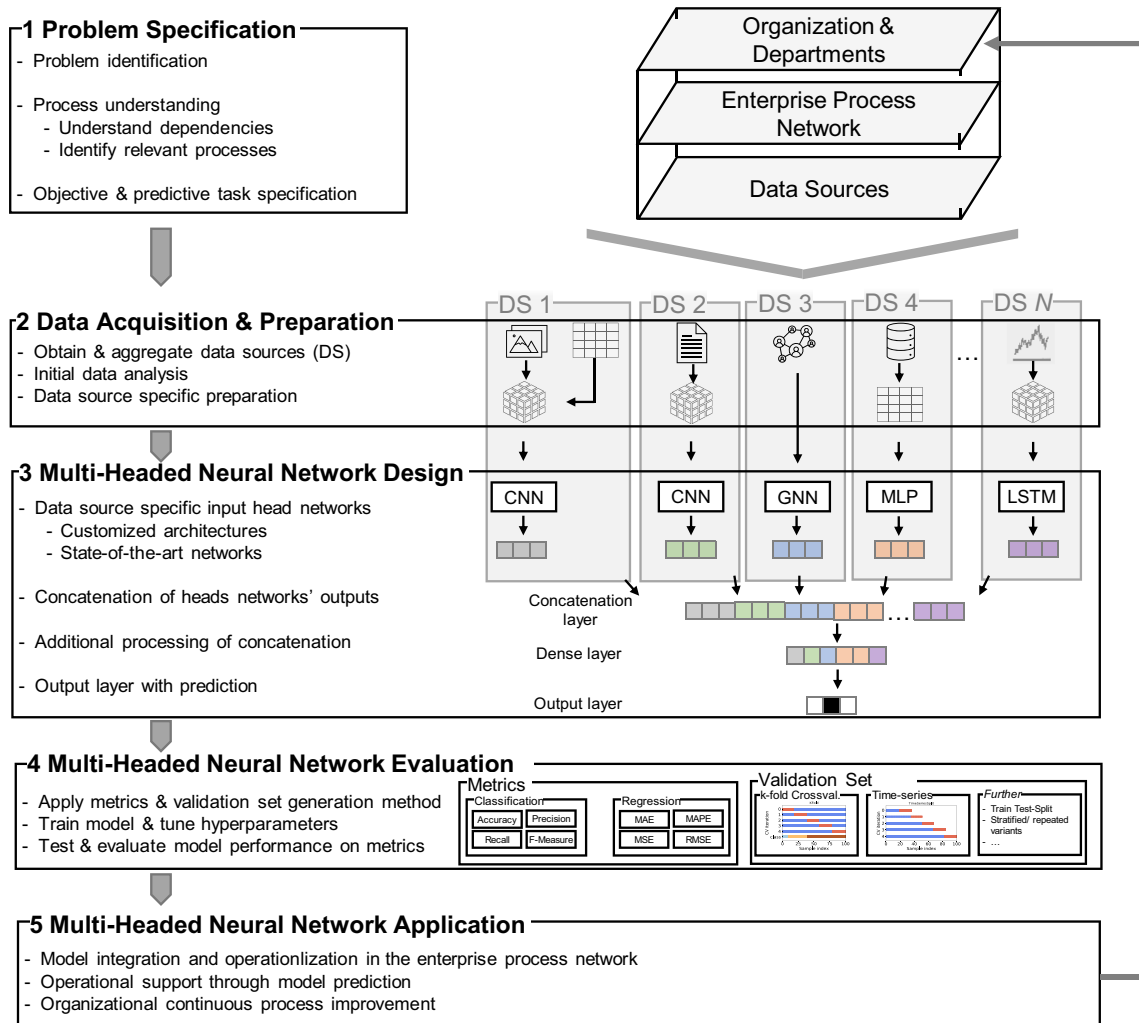


Fig. 3 Five-phase method for predictive end-to-end enterprise process network monitoring

way of thinking, consisting of directions and rules, structured in a systematic way in development activities” (Brinkkemper 1996). The method engineering process consists of three phases (Gupta and Prakash 2001): requirements engineering, method design, and method implementation. First, we define requirements for the construction of the PPNM method such as the application as an end-to-end approach, the integration of multiple data sources, and an outperforming predictive power. Second, we present the design, evaluation, and implementation of the PPNM method in this section and describe the method’s phases in detail in the context of a case study of a medium-sized German manufacturing company. Finally, we discuss the PPNM method critically and provide implications (Sect. 3.4).

In our PPNM method, at first, the underlying problem is specified. This includes (business) problem identification, (business) process understanding, and predictive task specification. Second, the method prescribes to acquire and prepare the input data for the MH-NN model. Third, the MH-NN model is designed and subsequently evaluated in the fourth phase. Lastly, PPNM describes aspects of the model application.

3.1 Problem Specification

The first phase specifies the problem by adapting the approach of Benscoter (2012), beginning with the *problem identification* at the business department or enterprise process network layer. Their approach to “identify and analyze problems in your organization” (Benscoter 2012) has a particular focus on identifying a situation’s impact on processes and workers as well as problem-relevant metrics. Subsequently, the establishment of an *understanding of the interdependent processes* and data sources is crucial. Within an organization’s layers, all relevant processes and data sources, which can add value to the predictive analysis task, should be identified. Then, their dependencies should be understood to identify common denominators for synchronizing heterogeneous data sources and how they relate to the organizational problem or situation. Based on this process and data understanding, the method prescribes to *define the organizational objective and the type of predictive task* (regression or classification).⁴

⁴ A regression relates to estimating a numerical output, such as the forecast of financial, sales, downtime information, or organizational key performance indicators. In contrast, a classification’s output incorporates the estimation of categorical types, such as if an event may happen (binary) or if an event has a particular type (multi-class).

3.2 Data Acquisition and Preparation

Having identified relevant processes and data sources, we next acquire and prepare input data for the MH-NN. *Data acquisition* relates to activities seeking to obtain the heterogeneous data. This data is analyzed to gain insights about the data source and subsequently prepare it for the MH-NN. The network processes each data source individually, without the need for prior aggregation and combination. We apply some standard preparation techniques (Han et al. 2011) but more generally follow the DL recommendation of focusing on standard DL architectures for feature extraction and limiting extensive preparation (LeCun et al. 2015).

As a crucial step of data preparation, PPM requires appropriately encoded events and sequences. *Events* can be encoded based on the attributes’ type. *Sequences* of events can be encoded as feature-outcome pairs (Van Dongen et al. 2008), n-grams of sub-sequences (Mehdiyev et al. 2020), feature vectors derived from Petri nets (Theis and Darabi 2019), or weighted adjacency matrices (Oberdorf et al. 2021a).

