



Deep Graph Representation Learning for Business Process Modeling

Chaima Afifi and Ali Khebizi

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

November 19, 2022

Deep Graph Representation Learning for Business Process Modeling

AFIFI CHAIMA*

Departement of Computer science, Labstic Laboratory, University of 8 Mai 45 - Algeria-
afifi.chaima@univ-guelma.dz

KHEBIZI ALI

Departement of Computer science, Labstic Laboratory, University of 8 Mai 45 - Algeria-
khebizi.ali@univ-guelma.dz

November 18, 2022

Abstract

Business process (BP) models can quickly become complex and expensive. In turn, the abstraction has proved to be a challenging key for establishing a comprehensible and high-level view of the BP model. Where the aggregated processes are preserved and irrelevant details are omitted. The promising research question explores the reasonable stones on merging and validating the produced high-level model. The semantic BP logic in its turn, is a cornerstone of extra-knowledge that contributes in the development of the ideal BP high abstraction model.

This study focuses on the BP abstraction problem. Furthermore, with the remarkable development in artificial intelligence (AI) techniques in the context of business process mining, BP models can be retrieved from execution data utilizing deep learning (DL) approaches in general, and Deep Graph Representation Learning (DGRL) in particular. This study emphasizes the unavailability of a DGRL model that generates a BP model from execution traces. Finally, a roadmap for future research directions is proposed.

Keywords : *Business Process, Abstraction level, Graph theory, Graph-based-modeling, Deep graph representation learning.*

I. INTRODUCTION

Business process models are developed in a broad variety of organizational initiatives [37], which may swiftly become complex and expensive owing to the massive amount of data generated by various Information systems (IS). Human consumers, on the other hand, have limited cognitive ability to make sense of huge and complex business process models. One well-known approach to address this issue is to use abstraction[9] by keeping the relevant information while leaving out the insignificant details. Indeed, in BP, abstraction served as the foundation for achieving diverse business goals. The adaptation of the BP supported by company employees, the combination of two or more processes together according to their similar goals, the collaboration of process parts with other companies to be collaborated together, and the elimination of redundant processes are the pillar challenges toward high-abstraction model that aim to optimize the complex BP

*Corresponding author

executed on the real firm. In such situation, the high abstract models should have a coherent semantic groups that aggregate information from lower-level activities.

In fact, while it has been established that abstraction may significantly improve the sense of complex process models, there are limited insights into the criteria adopted by different experienced designers to determine which activities might be aggregated into a new one. In this instance, the semantic properties of the Event-Log data, such as the performance parameter, may be exploited to semantically aggregate information. In this area of interest, a variety of approaches have been presented that take into account a wide range of BP characteristics. Two tasks, for example, can be grouped if they share documentation, resources, and are carried out by the same role. However, each of such features has its own view to aggregate information and create the high abstraction level. Searching this high combination of features lead to the abstract level is challenge.

In this research, we address the above challenge by introducing an abstraction approach that incorporates the semantic component included inside each feature of the event-log in order to provide a high Abstraction model that aggregates the collection of sub-views of each feature. The key focus is to discover the best semantic concealed in the traces by searching for an optimal arrangement of characteristics. This leads to a high degree of abstraction model.

On top of that, with the upsurge of Artificial Intelligence (AI) [27] techniques in general, and machine learning (ML) [24] as well as Deep Learning (DL) [38] in particular, the BP research was shoot up. Where, Graph Theories (GT) demonstrate substantial limitations despite all of its strengths and consistent typologies[30]. The Poor interoperability of the abstract graph models are the major drawbacks. As a consequence, Deep Graph representation learning (DGRL) [4] serves as a foundation for the Graph-based-BP modeling approach. Whereas no DGRL model creates a BP model from execution traces.

This study focuses on the BP abstraction problem. Furthermore, it emphasizes the unavailability of a DGRL model that generates a BP model from execution traces. where a road-map toward the future research direction are presented.

