The Promising Role of Representation Learning for Distributed Computing Continuum Systems

Praveen Kumar Donta, and Schahram Dustdar

Distributed Systems Group, *TU Wien*, Vienna, Austria.
Email: {pdonta,dustdar}@dsg.tuwien.ac.at

***Abstract*—The distributed computing continuum systems (DCCS) and representation learning (ReL) are two diverse computer science technologies with their use cases, applications, and benefits. The DCCS helps increase flexibility with improved performance of hybrid IoT-Edge-Cloud infrastructures. In contrast, representation learning extracts the features (meaningful information) and underlying explanatory factors from the given datasets. With these benefits, using ReL for DCCS to improve its performance by monitoring the devices will increase the utilization efficiency, zero downtime, etc. In this context, this paper discusses the promising role of ReL for DCCS in terms of different aspects, including device condition monitoring, predictions, management of the systems, etc. This paper also provides a list of ReL algorithms and their pitfalls which helps DCCS by considering various constraints. In addition, this paper list different challenges imposed on ReL to analyze DCCS data. It also provides future research directions to make the systems autonomous, performing multiple tasks simultaneously with the help of other AI/ML approaches.**

***Index Terms*—Representation learning; Distributed systems; Compute continuum; Causal inference;**

I. Introduction

The ever-growing complexity of systems, reflected in geospatial distribution, heterogeneous infrastructures, and strategies, calls for developing novel paradigms for their management and orchestration. In this direction, it is clear that we cannot consider the hypothesis of having separate, independent strategies for IoT, edge, and cloud anymore; instead, we need to consider systems in their entirety,i.e., the distributed computing continuum systems (DCCS) and manage them accordingly [1], [2]. A pictorial structure of a distributed compute continuum systems architecture is presented in Fig. 1. From this figure, we can understand that enormous computing devices are participating in DCCS. Therefore, there is necessary to advance novel designs, technologies, and methodologies, to deal with this vast scenario.

To this end, inspecting a novel set of methodologies, called Representation Learning (ReL) techniques, for the DCCS management is particularly interesting. Specifically, this set of approaches aims to extract a high-level, information-rich representation of data that can be used for pattern recognition, behavior prediction, or classification [3]. However, ReL and DCCS are two diverse technologies, the benefits of ReL can be used in DCCS to continuously monitor the systems to achieve zero downtime with prominent performance in terms of rapid deliveries, fault detection, route cause of the faults and etc. Most of these operations are also doing by the traditional machine learning (ML) or Artificial Intelligence (AI) approaches, but they are computationally complex than the ReL [4]–[6].

However, despite the ongoing research on ReL methodologies, there are no approaches that try to apply them to the DCCS [7]. Nevertheless, this is a particularly challenging and promising direction. DCSS generates a wide variety of metrics from heterogeneous sources – from limited IoT devices to high-performative cloud servers. When managing such systems, a demanding process involves understanding the causes and the main actors that drive specific behaviors [8]. Usually, a restricted subset of this information is directly connected to the system's behavior. Knowing this relevant information could open up for fast and rapid actions in terms of management and adjustment of the system state. In this direction, ReL can represent a game-changing tool [9].

In this paper, we discuss the various methodologies flavors. ReL techniques are developed with different structures, like graphs (graph representation learning), matrix factorization, Contrastive representation learning, and Bayesian structures learning, adapting to several use cases. We then analyze their advantages and where they can make a difference. Furthermore, we go a step forward and examine where ReL can be applied in the future, focusing on the ongoing challenges in DCCS, and inspecting what is needed to make ReL techniques work in these scenarios. In these aspect, this paper contributions are summarized as follows:

- We initially discuss different representation learning algorithms, their functions, benefits, and limitations. In addition, we provide an extensive discussion on the benefits of ReL with respect to the distributed computing continuum systems.
- We discuss diversity in DCCS data along with their challenges towards their analytics to extract the knowledge. In continuation, we mention different ways the ReL will treat this data to extract the information to break these challenges.
- Next, the objective of the ReL in DCCS is discussed through simple illustrative examples in this paper. However, there are several learning objectives for the ReL, but consider only a few which help to improve the performance of DCCS.
- Further, we discuss the promising role of ReL for DCCS