3.3 Multi-headed Neural Network Design

Designing the multi-headed NN, we follow recent work on PPM methods, which move from explicit process models and traditional ML approaches to NN-based approaches (Mehdiyev et al. 2020). Yet, for some scenarios, the sequential structure of these NNs is not sufficiently flexible such as, if data from different sources with different dimensions are required to explain the output variable. Following Chollet (2018, p. 301), the proposed architecture for these cases is a multi-head NN. Architectures with multiple heads use *independent single-channel input heads* to process each input individually. With this approach, each *data source can be processed*, according to its data type and structure. Head outputs are then *concatenated* and *further processed* to obtain a *prediction in the output layer*.

For the design of the multi-headed NN, the method facilitates the use of a multitude of architectures (Fig. 4). In general, it distinguishes customized and state-of-the-art architectures.

For *customized architectures*, a combination of NN layers can be selected (Sect. 2.1). Following Goodfellow et al. (2016), combining various layers in a task-specific manner enables the implicit extraction of valuable features. To this end, distinct properties of architectures can be leveraged, such as the particular suitability of LSTM layers to process time-series or CNN layers for matrix data. These properties can even be combined to process time-series, such as a combination of LSTM and CNN layers (Brownlee 2017).

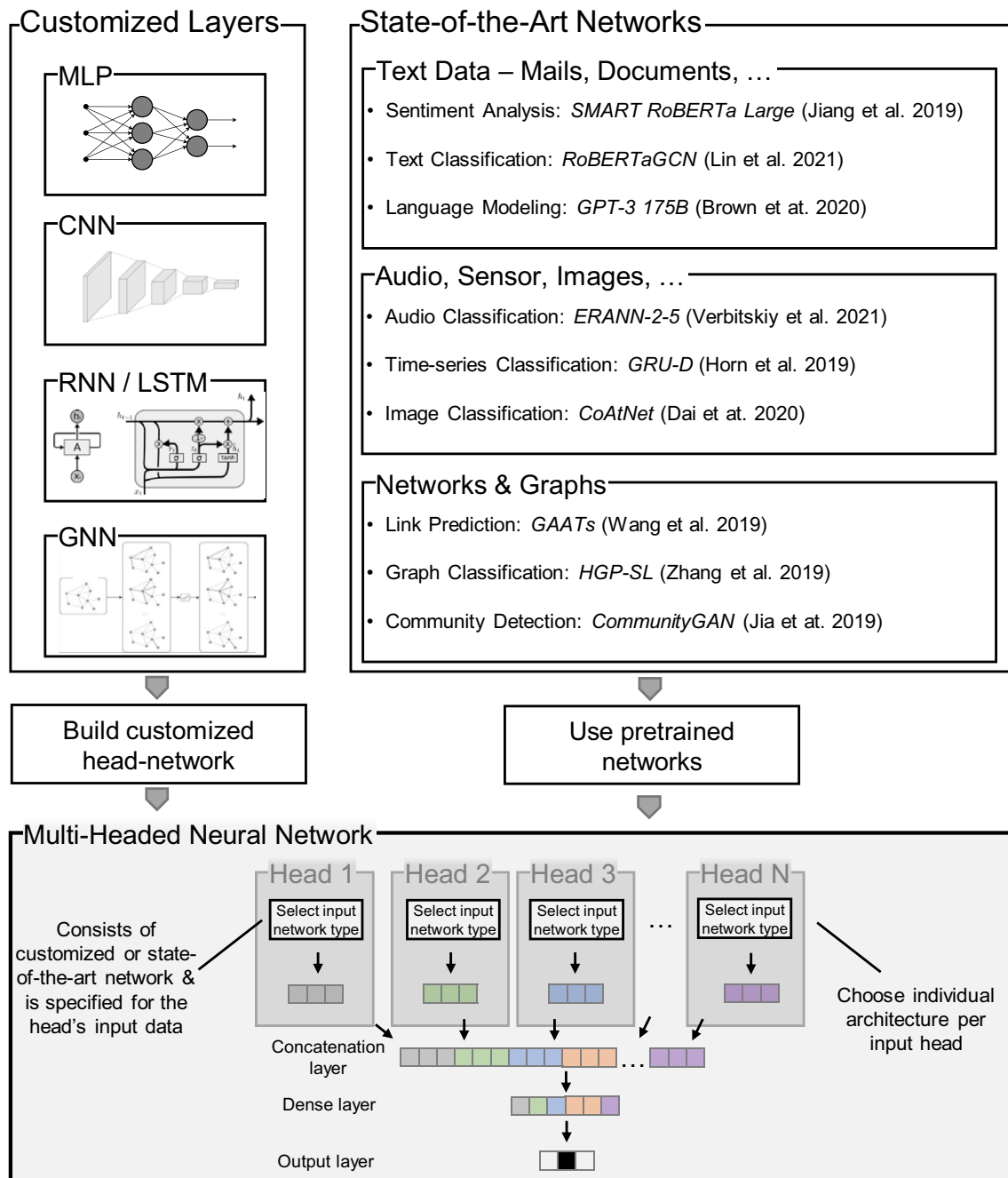


Fig. 4 Overview of potential NN network layers and state-of-the-art networks (Papers with Code 2021) for the NN’s multiple input heads

In addition to the customized architectures, the method taps into recent advances in the DL domain by incorporating established architectures. There are *state-of-the-art* architectures for the various domains such as image, text, or signal processing. As the numbers of available architectures are constantly changing, we suggest checking for currently available state-of-the-art networks during a model’s design phase to build on recent research

advances.⁵ Figure 4 provides an overview of currently established state-of-the-art methods for various tasks. Depending on the data type, we show current DL solutions for problems, such as sentiment analysis (Jiang et al.

⁵ Besides recent publications, more practical related sources for recent advances are <https://paperswithcode.com/>, <https://github.com/sebastianruder/NLP-progress>, or <https://github.com/rwightman/pytorch-image-models>.

2019), language modeling (Brown et al. 2020), text, time-series, audio, image, or graph classification (Lin et al. 2021; Horn et al. 2020; Verbitskiy and Vyshegorodtsev 2021; Dai et al. 2021; Zhang et al. 2019), as well as link prediction (Wang et al. 2019), or community detection (Jia et al. 2019) in networks.

The common denominator for such models is that they consist of complex DL architectures with many hidden layers and trainable parameters. Because the training of such models is computationally demanding, they are usually provided with pretrained weights, which can then be leveraged for the prediction task at hand or even fine-tuned based on the task's specific data.

3.4 Multi-headed Neural Network Evaluation

The method next requires to consider aspects of model evaluation. For this purpose, we follow Brownlee (2020)'s approach, including the generation of a validation set and the use of performance metrics to assess a model's performance. The evaluation of the resulting model is crucial for the selection of a proper configuration. It reveals whether the model is suitable to estimate the desired target variables. To this end, test and validation sets are artificially generated through validation methods. In particular, in the field of PPM, selecting an appropriate validation set method is challenging. There are three established *validation set generation* methods (Fig. 3). In addition to the validation set generation, it is common to keep a holdout set containing exclusive data for a final model evaluation.