The remainder of the paper is organized as follows. The second section introduces basic concepts and definitions of business processes Abstraction and the DGRL. In section 3, we expose a junction between BP and DGRL. Section 4 is dedicated to present the Abstraction gap on the BP-DGRL. Section 6 concludes the paper and it present future perspectives.

II. BACKGROUND AND PRELIMINARIES

In this section, the BP's concepts and tools are introduced. The relevance of abstract specifications in BP modeling is also emphasized. Then, the DGRL approaches and architectures are presented.

i. BP Concepts

The astonishing growth of connectivity and broad usage of the internet have forced companies to execute their BP in a more competitive environment in which they must exhibit a high level of adaptability and flexibility [7]. As a consequence, regardless of their activities nature, businesses must be conducted with a high and quick level of reconfiguration that allows constructing new BP and updating the already deployed ones. In this viewpoint, the BP paradigm becomes an unavoidable tool which constitutes the cornerstone of any organization, because it offers a simplified representation of business rules while ensuring the effective resources usage. This ambition has completely altered the way in which software and IS are delivered to end users. Hence, the development of BPM technology is promoted as a comprehensive and systematic strategy,

methodologies, and techniques for radically changing the perception of enterprises' businesses. Actually, it's observed that business protocol managers acts by identifying, modeling, analyzing, improving, and optimizing their business processes in an automata fashion and by using modern BP environments and adequate software suites.

In what follows, we offer a detailed overview of BP and we discuss the importance of employing BP models inside a corporation. We also provide explicit definitions of BP events and execution traces.

Definition II.1 (Business Process). *A BP is a set of activities undertaken by one or more organizations in coordination for achieving some particular business goals [34].*

As an illustration, booking a trip, ordering goods or processing citizens' retirement applications are all real-life BP that are made up of a series task and where each BP is governed by its intrinsic business rules representing the company's logic.

To make management and maintenance of BPs easier, the **Business Process Management (BPM)** technology stipulates that BPs must satisfy a predefined life-cycle principle. The BP life cycle [35] is articulated around four consecutive steps including design, model, execution and the last step aims to monitor and optimize the process. During each stage, specific activities are carried out in order to achieve the most efficient way to conduct business operations.

From an operational perspective, **Business Process Management Systems (BPMS)** [8] are used as software tools to manage these critical phases of the BP where a large panoply of mature industrial technologies can be exploited for creating, automating, and evaluating BP.

In depth, the BP models produced in the first step of the BPM life-cycle exemplify the crucial concept in the BPM ecosystem, since they are abstract specifications that describe the business logic supported by the present procedure of the firm. The next stages of the life-cycle highlight critical complimentary stages relevant for deploying, analyzing, monitoring, and improving the models and their related data to ensure accurate BP agility.

To represent such BP, the literature extensively uses formal, abstract, and graphical tools such as graphs, Petri-nets, FSM, and UML diagrams to encapsulate a set of constraints such as order and time constraints [1]. Given this variety of BP representation models, picking the most expressive one is crucial task.

The set of such BP-models are represented to define the relationship between a set of events II.2 throughout all the event-log data trace's II.3 described by this set of properties:

- **The trace identification:** indicates a reference (ID) of the trace having created the event;
- **The activity name:** designates the activity responsible for the event to which it relates;
- **The time-stamp:** indicates the activity's start and completion time.

Definition II.2 (Event). An event e is a tuple $(a, c, t, (d_1, v_1), \dots, (d_m, v_m))$, where a is the activity name, c is the trace ID, t is the time-stamp and $(d_1, v_1), \dots, (d_m, v_m)$ where $m > 0$ are a set of attributes and their values [3].