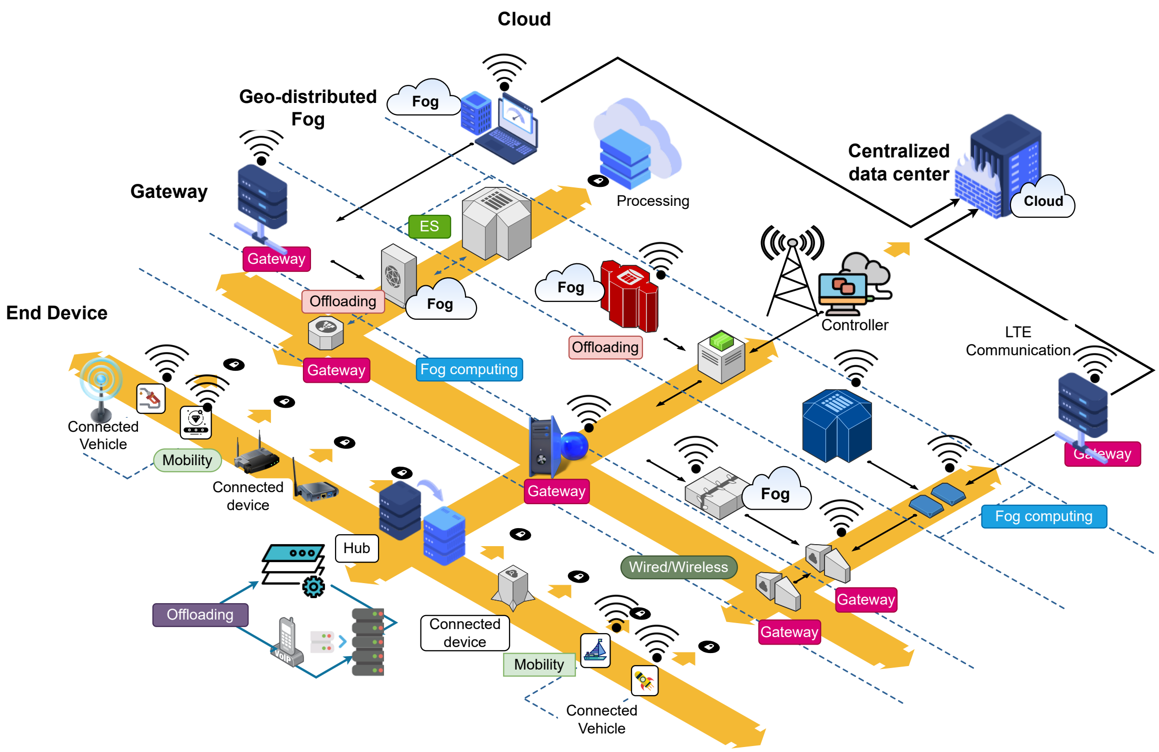


Fig. 1: A visual representation of distributed computing continuum systems architecture

to improve the performance, such as zero downtime, identification of route cause for a problem, and predicting the faults, network, or energy issues in the network.

- Finally, we propose future research direction to address current challenges and make the DCCS autonomous by avoiding human interventions, performing multiple tasks simultaneously, and complexity minimization at the edge, etc.

The rest of the paper is organized as follows: Section II summarizes the representation learning and its benefits in different fields. Section III formulates usability of representation learning in the field of distributed computing continuum systems in terms of addressing different challenges. The research challenges and future directions are discussed in IV. Section V presents the concluding remarks.

II. Representation Learning

The representation learning (ReL) approach builds a predictor or classifier to smoothly extract relevant information by avoiding unnecessary information (including, among the others, noise, outliers, and inconsistency) from the raw data. The well-learned representation helps extract the hidden underlying explanatory features from the data and further improves the performance of machine learning (ML) models. More importantly, a good choice of ReL constructs the ML model faster and efficiently. Several fields use the advantages (subsection II-B) of ReL [3]; worth to mention are signal processing, natural language processing, and object recognition. Several ReL algorithms in the literature allow learning the efficient representation from the given data [10], [11]. Next, in this section, we show and discuss the most promising ones concerning improving the performance of the DCCS.

A. *Representation learning Algorithms*

By studying the various ReL algorithms proposed in the literature, we can extract their main benefits and limitations and explore to which extent they are helpful to get improving the performance of the DCCS. ReL usually encompasses a set of rules and approaches for learning a representation from the given input data. These learning algorithms help extract the latent features, removing conflicting information, compression, classification, and other noise. We extend the first overview of the most promising ReL algorithms for DCCS discussed in [1] by summarizing hereinafter other relevant approaches:

1) Matrix factorization: It helps to reduce the given data while removing the constituent parts to estimate the complex relations among the attributes rapidly and efficiently. It helps to classify the contested relations, irrelevant features, and relations. The low computational complexity makes it easy to deploy and work with constrained devices [12]. However, there are a few limitations, such as failure to handle the noise data or outliers, a complex in working with Heterogeneous data, and not working for streaming data.