The most common method used is a straightforward strategy, referred to as a train-test split procedure (James et al. 2017, p.176–178). An alternative evaluation procedure is k -fold cross-validation for estimating the prediction error (James et al. 2017, p.181–186). It splits the data set into k folds, uses $k - 1$ of folds for training and the other fold for validation.

In some settings, regular k -fold cross-validation is not directly applicable. This is the case for time-series data, where observations are samples with fixed time intervals. The constraint is the temporal components inherent in the problem. Here, a time-series split is an appropriate method, where in the k^{th} split, the first k folds are used as a train set, and the $(k + 1)^{\text{th}}$ fold is used as a test set. Time-series splits have the drawback that there is overlap between the training and testing data. This limitation can be resolved by forward testing techniques where the model is automatically retrained at each time step when new data is added (Kohzadi et al. 1996).

After selecting an appropriate validation technique, the next step is choosing a *performance metric* for the predictive problem. For classification tasks, accuracy is a very

commonly applied metric. It measures the ratio between the number of correctly predicted target labels and the total number of predictions. The accuracy metric is only designed for tasks considering all classes as equally important, and its usefulness suffers if the samples within the classes are not equally distributed. For imbalanced data sets, the preferable metrics are balanced accuracy, the weighted F1-score, or the Matthews correlation coefficient. The most common metrics for evaluating predictive regression tasks are mean absolute error (MAE), or the mean squared error (MSE). To provide relational insights, in particular in an organizational context, the mean absolute percentage error (MAPE) is useful. One of the metrics is then chosen for model training, yet it is common to provide an overview of multiple metrics for the evaluation.

Based on the validation set and performance metrics, the model is *trained and tuned*. For effective and efficient tuning of training parameters, several software packages such as Hyperopt (Komer et al. 2019), keras-tuner (O'Malley et al. 2019), or auto-sklearn (Feurer et al. 2019), can be used. These tools instantiate intelligent search procedures (Bergstra and Bengio 2012; Snoek et al. 2012). Finally, the tuned models are *tested* and the learning curves *evaluated*, to ensure a robust model for the prediction task.

3.5 Multi-headed Neural Network Application

In the last phase, the method describes aspects for MH-NN application. This includes the *operationalization* of data acquisition and preparation as well as the deployment of an evaluated MH-NN. Of particular importance is the live connection to the enterprise process network and the data sources. Instead of training on historical data, the MH-NN must handle live data to provide real-time predictions. Thus, besides model performance, runtime performance becomes particularly relevant during model deployment.

If the model is integrated into the enterprise process network and connected to (live) data sources, it facilitates the prediction of the desired variable. Such a prediction then affects an organizational process, for example, through the prediction of upcoming events or the classification of an event's type, which can be used to provide better solutions in organizations. As the processes are improved due to the prediction, the designed model then assists in the organizational goal of *process improvement*.

4 Method Evaluation

To evaluate the PPNM method, we use a real-world use case and present the processing of the method's five phases. We provide insights about the real-world application

and discuss the method's engineering as well as application.

4.1 Problem Specification and Industry Background

We collaborated with a medium-sized German manufacturing company. The firm has multiple distributed production and assembly lines for highly customized mechatronics products. Competitive pressure necessitates the firm to offer high-quality products with (mass) customization options. This combination can lead to fairly complex production processes. Here, disruptions⁶ where a worker has to interrupt work, are not uncommon.

To efficiently handle such disruptions, our cooperation partner has deployed a disruption management system (Oberdorf et al. 2021b). The system automates responder notification for solving a disruption.⁷ As a disruption is solved through the responding agent, the agent provides the system additional information, such as one of 32 disruption reasons (types). We identified the disruption's type as a central component of the problem specification. If the type was already known, an agent could already prepare the solution process (e.g., bringing relevant tools or documentation), which reduces the disruption associated downtime.

In parallel, the production processes have been analyzed with PM techniques to identify optimization potentials. However, due to the enterprise process network's complexity, interrelations, and dependencies, the respective analyses are very time-consuming. Consequently, the realization horizon of possible benefits is long. Striving for immediate benefit with minimal analysis effort, we adopt the PPNM method and provide an end-to-end PPNM solution. Thereby, the MH-NN is integrated into the organizational enterprise process network. The organizational objective is to improve the production process through better disruption handling, resulting in reduced downtime. We do so by predicting the disruption type and providing a solution suggestion to a notified agent based on the prediction. Accurate predictions are essential for meaningful notifications and suggestions.

We engaged with various departments (digitalization, logistics, and production) to evaluate the PPNM method in

Table 1 Overview of the *production* and *logistic* event log with a summary of descriptive statistics

| | Data sources | Production | Logistics |
|---------------------|-------------------|------------|-----------|
| Number of | Process instances | 24,581 | 24,581 |
| | Process variants | 859 | 240 |
| | Activity types | 156 | 69 |
| Events per instance | Minimum | 4 | 2 |
| | Average | 5 | 4 |
| | Maximum | 34 | 20 |
| Process | Sparsity | 0.006 | 0.002 |
| | Variation | 0.034 | 0.010 |
| | Repetitiveness | 0.425 | 0.434 |

practice. Thereby, we elaborated on each department's process event log and related databases.⁸

4.2 Data Acquisition and Preparation

We compute basic statistics and advanced event log characteristics such as sparsity, variation, or repetitiveness (Heinrich et al. 2021; Di Francescomarino et al. 2017) to better understand the *production* and *logistics* event log data used (Table 1) as well as the disruption context information (Table 2). The descriptives demonstrate the high complexity of the semi-structured event logs with many unique process variants and activity types. Furthermore, we combine both event logs and obtain the combined production event log, which contains information about the logistics and production process, its control flow, and context information.

The disruption log is closely related to the intra-logistics and production departments and processes, as disruptions occur in both departments. It contains information about historical disruptions with features such as the disruption hardware id and timestamp. This way disruptions can be mapped to a workplace through the hardware device database. This enables us to retrieve product information from the respective data sources, which we can also leverage as features for the predictive task.

We follow the PPNM method to design a multi-head NN: We start with the data preparation for the disruption log. Concerning the hardware id, we include additional workstation and product information using one-hot encoding. Besides, we can extract time features, such as days, weekdays, hours, and minutes, from the disruption-associated timestamp, which we subsequently normalize.

⁸ Production and logistics processes span across the departments, such as logistics events performed in the production department. However, the respective logs mainly originate from one of the departments.

⁶ Typical reasons include, e.g., missing materials, damaged parts, or non-functional machines.

⁷ When an employee detects a disruption during the production or logistics process, the employee presses one of the system's hardware devices. The system then automatically notifies a responding agent (employee with specialized skills for disruption solving), who assists in solving the disruption.