Definition II.3 (Trace). A trace \mathcal{T} having an $ID = c$ is a non-empty sequence $[e_1, e_2, \dots, e_n]$ of events such that:

$(\forall i \text{ and } j \in [1 \dots n], \text{with } e_i \in \mathcal{T} \text{ and } e_j \in \mathcal{T}: \text{thus } e_i.c = e_j.c).$

In other words, all the events in the same trace are characterized by the same ID [3].

where, the detailed examples is on our previous paper[3]. After describing the important aspects of the BP, the next section provides a broad overview of the Deep graph representation techniques for mining the BP.

ii. DGRL concepts

Mining BPs models basing on their execution traces for building the corresponding graph-model can be perceived as an incremental and continuous process and, thus, conducted under machine learning aspects. From this point of view, graphs and consequently BPs modeling have taken full advantage of the achievements induced in recent years by the spectacular explosion of Artificial Intelligence (IA) techniques, particularly, the advances made in the field of deep learning.

In a nutshell, deep learning architectures allow developing systems that can learn, reason, and generalize knowledge from data. Thus, techniques for deep graph embedding [2][12], expansions of convolution neural networks to graph-structured data [40], and neural message-passing systems inspired by belief propagation [22] have emerged and proved their effectiveness in various fields related to knowledge representation and data science.

Such achievements in graph representation learning have resulted in new state-of-the-art findings in a variety of fields, including chemical synthesis, 3D vision, recommend-er systems as well as social network analysis.

The field of BP-mining has not been spared by the ubiquity of AI techniques and it was strongly impacted by the made progress. In fact, during BP mining -based deep learning techniques- the BP graph-model is **generated** progressively as training data are introduced.

The purpose of BP graph generation is to extract the knowledge stored in organisations' information systems as log files, by creating models capable of reproducing facts and events observed in the real-world. Thus, instead of assuming that we are provided a graph structure \mathcal{G} as an input of our system, we want our system's output to create the target graph \mathcal{G} .

Obviously, merely creating an arbitrary graph is not inherently a difficult task. In what follows, an overview of approaches used for generating graph is presented and the focus is made on IA supported approaches.

ii.1 Conventional approaches

Historically, **traditional graph generating algorithms** [14, 16, 21] were useful in a variety of contexts, such as transport, urban and computer networks and were used to efficiently construct synthetic graphs with specific properties (*nodes order, precedence, adjacency,...*). Further, they provide insight into how particular graph topologies may occur in the real- world. These techniques, however, have a significant disadvantage related to a predefined and hand-crafted generating process. In short, traditional approaches can build graphs but cannot train a generative model from data.

To overcome this difficulty, authors in [20] propose a set of methods to develop a generative graph model from a set of training graphs by creating models that can observe a group of graphs and learn to construct graphs that are similar to the training set. These methods avoid manually coding certain attributes into a generative model, such as the events and performance attributes of the event-Log data. The described methods are called **DGRL** and encompass the three following most common approaches for developing generic deep generative models [13] (i) Variational auto-encoders (VAEs), (ii) Generative adversarial networks (GANs), and (iii) Auto-regressive models.

These approaches are presented and discussed bellow.

ii.2 The DGRL approaches

The DGRL is the key to overcome the different drawbacks of the conventional graph theory. This section explain a different architectures of the DGRL.

a. Variational Auto-Encoders (VAEs)

VAEs are one of the most widely used methods for creating deep generative models [18]. The primary concept behind the VAE models for graphs is to build a full graph all at once in an auto-encoder way. These modeling tools are a popular framework for deep generative models, not only for graphs but also for photos, text, and a broad range of data formats. Also, they have a well-defined probabilistic rationale and numerous research works, such as [6, 25], exploit and evaluate the structure of the latent spaces learnt by VAE models.

As shown in Figure 1, the goal is to train a probabilistic decoder model from which we may sample actual graphs (*i.e.*, adjacency matrices) using a latent variable Z . In a probabilistic sense, we want to train a conditional distribution over adjacency matrices (*with the distribution's start conditional on some latent variable*). The basic idea consists to maximize the reconstruction ability of our decoder while minimizing the Kullback-Leibler-Loss (*KL-divergence*) [ref] between our posterior latent distribution q and the prior one $p(Z)$.