2) Random walk learning: The random walk ReL efficiently identifies the structural properties within the data and quickly removes redundancy. It is more promising in extracting the causal relations among the attributes and also predicts their strengths [13], [14]. Unlike matrix factorization, they can efficiently handle the missing, noise, or outliers in linear complexity. However, they need an extensive data set to result from accurate results. Since the DCCS is vast data of multiple systems, it can efficiently process and get thriving representation.

3) Graph representation learning: We can consider the whole DCCS as a semantic network structure (or Graph) where the vertices are associated with a rich set of attributes, and relations with their neighborhood nodes are considered edges. The graph ReL is used to identify or predict the faults, irrelevant nodes, or links from the complex data using local neighborhood information [15], [16]. The graph neural networks (GNN) help to learn from the graph representation learning to predict, for example, the strength of relations, classify the nodes or connections, cluster similar nodes [17]. However, this approach is computationally expensive, and customizing the model by choosing various factors such as aggregation function, the number of layers,and style of message passing is complex.

4) Bayesian network structure learning: The Bayesian network structure learning (BNSL) is used to learn a structure from the given data, which infers a distribution over a dependency graph that looks like a direct acyclic graph (DAG). Choosing the best among the many possible DAGs for a given data is burdensome, but the Bayesian network uses different approaches to select the best DAG over the many candidates [18]–[20]. The BNSL can be used in constrained and score-based depending on the type and available data, whereas they have their benefits and challenges. We can use BNSL to run streaming data. This approach uses low computational resources, handles missing values or outliers, and results in high accuracy.

5) Contrastive representation learning: It learns the representation in a discriminative manner by differentiate the input data feature instead of learning from the discrete data sample. It can work on both labeled and unlabelled data. When working with the unsupervised data, the contrastive ReL (CRL) can predict the labels and works like a self-supervised learning approach [21], [22]. This approach is very efficient in classifications, clustering, and prediction operations. The CRL results in high accuracy when the input data is more.

B. Benefits of Representation learning

There are several benefits of the ReL in different fields, whereas the following are the significant advantages of DCCS:

1) Observe the underlying explanatory factors: There are several underlying factors in DCCS data, where the ReL can easily extract them from the data of the raw system. These factors may be, for example, Causal relations, dependencies, or usage of the resources. There are multiple(or different) categories of explanatory factors of DCCS; still, the ReL can extract them independently.

2) Interpretability gets improved: Interpretability is the possibility of consistent predictions and decisions from the model. Because of the learned representation from the given data, it is simplified and improved Interpretability. The feature of observing underlying factors from the information also may help to get Interpretability improved.

3) Easy to extract helpful information: Once the representation of DCCS is learned, it is easy to answer the question related to the systems because it summarizes the meaningful data at a glance. The Causal relations and proximity information further simplify extracting meaningful information from the learned representations of DCCS.

4) Improved performance models: Due to the feature engineering done by the ReL, the machine learning models work better in providing the results accurately and rapidly.

5) Rapid processing: The ReL helps remove the inconsistent and redundant information and extract meaningful features from the raw data. These features can be inputs to the machine learning models. So, the model does not need to perform the data cleaning or prepossessing operations and run using this limited feature to increase the processing speed.

III. Representation Learning for Distributed Computing Continuum Systems

This section discusses the challenges of the DCCS' data, ReL outcomes, and possible operations for DCCS when using ReL. The general flow of ReL, AI/ML and their operations are summarized using Fig. 2.

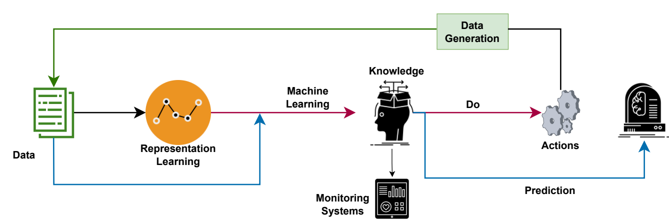


Fig. 2: The general learning paradigm using representation learning and possible outage

A. Challenges in distributed computing continuum systems' data

The challenges discussed in this section are primarily focused on the data about the systems/devices used in DCCS. There are several data formats, including the metadata of the participating devices of DCCS, log information, network I/O or energy information, etc. Several challenges are associated with these data [3], [11], [23] to analyze using traditional learning algorithms, whereas we summarize the essential and related to DCCS as follows.