Table 2 Overview of the disruption context information features

| | Data source | Disruption |
|-----------|----------------------|------------|
| Number of | Events | 4739 |
| | Numerical features | 4 |
| | Categorical features | 20 |

By aggregating the logistics and production log, we obtain a process event log with context information. To transform the event log into valuable features, we follow Oberdorf et al. (2021a) and select process instances within a time window, which we subsequently transform into a matrix representation. Thereby, rows and columns relate to specific workstations and the value of a distinct cell to the production quantity within the time window. For NN preparation, we scale each matrix by the maximum production quantity of all matrices. This process is used for the control flow data (*process matrices*) as well as for the context data (*context matrices*).

4.3 Multi-headed Neural Network Design

We choose a three-headed DNN architecture (Fig. 6 in the Appendix, available online via <http://link.springer.com>). The *disruption vector* is the first input for the multi-head NN and is processed with an MLP (head), including a batch normalization. For both input matrices (weighted adjacency and context matrices), we use CNN architectures, consisting of stacked CNN and fully connected (FC) layers. For the context information, we apply a CNN-FC architecture to perform best in combination with the other heads. It consists of three CNN-layers and a subsequent FC layer. The third head's design – the process event head – posts a more challenging task. We tried the architecture used for context information and appended the adjacency matrices to the context matrices in the fourth dimension.⁹ However, none of these approaches delivered satisfactory results. For this reason, we leverage process knowledge in the definition of the CNN kernel sizes. Basically, multiple sequential CNN layers extract features with distinct kernels.¹⁰ After feature extraction, both matrix head outputs have a 4D shape. To combine both with the disruption head's output vector, we flatten the matrix head outputs. The flattened features are subsequently processed by a

⁹ The first dimension relates to the batch size, dimensions two and three to the matrix, and the fourth dimension to the heads of a CNN. In image processing, the fourth dimension represents multiple color channels.

¹⁰ A small kernel is leveraged to extract information within a production line, up to large kernels, which extract information across multiple production lines.

dense layer and the final output dense layer for the multi-class classification task.

4.4 Multi-headed Neural Network Evaluation

For the quantitative evaluation, we classify the type of each disruption event with the constructed MH-NN. In addition, we compare traditional aggregation-based approaches, where we append the disruption input vector with engineered (process) adjacency list features and, in addition, a vector of context information. Instead of 24 disruption vector features, we use 291 input features for adjacency list combination. In combination with the 267 additional adjacency list features, we use a total of 558 features.

We perform a five-time repeated five-fold cross-validation with random initialization. To prevent the DNN models from overfitting, we integrate an early stopping rule for validation accuracy. We store the best-performing models during each training cycle and used a Bayesian optimization algorithm (O'Malley et al. 2019) for hyperparameter tuning. Our tuning objective is the validation accuracy with a maximum retrial of 50 configurations.

For the tuned FC, CNN, and multi-headed (MH) models, we first compare the validation loss (Fig. 5) at the stopping time. The multi-headed approach's loss clearly outperforms the other DNN architectures. In addition, it reaches a solid model with fewer epochs compared to the CNN or FC architecture with flattened feature inputs.

The final models are subsequently evaluated on the hold-out set, resulting in the metrics summarized in Table 3, where we compare basic benchmark approaches such as most frequent (mFreq) or k-nearest-neighbor (KNN) methods, as well as more advanced machine learning, deep learning, and the multi-headed architectures. All evaluated algorithms, ML, and DNN models outperform the naive benchmark in terms of BMACC as well as the (weighted) F1-score, Precision, and Recall-score. We observe that the FC architecture benefits from the additional adjacency list features. However, we also see that the additional context list features lead to a decrease in predictive power, indicating that the FC architecture cannot completely prevent overfitting.

A comparison of CNN with only adjacency matrix features shows that they contain some basic information. However, this performance does not match the FC architecture with disruption and adjacency list features. The proposed multi-headed NN approach outperforms all benchmark architectures. Besides the better training behavior of the multi-headed NN approach, the higher aggregation of the data seems to result in this information loss. Due to the matrix properties, the CNN can identify patterns in the data that lead to improved results. Note that the resulting multi-class accuracy refers to a 32-class

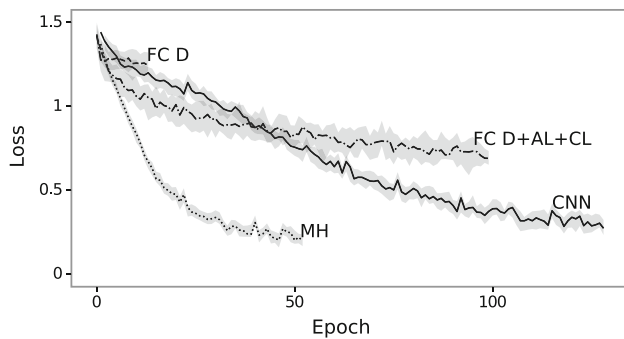


Fig. 5 Comparison of validation loss of FC, CNN, and MH algorithms for disruption classification with input scenarios for disruption vector (D), the combination with adjacency list (AL) as well as context list (CL) vector

classification problem. Accordingly, the 81% MH accuracy is a good result, allowing a reliable solution suggestion.

The experimental results of the multi-headed architecture are in line with recent research in computer vision (He et al. 2016) in general and predictive process monitoring (Rama-Maneiro et al. 2021) in particular. The DL algorithms show superior performance for the specific use case of multi-class classification. However, the superiority of the MH-NN architecture in terms of predictive power is tied to some drawbacks regarding implementation and training time. Compared to the standard ML models, that are readily implemented using libraries such as Scikit-learn (Pedregosa et al. 2011), finding and implementing optimal NN architectures for each network head is a complex and time-consuming task. Additionally, the training of the multi-headed NN takes significantly more time.¹¹ Clearly, this is a limitation of the MH-NN model. For our use case, however, the prediction duration is more relevant, which is acceptable and facilitates the application of the model.

4.5 Multi-headed Neural Network Application

In the last phase of the PPNM method, we deploy data acquisition and preparation as well as the identified best model. The method's resources are deployed on a standard commercial virtual machine with Linux OS. It is connected to the organizational enterprise process network through an MQTT connection, which enables the live interaction with the disruption management system. Whenever a disruption occurs and the worker triggers the notification process, the disruption data is transmitted through the MQTT connection and triggers the prediction process. Recent production and intra-logistic event log data are automatically obtained, and all data are prepared as well as forwarded to the MH-NN. The prediction result is then transmitted to the

disruption management system and improves the information, which a responding agent receives as part of the disruption notification. Therefore, better preparation for the disruption task at hand is possible, which ultimately reduces disruption downtimes and associated costs.