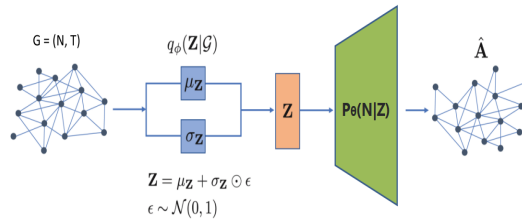


Figure 1: Illustration of a standard VAE model applied to the graph setting [18]

b. Generative Adversarial Networks (GANs)

VAEs are known to have major limitations, such as the tendency for VAEs to create fuzzy image outputs. Many current cutting-edge generative models make use of alternative generating frameworks and Generative Adversarial Networks (GANs) being one of the most common in this field [11]. As shown in the 2, the basic idea behind a general GAN-based generative models consists to define a trainable generator network which is trained to generate realistic (*but fake*) data samples by taking a random data-set as input (*e.g.*, a sample from a normal distribution). At the same time, a discriminator network is specified and its goal is to distinguish between real data samples and samples generated by the generator. In such systems, we will assume that discriminator outputs the probability that a given input is fake.

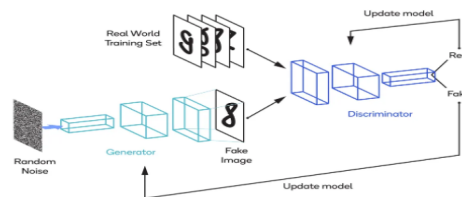


Figure 2: The GAN architecture [31]

b. Auto-Regressive Models (ARM)

The preceding basic GAN and VAE-based techniques employed simple Multi-Layer Perception (MLP) mechanism to produce adjacency matrices that to be used for generating graphs. In

what follows, we will look at more advanced auto-regressive algorithms [15] for decoding graph topologies from latent representations that describe an observation in some compressed representation.

Figure 3 below shows how the input image is described in terms of its latent attributes, with a single value describing each descriptive properties. However, it is possible that we would prefer to represent each latent attribute as a set of possible values. For example, if you fed in a photo of the Mona Lisa, what single value would you assign to the smile attribute? We can describe latent attributes in probabilistic terms using an ARM.

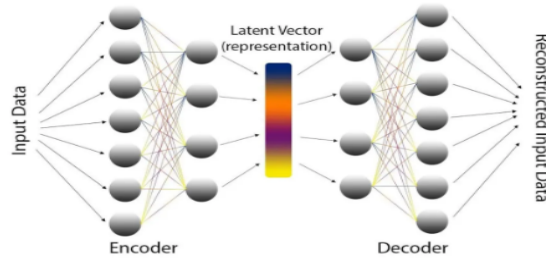


Figure 3: A latent representation illustration [15]

On such DGRL approaches, making the appropriate representation between the input, output, and latent representation is one of the DGRL architectures' pillars. In this scenario, incorporating such generative models into the text application domain necessitates a few key adjustments. Word embedding, which is merely a fancy way of stating numerical representation of words, is one of the advances of such DGRL architectures. Continuous bag of words[29], skip-gram[28] are the top of such word embedding models.

Following the introduction of the basic materials for understanding DGRL, we will concentrate on their use in the context of BP modeling. When text mining word embedding DGRL algorithms were coupled with BP field notions.

III. BP-DGRL MODELS

After having introduced the DGRL and their different variants, at this level of our development it is essential to ask the following question: What are the contributions of GRL for the field of BP and what are their impacts on graphs-BP?

In a matter of fact, the Process mining has been a popular tool for companies examining their business processes in recent years. They may be used to evaluate event logs produced by process-aware information systems to provide important insights into how a business process is really carried out. Process mining approaches, on the other hand, typically focus on a process's control-flow without taking into account the context in which a case is performed, such as the department, product, customer, or other qualities an event gives. This extra process context may aid process mining approaches by revealing trends in the event log that aren't obvious from a control-flow perspective.

From point of view, many approaches in process mining, such as trace clustering [23], prediction [32], and anomaly detection [26], need the vector representation of cases. By grouping comparable events, trace clustering tries to enhance the finding of process models. It is possible to create clusters of instances that are performed in comparable situations, allowing the user to compare

process models from various scenarios. Improved prediction models that take into account the process contexts can be learnt. Anomaly detection approaches based on extended vector representations, on the other hand, can yield more dependable findings. These are only a few instances of context’s potential applications, which also include vector representations.