1) High non-linearity: The non-linearity of DCCS data talks about the dependent and independent relations between the attributes. Due to the heterogeneity in DCCS, there is a lack of direct relationships among the features, which makes the learning strategy more complex to extract a good representation. In such cases, the regularized auto encoders can learn representation from these non-linear data using local variations.

2) Structure-preserving: As we know, multiple devices such as edge, IoT, data centers, and cloud are connected in DCCS. It is necessary to represent these diversities between the devices in the learned vectors in the local neighborhood or global structure. The overarching aim of structure-preserving is to conserve explicit properties of the continuous model in its discretization. So, ReL should simultaneously preserve the DCCS structures.

3) Sparsity: Sparsity can be defined in many ways. We simplify it as the missing values due to various factors, or the available data is insufficient to process or extract the latent features. Technically, a feature x can be reflected in the datasets only a few times, resulting in no latent features (zero latent features). In contrast, sometimes x is linearly dependent, resulting in a single latent feature that does not help for an efficient representation.

B. Learning objectives

This section discusses the objectives of the ReL, which help enhance the performance of the DCCS. In this context, we identify four essential objectives: classification, multi-clustering, causal inference, and knowledge extraction, and summarize each learning objective as follows:

1) Classification: Classification refers to the process of categorizing discrete class labels. It can be performed on both supervised and unsupervised data. The ReL is good in conducting the classifications in heterogeneous networks with the help of neural networks, where the different node properties [24]. In this context, the data about the DCCS is heterogeneous and has several challenges, as discussed in subsection III-A. So, the ReL can help classify the DCCS data by identifying similar categories (based on computing capabilities, other resources), properties or functionalities, etc.

2) Multi-clustering: The multi-clustering is considered as the same clustering strategy that can be applied to several parts of the systems or the same input on different clustering approaches take place parallel [25]. The ReL which uses multi-clustering is treated as a sparse or distributed representation. This architecture helps to increase the scalability, and availability of each individual cluster. It is also useful to perform the operations on each individual cluster selectively and separately. An illustrative example of multi-clustering is explained in Fig. 3. In Fig. 3(a), the data about the end devices are placed. Fig. 3(b) and Fig. 3(c), represent the multiple approaches to form clusters from the same data. For example, Fig. 3(b) create the cluster by considering based on the devices category vehicles, medical devices and home appliances from the data set. Similarly, Fig. 3(c) consider the applications such as healthcare, agriculture, and smartcity.

3) Causal inference: The Causal inference considered different study designs, assumptions, or estimation strategies to extract the Causal conclusions or interpretations from the data. How does the cause of a variable affect other variables or features? Learning causal inferences is easy to extract relation information from large individual systems. The ReL can extract the causal inference of the system using the low-level observations, which further help predictions (consequences of different actions) and answer the questionnaires [11]. It can rival explanations for the observed associations between the attributes. In the DCCS, the Causal ReL helps solve issues such as monitoring, load-balancing, and Causal predictions. The nodes in causal network are the variables or attributed whereas the link (arrows) indicates the causal connections between the variables. Fig. 4 explains an illustrative example of Causal inference. From Fig. 4, we can identify the inference between the four label buffer occupancy (B), delay (D), packet loss (L), and throughput (T). The relations are highlighted using arrows. The relation between the B and D is a causal relation, and probability values indicate the cause of complete buffer occupied leads to an increase in the delay. Similarly, packet loss depends on the buffer, and packet loss causes throughput efficiency.

4) Knowledge extraction: Selecting the best ReL algorithm always extracts the knowledge from the raw data. Regularization is a standard method to perform the knowledge extraction through ReL with new data outside the training data [26]. The quality of the knowledge of learned data is decided based on the ability to generalize the knowledge [27]. Knowledge extraction simplifies the decision-making systems and responds quickly.

C. Promising role of ReL for DCCS

While considering all the complications in the data with the learning objectives discussed in the previous subsections, the ReL play a promising role to enhance the performance of the distributed compute continuum systems while analysing their data. The list of benefits is long, so we try to group them into different categories and explained them in detail as follows.