To provide an evaluation based on the real-world setting, we follow the approach described by Kraus et al. (2020) and evaluate the prediction error costs (c_{err}). The costs originate from the downtimes for solving a disruption. We calculate the costs based on the production environment setup across the production lines with a mean disruption rate of 1.3% per produced part and report it in a relative monetary unit (MU). To do so, we leverage a previously established study that analyzes the prediction accuracy with respect to the resulting downtimes (Oberdorf et al. 2021b). Based on our quantitative study, increasing model accuracy results in decreasing downtimes due to better information and thus preparation of the notified agents. Further, an increasing accuracy, such as for the MH-NN, results in reduced prediction error costs. While, for example, the basic benchmark approach mFreq creates prediction error costs of about 3,246 MU, the MH-NN comes to prediction error costs of 695 MU.

In addition, we interviewed a data scientist and a project manager. According to the data scientist, the collaboration facilitated the awareness for the great interdependence of the processes. Clearly, processes affect each other, even across organizational borders, which the employees were aware of. However, combining these heterogeneous data sources meant great efforts. The proposed method provides a valuable tool for structured data combination across departments.

Of course, we are aware of interdependent processes, but leveraging the data was usually not practical. The multi-headed NN approaches bridge this gap, as we can further combine data without the downside of extensive aggregation. And due to the deployment, even without first searching and collecting the data. (Data Scientist)

We presented the initial results to data scientists, project managers, and managers of the cooperation partner and discussed the practical implications. Aligned with the data scientist's perspective, the project manager depicts the potential on an organizational scale. Beyond the digitalization, production, and logistics departments, applications to financial and controlling are of particular focus. Connections to the customer resources management (CRM) system or website user statistics may enable a better prediction of incoming orders, leading to improved production planning. In addition to better predictions, the deployment is then of special importance.

¹¹ We trained all models on a NVIDIA GeForce GTX 1080 TI with 11 GB GDDR5X RAM.

Table 3 Comparison of algorithms for disruption classification with input scenarios for disruption vector (D), the combination with adjacency list (AL) as well as context list (CL) vector, and the adjacency matrix (AM) as well as context matrix (CM) input. Reported are the balanced multi-class accuracy (BMACC), the F1-score, precision (Prec), recall (Rec), training duration (t_{train}), duration for predictions (t_{pred}), and the prediction error costs (c_{err} – an increasing model performance results in faster disruption handling and is transferred to monetary units)

Best performing model is highlighted bold for the accuracy associated metrics

| Type | Input features | $\overline{\text{BMACC}}$ | $\overline{\text{F1}}$ | $\overline{\text{Prec}}$ | $\overline{\text{Rec}}$ | $\overline{t_{\text{Train}}}$ (s) | $\overline{t_{\text{Pred}}}$ (s) | $\overline{c_{\text{err}}}$ |
|-----------------------------------|----------------|---------------------------|------------------------|--------------------------|-------------------------|-----------------------------------|----------------------------------|-----------------------------|
| <i>Basic benchmark approaches</i> | | | | | | | | |
| mFreq | D | 0.131 | 0.13 | 0.12 | 0.14 | – | – | 3246 |
| KNN | D | 0.332 | 0.41 | 0.34 | 0.51 | 0.005 | 0.028 | 2496 |
| KNN | D & AL | 0.384 | 0.49 | 0.49 | 0.50 | 0.027 | 0.184 | 2301 |
| <i>ML architectures</i> | | | | | | | | |
| SVM | D | 0.296 | 0.41 | 0.34 | 0.51 | 0.344 | 0.081 | 2630 |
| SVM | D & AL | 0.318 | 0.44 | 0.46 | 0.52 | 0.187 | 0.002 | 2548 |
| XGB | D | 0.366 | 0.48 | 0.51 | 0.53 | 1.046 | 0.005 | 2369 |
| XGB | D & AL | 0.367 | 0.49 | 0.53 | 0.55 | 4.498 | 0.005 | 2365 |
| RF | D | 0.389 | 0.50 | 0.51 | 0.54 | 0.184 | 0.015 | 2283 |
| RF | D & AL | 0.442 | 0.57 | 0.62 | 0.61 | 0.353 | 0.016 | 2085 |
| <i>DL architectures</i> | | | | | | | | |
| FC | D | 0.507 | 0.47 | 0.49 | 0.45 | 178.2 | 0.051 | 1842 |
| FC | D & AL | 0.666 | 0.64 | 0.66 | 0.62 | 839.4 | 0.138 | 1248 |
| FC | D & AL & CL | 0.586 | 0.53 | 0.51 | 0.55 | 1385.9 | 0.572 | 1547 |
| CNN | AM | 0.632 | 0.61 | 0.58 | 0.64 | 1089.0 | 0.481 | 1345 |
| <i>PPNM</i> | | | | | | | | |
| MH | D & AM & CM | 0.814 | 0.80 | 0.79 | 0.81 | 1728.4 | 3.472 | 695 |

We do not just want to have the [multi-headed NN] approach, but really looked forward to deployment of services. Without deployment, we can not generate the desired value.

(Project Manager)

5 Discussion and Implications

The presented method enables predictive end-to-end enterprise process network monitoring by leveraging a multi-headed NN architecture. Through a cross-organizational end-to-end view, interrelationships and dependencies between different departments, processes, and information systems can be jointly analyzed.

5.1 Critical Perspective on the PPNM Method

Through the first and last phase with particular focus on the organizational layers, we enable end-to-end analyses. Leveraging the multi-headed DNN architecture provides a scaleable solution to combine multiple data sources from across the organization and processes, each with specialized input heads. For the case study, we applied PPNM to a real-world use case and designed a three-headed DNN architecture with multi-log and context data input heads. Based on the numerical evaluation, combined with the employees' feedback, we can summarize that the PPNM

method helps guiding the development of predictive end-to-end enterprise process network monitoring.

Moreover, there are standard procedure models for data mining, such as CRISP-DM (Wirth and Hipp 2000), that someone may compare to our engineered method. Even though these procedure models work well for numerous use-cases in practical settings, they lack specifications and instructions for guiding the actual model design or combining multiple data sources, particularly considering the complex design process of a multi-headed neural network in an organizational context. For this purpose, the engineered PPNM establishes a more specialized perspective on defining the problem in the enterprise process network and particularly considers the combination of data sources in the design of a MH-NN with dedicated NN input heads.

Finally, considering the MH-NN, architecture alternatives may enhance predictive power. Thus, it may be worth comparing multiple architectures for the same input. We did so during the MH-NN design, resulting in the design with three customized heads. However, with ongoing advances in NN development, new layers or even (pre-trained) state-of-the-art methods may emerge. Thus, the chosen MH-NN should be regularly reviewed.