The exploration of the research literature shows that application of GRL progress to the domain of BP modeling is still extremely rare. The only five works encountered on the graph representation learning and which are closely related to BP modeling are Act2vec, Log2vec, trace2vec, Model2vec and Case2vec are the only challenge works. These works typically aim to turn raw event data into meaningful insights, activities or models [19]. In what follows these models are presented.

i. act2vec: Obtaining Representations of Activities

The first representation learning architecture relates to deriving representations of activities from execution traces. For doing so, we assume that we have input data in the form of an event log containing activities’ sequences of deployed BPs. In line with the word2vec [5] approach used in natural language processing, we can learn representations of activities by considering them as words in a corpus. In our context the corpus being the event log of BPs.

Similarly, the DGRL learns representations for activities by handling activities as words in the event-log dataset. As seen in Fig.4, the default design for act2vec is analogous to the CBOW paradigm for building word representations [10]. The CBOW neural network design is predicated on the idea that a word may be anticipated depending on its context (i.e. the words appearing before and after the focus word). Act2vec [17] considers traces to be sentences and events to be words. As a result, they develop general-purpose representations of activities that are not customized to, say, anticipating the next activity or the remaining duration of occurrences.

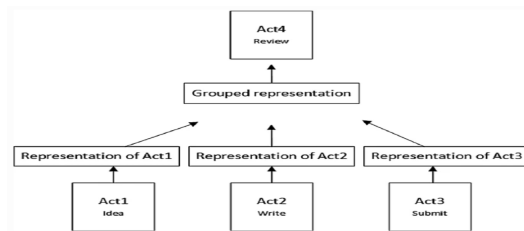


Figure 4: The act2vec-architecture for learning vector representations of activities. The context consisting of activities “idea”, “write”, and “submit” is used to predict activity “review” [17]

ii. trace2vec: Obtaining Representations of Traces

Following the similarity between activities and words, traces may also be thought of as sentences. And according to [36], the doc2vec strategy to learning distributed representations of sentences, paragraphs, or documents has been developed in the natural language processing sector. Such creators of doc2vec enhance the CBOW architecture [29] with a paragraph vector, resulting in the so-called Distributed Memory Model of Paragraph Vectors (PV-DM)[39]. This concept may be applied to traces as well, resulting in the trace2vec architecture seen in Fig.5. Given that this architecture contains a representation of traces (based on the trace identifier), it will enable cooperative learning of activity and trace representations. We are obviously mainly interested in the trace representations for further research.

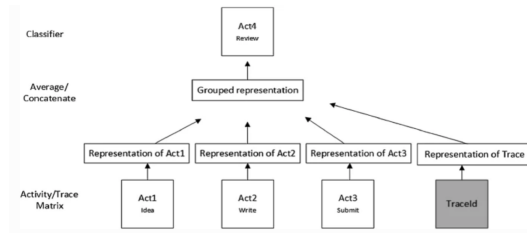


Figure 5: The *trace2vec*-architecture for learning vector representations of traces. The context consisting of activities “idea”, “write”, and “submit”, as well as the trace-id, is used to predict activity “review” [17]

iii. log2vec: Obtaining Representations of Logs

As such, in line with *trace2vec*, an architectural design can be devised to learn distributed representations of logs. A simple method [17] could be to replace the representation of a trace with a representation of a log in Fig.6. Though given that such a setup architecture would be unable to incorporate trace-level information, and thus will consider an event log as a set of ungrouped activities, it makes sense from a BP perspective to extend the architecture to also include trace information. To do so, we could simply use the trace identifier as before, however, given that business processes could share similar execution variants, we propose to include an artificial identifier relating to distinct process instances in the architecture (i.e. a trace “variant” identifier). More specifically, all traces from the different event logs under consideration are joined into one event log based on that input, a distinct process instance identifier is computed.