1) Monitoring: In DCCS, the systems or devices are connected to the Internet or our private network and operate using electric power or batteries. Monitoring systems here indicate observing how the device is responding or working? This means the device is a healthy power supply, is strongly connected to the Internet or private network, functions of the resources, responds to the commands, etc. These devices monitor manually is a tedious task and also not efficient. Since several monitoring tools in the market are poorly monitoring the IoT device health, [28], which visualizes the device's condition and alerts the users. However, these tools are made only to monitor the end devices with limited features, whereas the DCCS involves multiple layers of devices that are not monitored by these tools.

The ReL simplifies this task by considering the details of the devices, including device metadata, CPU, energy, network I/O, and log file information, to learn the representations while extracting the knowledge of the system. These learning strategies and knowledge extraction successfully find the root cause of a problem, such as manufacturing defects, issues with network connections, power connections, etc. This monitoring phase checks that all devices are meeting the reliability goal with Service Level Objectives (SLOs) [29]–[31], before notifying the administrator or user. The network or power connection

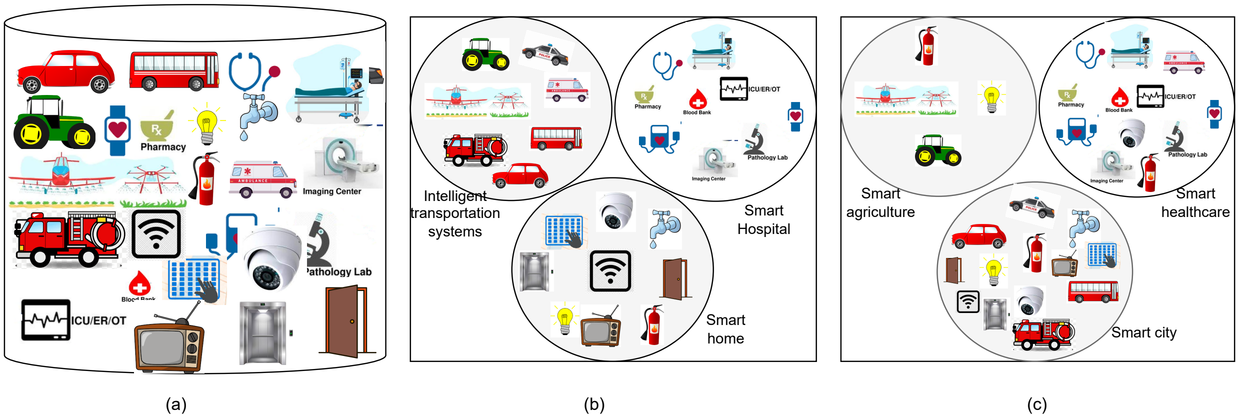
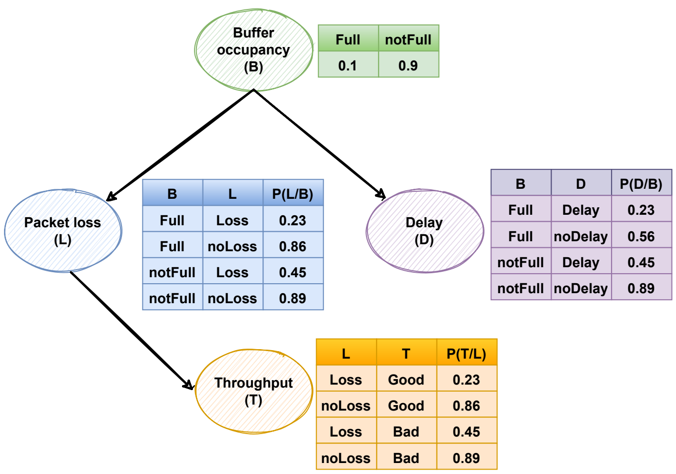


Fig. 3: Illustrative example for Multi-clustering (a) End devices data (b) Cluster 1 (c) Cluster 2



Full	notFull
0.1	0.9

B	L	P(L/B)
Full	Loss	0.23
Full	noLoss	0.86
notFull	Loss	0.45
notFull	noLoss	0.89

B	D	P(D/B)
Full	Delay	0.23
Full	noDelay	0.56
notFull	Delay	0.45
notFull	noDelay	0.89

L	T	P(T/L)
Loss	Good	0.23
noLoss	Good	0.86
Loss	Bad	0.45
noLoss	Bad	0.89

Fig. 4: An illustrative example for causal inference

issue can be recognized when any device's (complete) data values are missing. But, deciding the root cause correctly or the exact problem, the analysis or previous information on missing details is analyzed.