5.2 Concept Drift in the Enterprise Process Network

The fifth phase consists of the final step of model integration and operationalization in the enterprise process

network. It comprises the final online deployment, where (live) data sources are fed into the trained model for real-time predictions. With respect to the results, the prediction time of the MH model is worse compared to DL and ML or benchmark approaches. However, for the current use-case the prediction time is satisfying, whereas it may be optimization potential for future research. Once the predictive model has been put into production, it draws on the knowledge from the historical data used for training. Deployed models inevitably face the phenomenon of structural changes in data over time, which is referred to as *concept drift* and usually leads to a deterioration of the prediction performance. Maisenbacher and Weidlich (2017), Denisov et al. (2018) and Spenrath and Hassani (2020) mention respective observations in various organizational PPM contexts. Yet, the concept drift problem is neither limited to PPM, but also known in the more general fields of PM (Adams et al. 2021; de Sousa et al. 2021) and ML (Widmer and Kubat 1996).

For valid process predictions and analyses, the phenomenon of concept drift has to be detected and counteracted at an early stage. Currently the PPNM method, does not account for concept drift. To detect a concept drift, multiple methods are known (Seidl 2021; Kahani et al. 2021), such as local outlier detection, which can initiate retraining of the model with updated data to avoid wrong predictions and achieve temporal stability (Teinmaa et al. 2018).

5.3 Detailed Analytics vs. End-to-End Method

A common phenomenon of traditional enterprises with hierarchical organizational structures is silo thinking. The symptoms of it are weak collaboration throughout the organization. As a result, isolated process analysis within departmental boundaries is often observed, as there is little responsibility for end-to-end processes (Eggers et al. 2021). Nevertheless, a holistic view of the organization is necessary as processes often span several departments. Connected through information systems, inter-departmental information about processes is available. In this regard, digitalization and emerging technologies, such as PM or PPM, enable end-to-end insights into processes and a holistic view on the heterogeneous IT-landscape of enterprises (Armengaud et al. 2020). Both PM and PPM provide tools for generating insights on processes on an organizational scale, as they can process large amounts of data. For example, Lorenz et al. (2021) provide an end-to-end perspective for PM to improve the productivity in make-to-stock manufacturing processes, and Eggers et al. (2021) show how management decisions can drive an end-to-end perspective on process data by creating new process owner positions. However, the capability of end-to-end process

analysis is hardly considered in research as well as in practice.

Our proposed PPNM method contributes to this field of research by integrating the enterprise process network with all its interrelations and dependencies. In addition, for PPM as a subcategory of PM, our research has shown the benefits of taking an end-to-end view of processes for predictive tasks. The PPNM method and the fusion of inter-departmental data sources significantly increase the predictive power. This is already a first contribution, but it should not be the end of the research. Our approach for end-to-end PPNM is only an avenue towards general approaches for end-to-end PM. Therefore, future research should focus on leveraging the resources of the enterprise process network for PM and derive end-to-end insights.

6 Conclusion and Outlook

We present the PPNM method, for end-to-end enterprise process network monitoring, leveraging a MH-NN approach. In doing so, we overcome the phenomenon of silo-thinking and separated analysis of in data sources, as we enable the seamless combination of multiple data sources, combined with specialized processing and NN computation for each input. The resulting MH-NN outperforms classical ML and DL models and was applied and evaluated in an organizational context.

From a more general perspective, the method is an essential piece of research, enabling end-to-end PPNM on an organizational scale. Further, it guides the path towards a more general end-to-end PM, which then overcomes silo-thinking and enables an organization's enterprise process network's potential (van der Aalst 2021). However, the approach is not limited to single organizations. Due to the method's extend-ability, additional data sources, even across multiple organizations, could be combined and leveraged each best. Thus, we further contribute to research towards holistic supply chain analytics. Respective inter-organizational PM analyses are proposed by Hernandez-Resendiz et al. (2021) for descriptive supply chain analytics, yet predictive insights are neglected. Our research extends the scope and enables the inter-organizational combination of data, even for predictive tasks. With larger data integrated, additional analytics research streams such as federated learning or aspects such as data ownership become more relevant and should be investigated in future research. The transfer of improved process predictions within and across organizations is not only relevant for research, but especially for enterprises by means of scaling the respective solutions. Thus, our method not only enables new research but could be a fundamental component for

scaleable enterprise-ready PPNM solutions with heterogeneous intra- and inter-organizational data sources.

Funding Open Access funding enabled and organized by Projekt DEAL.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s12599-022-00778-4>) contains supplementary material, which is available to authorized users.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adams JN, Zelst SJv, Quack L, Hausmann K, van der Aalst WM, Rose T (2021) A framework for explainable concept drift detection in process mining. In: Proceedings of the international conference on business process management. Springer, Heidelberg: 400–416
- Armengaud E, Fruhwirth M, Rothbart M, Weinzerl M, Zembacher G (2020) Digitalization as an opportunity to remove silo-thinking and enable holistic value creation. *Syst Eng Automot Powertrain Dev*, 1–28
- Beheshti A, Benatallah B, Motahari-Nezhad HR (2018) ProcessAtlas: a scalable and extensible platform for business process analytics. *Softw Pract Exp* 48(4):842–866
- Benatallah B, Sakr S, Grigori D, Motahari-Nezhad HR, Barukh MC, Gater A, Ryu SH et al (2016) Process analytics: concepts and techniques for querying and analyzing process data. Springer Cham
- Benscoter B (2012) How to identify and analyze problems in your organization: 290–294. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118364727.ch29>
- Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J Mach Learn Res* 13(2):281–305
- Borkowski M, Fdhila W, Nardelli M, Rinderle-Ma S, Schulte S (2019) Event-based failure prediction in distributed business processes. *Inf Syst* 81:220–235
- Breuker D, Matzner M, Delfmann P, Becker J (2016) Comprehensive predictive models for business processes. *MIS Q* 40(4):1009–1034
- Brinkkemper S (1996) Method engineering: engineering of information systems development methods and tools. *Inf Softw Technol* 38(4):275–280
- Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. *Adv Neural Inf Process Syst* 33:1877–1901
- Brownlee J (2017) Long short-term memory networks with python. *Machine Learning Mastery*
- Brownlee J (2020) Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning. *Machine Learning Mastery*
- Brunk J, Stottmeister J, Weinzierl S, Matzner M, Becker J (2020) Exploring the effect of context information on deep learning business process predictions. *J Decis Syst* 29(sup1):328–343
- Canizo M, Triguero I, Conde A, Onieva E (2019) Multi-head CNN-RNN for multi-time series anomaly detection: an industrial case study. *Neurocomputing* 363:246–260
- Chollet F (2018) Deep learning mit python und keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. Mitp, Frechen
- Cuzzocrea A, Folino F, Guarascio M, Pontieri L (2019) Predictive monitoring of temporally-aggregated performance indicators of business processes against low-level streaming events. *Inf Syst* 81:236–266
- Da Cunha Mattos T, Santoro FM, Revoredo K, Nunes VT (2014) A formal representation for context-aware business processes. *Comput Ind* 65(8):1193–1214
- Dai Z, Liu H, Le QV, Tan M (2021) Coatnet: marrying convolution and attention for all data sizes. *Adv Neural Inf Process Syst* 34:3965–3977
- de Sousa RG, Peres SM, Fantinato M, Reijers HA (2021) Concept drift detection and localization in process mining: an integrated and efficient approach enabled by trace clustering. In: Proceedings of the 36th annual ACM symposium on applied computing: 364–373
- Denisov V, Belkina E, Fahland D (2018) Mining concept drift in performance spectra of processes. In: Proceedings of the 8th international business process intelligence challenge
- Di Francescomarino C, Dumas M, Federici M, Ghidini C, Maggi FM, Rizzi W (2016) Predictive business process monitoring framework with hyperparameter optimization. In: Nurcan S, Soffer P, Bajec M, Eder J (eds) Proceedings of the advanced information systems engineering, 361–376
- Di Francescomarino C, Ghidini C, Maggi FM, Petrucci G, Yeshchenko A (2017) An eye into the future: leveraging a-priori knowledge in predictive business process monitoring. In: Proceedings of the international conference on business process management. Springer, Heidelberg, 252–268
- Dumas M, La Rosa M, Mendling J, Reijers HA (2018) Introduction to business process management. In: Fundamentals of business process management. Springer, Berlin, Heidelberg, 1–33
- Eggers J, Hein A, Böhm M, Krmar H (2021) No longer out of sight, no longer out of mind? How organizations engage with process mining-induced transparency to achieve increased process awareness. *Bus Inf Syst Eng* 63:491–510
- Evermann J, Rehse JR, Fettke P (2016) A deep learning approach for predicting process behaviour at runtime. In: Proceedings of the international conference on business process management. Springer, Heidelberg, 327–338
- Eversheim W (2013) Prozessorientierte Unternehmensorganisation: Konzepte und Methoden zur Gestaltung, schlanker“ Organisationen. Springer, Heidelberg
- Feurer M, Klein A, Eggensperger K, Springenberg JT, Blum M, Hutter F (2019) Auto-sklearn: efficient and robust automated machine learning. In: Automated machine learning, In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham, 113–134
- Flath CM, Stein N (2018) Towards a data science toolbox for industrial analytics applications. *Comput Ind* 94:16–25
- Folino F, Guarascio M, Pontieri L (2012) Context-aware predictions on business processes: an ensemble-based solution. In:

- Proceedings of the international workshop on new frontiers in mining complex patterns. Springer, 215–229
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
- Gupta D, Prakash N (2001) Engineering methods from method requirements specifications. *Requir Eng* 6(3):135–160
- Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier MK, Amsterdam
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 770–778
- Heinrich K, Zschech P, Janiesch C, Bonin M (2021) Process data properties matter: introducing gated convolutional neural networks (GCNN) and key-value-predict attention networks (KVP) for next event prediction with deep learning. *Decis Support Syst* 143(113):494
- Hernandez-Resendiz JD, Tello-Leal E, Marin-Castro HM, Ramirez-Alcocer UM, Mata-Torres JA (2021) Merging event logs for inter-organizational process mining. In: Zapata-Cortes, J.A., Alor-Hernández, G., Sánchez-Ramírez, C., García-Alcaraz, J.L. (eds) *New Perspectives on Enterprise Decision-Making Applying Artificial Intelligence Techniques*. Studies in Computational Intelligence, vol 966. Springer, Cham, 3–26
- Horn M, Moor M, Bock C, Rieck B, Borgwardt K (2020) Set functions for time series. In: Proceedings of the international conference on machine learning. PMLR, 4353–4363
- James G, Witten D, Hastie T, Tibshirani R (2017) An introduction to statistical learning: with applications in R. Springer New York
- Jia Y, Zhang Q, Zhang W, Wang X (2019) Communitygan: community detection with generative adversarial nets. In: Proceedings of the The World Wide Web conference, 784–794
- Jiang H, He P, Chen W, Liu X, Gao J, Zhao T (2019) Smart: robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. arXiv:1911.03437
- Kahani M, Behkamal B et al (2021) Concept drift detection in business process logs using deep learning. *Signal Data Process* 17(4):33–48
- Kohzadi N, Boyd MS, Kermanshahi B, Kaastra I (1996) A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing* 10(2):169–181
- Komer B, Bergstra J, Eliasmith C (2019) Hyperopt-sklearn. In: *Automated machine learning. The Springer Series on Challenges in Machine Learning*. Springer, Cham: 97–111
- Kratsch W, Manderscheid J, Röglinger M, Seyfried J (2020) Machine learning in business process monitoring: a comparison of deep learning and classical approaches used for outcome prediction. *Bus Inf Syst Eng* 63:261–276
- Kraus M, Feuerriegel S, Oztekin A (2020) Deep learning in business analytics and operations research: models, applications and managerial implications. *Eur J Oper Res* 281(3):628–641
- Lakshmanan GT, Shamsi D, Doganata YN, Unuvar M, Khalaf R (2015) A markov prediction model for data-driven semi-structured business processes. *Knowl Inf Syst* 42(1):97–126
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
- Lin Y, Meng Y, Sun X, Han Q, Kuang K, Li J, Wu F (2021) Bertgcn: transductive text classification by combining GCN and BERT. arXiv:2105.05727
- Lorenz R, Senoner J, Sihn W, Netland T (2021) Using process mining to improve productivity in make-to-stock manufacturing. *Int J Prod Res* 59:4869–4880
- Maggi FM, Di Francescomarino C, Dumas M, Ghidini C (2014) Predictive monitoring of business processes. In: Proceedings of the international conference on advanced information systems engineering, 457–472
- Maisenbacher M, Weidlich M (2017) Handling concept drift in predictive process monitoring. *SCC* 17:1–8
- Márquez-Chamorro AE, Resinas M, Ruiz-Cortés A (2017) Predictive monitoring of business processes: a survey. *IEEE Trans Serv Comput* 11(6):962–977
- Mehdiyev N, Evermann J, Fettke P (2017) A multi-stage deep learning approach for business process event prediction. In: Proceedings of the 19th IEEE conference on business informatics, vol 1, 119–128
- Mehdiyev N, Evermann J, Fettke P (2020) A novel business process prediction model using a deep learning method. *Bus Inf Syst Eng* 62(2):143–157
- Mo H, Lucca F, Malacarne J, Iacca G (2020) Multi-head CNN-LSTM with prediction error analysis for remaining useful life prediction. In: Proceedings of the 27th conference of Open Innovations Association (FRUCT). IEEE, 164–171
- Moon J, Park G, Jeong J (2021) Pop-on: prediction of process using one-way language model based on NLP approach. *Appl Sci* 11(2):864
- Oberdorf F, Schaschek M, Stein N, Flath C (2021a) Neural process mining: multi-headed predictive process analytics in practice. *ECIS 2021 Research Papers*
- Oberdorf F, Stein N, Flath CM (2021b) Analytics-enabled escalation management: system development and business value assessment. *Comput Ind* 131:103481
- O'Malley T, Bursztein E, Long J, Chollet F, Jin H, Invernizzi L et al (2019) Keras tuner. <https://github.com/keras-team/keras-tuner>
- Papers with Code (2021) Browse state-of-the-art. <https://paperswithcode.com/sota>. Accessed 23 Sept 2021
- Park G, Song M (2020) Predicting performances in business processes using deep neural networks. *Decis Support Syst* 129:113–191
- Pasquardibisceglie V, Appice A, Castellano G, Malerba D (2019) Using convolutional neural networks for predictive process analytics. In: Proceedings of the international conference on process mining (ICPM). IEEE, 129–136
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
- Polyvyanyy A, Ouyang C, Barros A, van der Aalst WM (2017) Process querying: enabling business intelligence through query-based process analytics. *Decis Support Syst* 100:41–56
- Rama-Maneiro E, Vidal JC, Lama M (2021) Deep learning for predictive business process monitoring: review and benchmark. *IEEE Trans Serv Comput* (1)
- Rogge-Solti A, van der Aalst WMP, Weske M (2013) Discovering stochastic petri nets with arbitrary delay distributions from event logs. In: Proceedings of the international conference on business process management. Springer, Heidelberg, 15–27
- Rosemann M, Recker J, Flender C (2008) Contextualisation of business processes. *Int J Bus Process Integr Manag* 3(1):47–60
- Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science
- Scheer AW (2013) ARIS - vom Geschäftsprozess zum Anwendungssystem
- Schwegmann B, Matzner M, Janiesch C (2013) A method and tool for predictive event-driven process analytics. In: Proceedings of the Wirtschaftsinformatik. Citeseer, p 46
- Seidl T (2021) Concept drift detection on streaming data with dynamic outlier aggregation. In: Proceedings of the process mining workshops: ICPM 2020 international workshops, vol 406. Springer Nature, p 206
- Senderovich A, Di Francescomarino C, Ghidini C, Jorbina K, Maggi FM (2017) Intra and inter-case features in predictive process