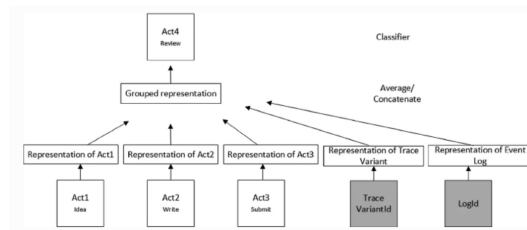


Figure 6: The *log2vec*-architecture for learning vector representations of logs. The context consisting of activities “idea”, “write”, and “submit”, an identifier for each trace variant, as well as the log-id from which the words are sampled, is used to predict activity “review”. [17]

IV. ABSTRACTION LEVEL IS THE LACUNA OF BP-DGRL MODELS

Process abstraction is a method of giving alternative process views (which maintain information useful for a specific purpose) and lowering the size and complexity of process models by keeping fundamental attributes while removing irrelevant elements [33]. It may be used to focus on certain process model attributes (for example, preserving expensive/frequent/long activities), change process model for an external partner, trace data/task relationships, and acquire a process fast view while maintaining ordering constraints/roles. Alternatively, an analyst may be interested in tasks that take up more time than others in the process.

However, despite the pivotal role of the DGRL on the BP, where several advanced architectures were established, none of the studies use the DGRL to develop a simple straight-forward BP-Model from event-log data. From this perspective, different future research directions appears.

The following are the keystone ones:

- **Graph event-log representation** Which is the best BP-model representation that can be used by the DGRL model?
- **Graph2Graph using DGRL model** How can this representation can be exploited by the DGRL model to tackle the abstraction issue?

V. CONCLUSION

Business process models can quickly become complex and expensive. In turn, the abstraction has proved to be a challenging key for establishing a comprehensible and high-level view of the BP model. This study proved that the BP abstraction problem is open research question. Furthermore, it emphasizes the unavailability of a DGRL model that generates a BP model from execution traces. Whereas, *building a customized DGRL architecture for the BP abstraction level still an open research question*

REFERENCES

- [1] Bouafia.al. Hypergraph application on business process performance. *Information*, 12(9):370, 2021.
- [2] Cai.al. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.
- [3] Afifi Chaima and Ali KHEBIZI. A road-map for mining business process models via artificial intelligence techniques. *International Journal of Informatics and Applied Mathematics*, 5(1):27–51.
- [4] Ines Chami. *Representation Learning and Algorithms in Hyperbolic Spaces*. Stanford University, 2021.
- [5] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.
- [6] Cosmo.al. Limp: Learning latent shape representations with metric preservation priors. In *European Conference on Computer Vision*, pages 19–35. Springer, 2020.
- [7] M Dachyar, Teuku Yuri M Zagloel, and L Ranjaliba Saragih. Knowledge growth and development: internet of things (iot) research, 2006–2018. *Heliyon*, 5(8):e02264, 2019.
- [8] dos Santos Ferreira.al. Roadmap for operational sustainability: an approach based on integrated management systems and business process management. *Revista Gestão da Produção Operações e Sistemas*, 16(1), 2021.
- [9] Amal Elgammal, Oktay Turetken, Willem-Jan van den Heuvel, and Mike Papazoglou. Formalizing and applying compliance patterns for business process compliance. *Software & Systems Modeling*, 15(1):119–146, 2016.
- [10] Agostino Forestiero and Giuseppe Papuzzo. Agents-based algorithm for a distributed information system in internet of things. *IEEE Internet of Things Journal*, 8(22):16548–16558, 2021.
- [11] David Foster. *Generative deep learning: teaching machines to paint, write, compose, and play*. O’Reilly Media, 2019.