The ReL help in monitoring the performance of each device DCCS to improve the overall performance. It also helps to debug devices and prevent errors. Monitoring systems through ReL will not affect the time-critical application, which is a challenging issue. Still, the ReL is faster than the traditional AI/ML algorithms. In addition, the monitoring systems need intelligent protocols for data and control signal transmissions between the devices. During performance monitoring, privacy, safety, and security issues are a high priority with the productive utilization of the system's data. Representation learning plays a promising role in distributed compute continuum systems, achieving zero downtime through monitoring.

2) Predictions: The ReL helps to predict the future issues of the DCCS. These failures include failures of devices, software faults, threats, network issues, resource availability, etc. Once these faults are identified, it is easy to perform the predictive maintenance of the DCCS devices. For example, the energy issues of battery-operated devices can be predicted by the amount of time it can be sustained using the previously collected energy-related information and the number of operations the systems do. Based on these predictions, the systems can alert the administrator to take further actions or rectify the issues autonomously to avoid downtime.

The predictions of the ReL estimate the availability of the devices for computation for the future generation data in systems. It helps to balance the computational loads in the systems to avoid latency in the system. Estimate the system's buffer occupancy to control the packet loss during the data transmissions within the systems. Further, ReL can predict the resource availability to decide where the computations are to be done in the computing continuum hierarchy. So, the data and computational offloading performances are increased. The ReL can provide an efficient solutions than the existing solutions [32], [33], in terms of transmission scheduling and computational offloading through predictions. Overall, these predictions help reduce the mean time to resolution through ReL in distributed compute continuum systems.

IV. Future Research Directions

The previous discussions show the ReL potential in analyzing the system's data and providing relevant information about the systems. Based on this information, we can have an aggregated and rich view of the system, enabling advanced and autonomous management strategies. In the following, we evaluate use cases and situations for DCCS where ReL could potentially have an impact in the future, discussing what they can add and what is still missing.

A. Streaming analytics

In DCCS, many new devices are added daily. Due to this, the representations of DCCS change timely. Whereas the existing ReL algorithms are not intended for dynamic infrastructures, it is possible to adapt them to this scenario. To this end, it is necessary to embed the dynamic policies in ReL to analyze newly added devices' data on-the-fly.

B. Zero-touch provisioning

Zero-touch provisioning (ZTP) is a virtual software for management and services to the end-to-end network with no human intervention. With an increasing number of resources being managed, delivering and managing dynamic user service requests becomes ever more complex [34] in DCCS. To overcome this complexity, ETSI offers the idea that ZTP is a new breed of network management functionality, seeking to integrate network functionality, cutting-edge communication technologies (eMBB, URLLC, and mMTC), and automatically carrying out edge computing processes. In the future, the ZTP services will be extended in computing and communications aspects along with the network and service management to justify the title. As we discussed in subsection III-C, the ReL can identify the faults by monitoring the systems, predicting the failures, and also balancing the load. Here, the ZTP can work to address any problem, such as fault tolerance if any device or system is in trouble, without waiting for human instructions. Similarly, ZTP can rectify the errors if ReL predicts any in the system. In summary, taking the advantages of ZTP and ReL make the system more autonomous and minimizes the system's latency or faults without waiting for human instructions.

C. Inference at the edge

The edge and end devices are resource-constrained, so it is necessary to minimize the computation, which does not affect the performance, including accuracy, cost, and delay. There are several ways in the literature to reduce the computational complexity at the edge devices while using the machine learning or Artificial intelligence approaches [35]. The primary way is to remove the inconsistent or redundant details from the input data. Another side, there are few model compression techniques to lower the computations. *Lippmann* [36] introduced to choose the number of layers in neural networks to process the information with minimal computations and no loss of accuracy. *Lippmann* also focused on parallel computations to converge quickly. Similarly, *Volodymyr et al.* [37] present the model compression techniques such as processing the alternate frames to minimize the computational load on the systems without altering the performance. As discussed in Section III, the ReL is very efficient in identifying the best feature out of the given data while managing the missing information, eliminating redundant or inconsistent information without affecting the performance. So, these advantages must help improve the inference performance in the edge.