- monitoring: a tale of two dimensions. In: *Proceedings of the international conference on business process management*. Springer, 306–323
- Senderovich A, Di Francescomarino C, Maggi FM (2019) From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring. *Inf Syst* 84:255–264
- Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. *Adv Neural Inf Process Syst* 25: 2951–2959
- Spennath Y, Hassani M (2020) Predicting business process bottlenecks in online events streams under concept drifts. In: *Proceedings of the Ecms*, 190–196
- Stierle M, Weinzierl S, Harl M, Matzner M (2021) A technique for determining relevance scores of process activities using graph-based neural networks. *Decis Support Syst* 144(113):511
- Teinmaa I, Dumas M, Leontjeva A, Maggi FM (2018) Temporal stability in predictive process monitoring. *Data Min Knowl Discov* 32(5):1306–1338
- Theis J, Darabi H (2019) Decay replay mining to predict next process events. *IEEE Access* 7:119,787–119,803
- van der Aalst WMP (2011) Process mining: discovering and improving spaghetti and lasagna processes. In: *Proceedings of the IEEE symposium on computational intelligence and data mining*. IEEE, 1–7
- van der Aalst WMP (2016) *Data Science in Action*. In: *Process Mining*. Springer, Berlin, Heidelberg
- van der Aalst WMP (2021) Federated process mining: exploiting event data across organizational boundaries. In: *Proceedings of the IEEE international conference on smart data services (SMDS)*. IEEE, 1–7
- van der Aalst WMP, Pesic M, Song M (2010) Beyond process mining: from the past to present and future. In: *Proceedings of the international conference on advanced information systems engineering*. Springer, Heidelberg, 38–52
- van der Aalst WMP, Adriansyah A, De Medeiros AKA, Arcieri F, Baier T, Blicke T, Bose JC, Van Den Brand P, Brandtjen R, Buijs J et al (2011a) Process mining manifesto. In: *Proceedings of the international conference on business process management*. Springer, Heidelberg, 169–194
- van der Aalst WMP, Schonenberg MH, Song M (2011b) Time prediction based on process mining. *Inf Syst* 36(2):450–475
- Van Dongen BF, Crooy RA, van der Aalst WM (2008) Cycle time prediction: When will this case finally be finished? In: *Proceedings of the OTM confederated international conferences” on the move to meaningful internet systems”*, 319–336
- Vera-Baquero A, Colomo-Palacios R, Molloy O (2013) Business process analytics using a big data approach. *It Prof* 15(6):29–35
- Verbitskiy S, Vyshegorodtsev V (2021) ERANNs: efficient residual audio neural networks for audio pattern recognition. arXiv:2106.01621
- Verenich I, Dumas M, Rosa ML, Maggi FM, Teinmaa I (2019) Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Trans Intell Syst Technol: TIST* 10(4):1–34
- vom Brocke J, Rosemann M (2014) *Handbook on business process management 1: introduction, methods, and information systems*. Springer, Heidelberg
- Wahid NA, Adi TN, Bae H, Choi Y (2019) Predictive business process monitoring-remaining time prediction using deep neural network with entity embedding. *Procedia Comput Sci* 161:1080–1088
- Wan R, Mei S, Wang J, Liu M, Yang F (2019) Multivariate temporal convolutional network: a deep neural networks approach for multivariate time series forecasting. *Electron* 8(8):876
- Wang R, Li B, Hu S, Du W, Zhang M (2019) Knowledge graph embedding via graph attenuated attention networks. *IEEE Access* 8:5212–5224
- Widmer G, Kubat M (1996) Learning in the presence of concept drift and hidden contexts. *Mach Learn* 23(1):69–101
- Wirth R, Hipp J (2000) CRISP-DM: towards a standard process model for data mining. In: *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, Manchester, vol 1: 29–39
- Yeshchenko A, Durier F, Revoredo K, Mendling J, Santoro F (2018) Context-aware predictive process monitoring: the impact of news sentiment. In: *Proceedings of the OTM confederated international conferences” on the move to meaningful internet systems”*. Springer, Heidelberg, 586–603
- Zhang Z, Bu J, Ester M, Zhang J, Yao C, Yu Z, Wang C (2019) Hierarchical graph pooling with structure learning. arXiv:1911.05954
- Zhao B, Lu H, Chen S, Liu J, Wu D (2017) Convolutional neural networks for time series classification. *J Syst Eng Electron* 28(1):162–169
- Zhu Y, Zhuang F, Wang J, Chen J, Shi Z, Wu W, He Q (2019) Multi-representation adaptation network for cross-domain image classification. *Neural Netw* 119:214–221
- Zur Muehlen M, Shapiro R (2015) *Business process analytics*. In: *Handbook on business process management*, vol 2, Springer-Verlag Berlin Heidelberg, 243–263