-
- [12] Goyal.al. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.
- [13] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.
- [14] He.al. Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):57–71, 2016.
- [15] Jeremy Jordan. Variational autoencoders. 2018. URL: <https://www.jeremyjordan.me/variational-autoencoders/>(visited on 03/03/2019), 2019.
- [16] George Karypis and Vipin Kumar. Multilevel algorithms for multi-constraint graph partitioning. In *SC'98: Proceedings of the 1998 ACM/IEEE Conference on Supercomputing*, pages 28–28. IEEE, 1998.
- [17] Khajenezhad.al. Gransformer: Transformer-based graph generation. *arXiv preprint arXiv:2203.13655*, 2022.
- [18] Kingma.al. Auto-encoding variational bayes. arxiv e-prints. *arXiv preprint arXiv:1312.6114*, 1312, 2013.
- [19] Koninck.al. act2vec, trace2vec, log2vec, and model2vec: representation learning for business processes. In *International Conference on Business Process Management*, pages 305–321. Springer, 2018.
- [20] Yujia .al Li. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.
- [21] Lin.al. Glimg: Global and local item graphs for top-n recommender systems. *Information Sciences*, 580:1–14, 2021.
- [22] Lucibello.al. Deep learning via message passing algorithms based on belief propagation. *arXiv preprint arXiv:2110.14583*, 2021.
- [23] Luetngen.al. Case2vec: Advances in representation learning for business processes. In *International Conference on Process Mining*, pages 162–174. Springer, 2020.
- [24] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.
- [25] Mani.al. On variational inference for user modeling in attribute-driven collaborative filtering. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3156–3164. IEEE, 2020.
- [26] Heidy M Marin-Castro and Edgar Tello-Leal. Event log preprocessing for process mining: a review. *Applied Sciences*, 11(22):10556, 2021.
- [27] Javier Mata, Ignacio de Miguel, Ramon J Duran, Noemí Merayo, Sandeep Kumar Singh, Admela Jukan, and Mohit Chamania. Artificial intelligence (ai) methods in optical networks: A comprehensive survey. *Optical switching and networking*, 28:43–57, 2018.
- [28] Chris McCormick. Word2vec tutorial-the skip-gram model. *Apr-2016.[Online]. Available: <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model>*, 2016.

-
- [29] Dong Qiu, Haihuan Jiang, and Shuqiao Chen. Fuzzy information retrieval based on continuous bag-of-words model. *Symmetry*, 12(2):225, 2020.
- [30] Nicholas Revett Rolle. *Grammatical tone: Typology and theory*. University of California, Berkeley, 2018.
- [31] Schonfeld.al. A u-net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2020.
- [32] Alexander Seeliger, Stefan Luetzgen, Timo Nolle, and Max Mühlhäuser. Learning of process representations using recurrent neural networks. In *International Conference on Advanced Information Systems Engineering*, pages 109–124. Springer, 2021.
- [33] Tobias Seyffarth and Stephan Kuehnel. Maintaining business process compliance despite changes: a decision support approach based on process adaptations. *Journal of Decision Systems*, 31(3):305–335, 2022.
- [34] Tama.al. An empirical investigation of different classifiers, encoding, and ensemble schemes for next event prediction using business process event logs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(6):1–34, 2020.
- [35] Tenschert.al. Enterprise system capabilities for organizational change in the bpm life cycle. In *International Conference on Business Process Management*, pages 70–82. Springer, 2020.
- [36] Trappey.al. Intelligent compilation of patent summaries using machine learning and natural language processing techniques. *Advanced Engineering Informatics*, 43:101027, 2020.
- [37] Jan vom Brocke, Sarah Zelt, and Theresa Schmiedel. On the role of context in business process management. *International Journal of Information Management*, 36(3):486–495, 2016.
- [38] Xizhao Wang, Yanxia Zhao, and Farhad Pourpanah. Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics*, 11(4):747–750, 2020.
- [39] Adam Wawrzyński and Julian Szymański. Study of statistical text representation methods for performance improvement of a hierarchical attention network. *Applied Sciences*, 11(13):6113, 2021.
- [40] Wu.al. Learning the implicit semantic representation on graph-structured data. In *International Conference on Database Systems for Advanced Applications*, pages 3–19. Springer, 2021.