D. Multi-task learning

As we know in DCCS, multiple devices are interconnected and acquire data from various sources such as industries, transportation, cities, buildings, and homes. The traditional ML algorithms solve a particular task and optimize a single objective function simultaneously. These approaches do not fit DCCS because of their data diversity and formats. So, there is a need for multi-task learning (MTL) in DCCS, which tries to learn different tasks simultaneously to optimize the various objective functions simultaneously. In the literature, the MTL is used to forecast time-dependent cloud workloads [38], inference attacks [39], etc. Besides, the ReL algorithms are useful for the Causal inference among these independent data sources to acquire knowledge from these data. They efficiently find the proximity within the data and analyze the relationship strength. It also efficiently classifies the data by reducing the burden on MTL.

E. Interoperability

As we discussed in subsection III-A, given the heterogeneity in devices, their execution platforms, interfaces, or communication patterns in the DCCS, interoperability is one of the primary challenges to be addressed in the future [40], [41]. Interoperability is still a critical component of AI or ML approaches. However, it is necessary to establish autonomous interoperability in DCCS using their semantic metadata and latent features by applying ReL. The ReL, such as node Embedding, with the help of transfer learning approaches, helps compress the data, reduce the noise, and provide meaningful latent features [42]. In addition, the ReL is helpful for the model and optimization explorations using the metadata and utility functions, respectively, to address the interoperability [43]. So, the ReL plays a promising role in DCCS while addressing the Interoperability issues using the system's metadata.

V. Conclusions

This paper talks about two emerging and diverse technologies: representation learning and distributed computing continuum systems. The primary focus of this paper is to extend the discussion on the promising role of representation learning for distributed computing continuum systems to improve performance through monitoring, predictions, etc., to achieve zero downtime for the system with improved service provisioning. In this context, we describe different representation learning algorithms and their pitfalls. The challenges concerning data formats of the systems and their log files are studied, and possible solutions for analysis with the help of ReL. The promising role of ReL in monitoring and making predictions in the system are explained in this paper. We analyze the learning outcomes of representation learning concerning distributed compute continuum systems are studied. Finally, we provide future research direction and use of ZTP, MTL, etc., to make the computing continuum systems autonomous. Future research also needs to handle the interoperability issues in the systems with the help of representation learning.

Acknowledgement

We thank Mr. Abhishek Hazra (Indian Institute of Technology, Dhanbad, India) for helping in drawing figures. We thank Dr. Andrea Morichetta and Dr. Ilir Murturi (Distributed Systems Group, TU Wien, Austria) for helping with proofreading and language corrections.

REFERENCES

[1] S. Dustdar, V. C. Pujol, and P. K. Donta, "On distributed computing continuum systems," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[4] A. Y. Ding, E. Peltonen, T. Meuser, A. Aral, C. Becker, S. Dustdar, T. Hiessl, D. Kranzlmüller, M. Liyanage, S. Maghsudi *et al.*, "Roadmap for edge ai: a dagstuhl perspective," pp. 28–33, 2022.

[5] S. S. Gill, M. Xu, C. Ottaviani, P. Patros, R. Bahsoon, A. Shaghaghi, M. Golec, V. Stankovski, H. Wu, A. Abraham *et al.*, "Ai for next generation computing: Emerging trends and future directions," *Internet of Things*, vol. 19, p. 100514, 2022.

[6] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020.

[7] V. C. Pujol, P. Raith, and S. Dustdar, "Towards a new paradigm for managing computing continuum applications," in *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 2021, pp. 180–188.

[8] A. Hazra and T. Amgoth, "Ceco: Cost-efficient computation offloading of iot applications in green industrial fog networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6255–6263, 2022.

[9] A. Morichetta, V. C. Pujol, and S. Dustdar, "A roadmap on learning and reasoning for distributed computing continuum ecosystems," in *2021 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2021, pp. 25–31.

[10] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[11] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[13] M. Xue, W. Cai, J. Su, L. Song, Y. Ge, Y. Liu, and B. Wang, "Neural collective entity linking based on recurrent random walk network learning," *arXiv preprint arXiv:1906.09320*, 2019.

[14] J. Li, L. Wu, R. Hong, and J. Hou, "Random walk based distributed representation learning and prediction on social networking services," *Information Sciences*, vol. 549, pp. 328–346, 2021.

[15] C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, and W. Wang, "Robust graph representation learning via neural sparsification," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 11 458–11 468.

[16] W. L. Hamilton, "Graph representation learning," *Synthesis Lectures on Artifical Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, 2020.

[17] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Systems with Applications*, p. 117921, 2022.

[18] W. Zhu, N. L. C. Nguyen, and S. Cripps, "Structure learning for hybrid bayesian networks," *arXiv preprint arXiv:2206.01356*, 2022.

[19] M. Scutari, C. Vitolo, and A. Tucker, "Learning bayesian networks from big data with greedy search: computational complexity and efficient implementation," *Statistics and Computing*, vol. 29, no. 5, pp. 1095–1108, 2019.

[20] M. Scanagatta, A. Salmerón, and F. Stella, "A survey on bayesian network structure learning from data," *Progress in Artificial Intelligence*, vol. 8, no. 4, pp. 425–439, 2019.

[21] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *IEEE Access*, vol. 8, pp. 193 907–193 934, 2020.

[22] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9929–9939.

[23] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," *IEEE Transactions on Big Data*, vol. 6, no. 1, pp. 3–28, 2018.

[24] L. D. Santos, B. Piwowarski, L. Denoyer, and P. Gallinari, "Representation learning for classification in heterogeneous graphs with application to social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 5, pp. 1–33, 2018.

[25] J. Hu and J. Pei, "Subspace multi-clustering: a review," *Knowledge and information systems*, vol. 56, no. 2, pp. 257–284, 2018.

[26] J. Rodriguez, "Knowledge is everything: Using representation learning to optimize feature extraction and knowledge quality," https://www.linkedin.com/pulse/knowledge-everything-using-representation-learning-jesus-rodriguez/, accessed: August 17, 2022.

[27] B. P. Majumder, N. Potti, S. Tata, J. B. Wendt, Q. Zhao, and M. Najork, "Representation learning for information extraction from form-like documents," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 6495–6504. [Online]. Available: https://aclanthology.org/2020.acl-main.580

[28] S. Cooper, "Best IoT device monitoring tools," https://www.comparitech.com/net-admin/iot-device-monitoring/, accessed: August 17, 2022.

[29] S. Nastic, A. Morichetta, T. Pusztai, S. Dustdar, X. Ding, D. Vij, and Y. Xiong, "SLOC: Service level objectives for next generation cloud computing," *IEEE Internet Computing*, vol. 24, no. 3, pp. 39–50, 2020.

[30] T. Pusztai, A. Morichetta, V. C. Pujol, S. Dustdar, S. Nastic, X. Ding, D. Vij, and Y. Xiong, "SLO Script: A novel language for implementing complex cloud-native elasticity-driven slos," in *2021 IEEE International Conference on Web Services (ICWS)*. IEEE, 2021, pp. 21–31.

[31] S. Nastic, T. Pusztai, A. Morichetta, V. C. Pujol, S. Dustdar, D. Vii, and Y. Xiong, "Polaris scheduler: Edge sensitive and slo aware workload scheduling in cloud-edge-iot clusters," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*. IEEE, 2021, pp. 206–216.

[32] A. Hazra, P. K. Donta, T. Amgoth, and S. Dustdar, "Cooperative transmission scheduling and computation offloading with collaboration of fog and cloud for industrial iot applications," *IEEE Internet of Things Journal*, 2022.

[33] D. Lan, A. Taherkordi, F. Eliassen, L. Liu, S. Delbruel, S. Dustdar, and Y. Yang, "Task partitioning and orchestration on heterogeneous edge platforms: The case of vision applications," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7418–7432, 2022.

[34] I. Murturi and S. Dustdar, "Decent: A decentralized configurator for controlling elasticity in dynamic edge networks," *ACM Transactions on Internet Technology (TOIT)*, 2022.

[35] J. Zhang, Z. Qu, C. Chen, H. Wang, Y. Zhan, B. Ye, and S. Guo, "Edge learning: The enabling technology for distributed big data analytics in the edge," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–36, 2021.

[36] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, 1987.

[37] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[38] J. Bi, H. Yuan, M. Zhou, and Q. Liu, "Time-dependent cloud workload forecasting via multi-task learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2401–2406, 2019.

[39] X. Ma, J. Ma, S. Kumari, F. Wei, M. Shojafar, and M. Alazab, "Privacy-preserving distributed multi-task learning against inference attack in cloud computing," *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 2, pp. 1–24, 2021.

[40] P. Wegner, "Interoperability," *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 285–287, 1996.

[41] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "A comprehensive survey on interoperability for iiot: taxonomy, standards, and future directions," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–35, 2021.

[42] J. Nilsson, "System of systems interoperability machine learning model," Ph.D. dissertation, Luleå University of Technology, 2019.

[43] J. Nilsson, F. Sandin, and J. Delsing, "Interoperability and machine-to-machine translation model with mappings to machine learning tasks," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1. IEEE, 2019, pp. 284–289.