Deep Meta-Learning and Dynamic Runtime Exploitation of Knowledge Sources for Traffic Control

Sandra Ottl
University of Passau
Passau, Germany
ottl@uni-passau.de

***Abstract*—In the field of machine learning and artificial intelligence, meta-learning describes how previous learning experiences can be used to increase the performance on a new task. For this purpose, it can be investigated how prior (similar) tasks have been approached and improved, and knowledge can be obtained about achieving the same goal for the new task. This paper outlines the basic meta-learning process which consists of learning meta-models from meta-data of tasks, algorithms and how these algorithms perform on the respective tasks. Further, a focus is set on how this approach can be applied and is already used in the context of deep learning. Here, meta-learning is concerned with the respective machine learning models themselves, for example how their parameters are initialised or adapted during training. Also, meta-learning is assessed from the viewpoint of Organic Computing (OC) where finding effective learning techniques that are able to handle sparse and unseen data is of importance. An alternative perspective on meta-learning coming from this domain that focuses on how an OC system can improve its behaviour with the help of external knowledge sources, is highlighted. To bridge the gap between those two perspectives, a model is proposed that integrates a deep, meta-learned traffic flow predictor into an organic traffic control (OTC) system that dynamically exploits knowledge sources during runtime.**

***Index Terms*—meta-learning, organic computing, autonomous learning, deep learning**

I. Introduction

In the field of artificial intelligence, it is a natural move to look at the way humans deal with issues of learning in order to find ways of improving machine learning approaches. For instance, considering the act of learning a new language, one can observe that we improve the process of learning how to learn a new language with each language. Hence, when one has already learned, for example, french and italian, another roman language such as spanish, can be learned a lot faster. This is the case as the vocabulary and the grammar are similar and our brain is pre-trained to deal with new tasks that are similar to previously learned ones. Further, when regarding musical instruments, if a person can read and play by notes, it is not only easy for them to play a certain piece of music but it is also possible for them to play any other piece. Further, when they can play the violin, they will learn the cello with less effort than they would have without their prior knowledge. On the other hand, when a person exclusively learned to play the piano, learning to play the drums will need more exercise. Accordingly, machine learning algorithms would improve and get faster if they could apply this human approach in order to achieve learning to learn a new task out of knowledge of previous, similar tasks, generally known as meta-learning [1].

This is also of significant importance to the field of deep learning, where the human neuron system was a role model to create multi-layered neural network architectures [2]. To achieve the best results with deep learning, traditionally a lot of data is needed as input as the underlying architectures are quite complex.

In contrast to this, in the field of OC not as much data is available because its basic concept is to let the organic system learn, adapt, organise and configure itself autonomously and dynamically to specific conditions [3]. Again, nature is employed as inspiration, as, for instance, the behaviour of ants can be seen as such an organic system. When looking at the way worker ants search for and transport food, it can be observed that they naturally find the best route for doing so. This does not happen in a manner humans may fully perceive but it just works itself out in the best possible way and when obstructions hit the path, the system adapts itself automatically without coming to a stop. With real-life examples like this in mind, it is desired to create a system where its autonomous sub-systems are interacting with each other and overall system decisions are based on local knowledge. By being able to work on problems while the system is running, it is possible to overcome decisions at design-time and shift them to runtime [4].

Learning based on prior knowledge, or meta-learning, in connection with deep learning methods could be a good asset in work relating to OC as data for these systems is often not available before runtime. Hence, to address this issue, concepts and state-of-the-art research of meta-learning and deep learning are reviewed and linked to a specific OC environment — an OTC system.

This paper presents both the traditional notion of meta-learning as well as an alternative perspective from the field of OC that deals with exploiting external knowledge sources and bridges the gap between the two by investigating how they could be combined in an OTC system.

The paper is laid out as follows: First, a general overview of the field of meta-learning is given in Section II. Afterwards, two current research perspectives in the field are reviewed in Section III and Section IV. The first perspective describes how meta-learning is related to the popular research field of deep learning while the second provides an alternative model for use in OC systems that focuses on ways in which different knowledge sources can help the system's self-adaptation and self-organising properties. Addressing this paper's research question, an idea of how both of these perspectives could be integrated into an OTC system is given in Section V. Finally, the paper is concluded in Section VI, outlining open issues and possible future research.

II. An Overview of Meta-Learning

In conventional machine learning, data from a single task is used to train and evaluate a learner. E.g., in supervised learning a model is trained on data instances that are gathered in a training set and afterwards this learner is tested on instances of a test set, cf. Figure 1. In contrast to this, in meta-learning we consider not instances but entire tasks [1]. So when we talk about *Meta-Learning*, or Learning to Learn, we mean that prior tasks are observed and knowledge obtained from this is used to learn a similar task. Each of those tasks consists of a training and test set including some instances. When training the meta-model, each of those tasks is considered and the emerging meta-model is then used to form a learner that is handling the new task, which is depicted in the lower part of Figure 1.

A. Meta-Data

The first step to meta-learning is to exactly look at possible information that can be obtained from previous tasks. Those *meta-data* include knowledge received directly from the tasks, but also entail the configurations of the algorithm, its parameters and the evaluations of the applied model containing all measurable contents [1].

The characterisations of a specific task are defined as a set of *meta-features* [1]. Some of the most commonly used, simple meta-features are the numbers of instances, classes and outliers. The number of instances can indicate how fast and scalable a model is, the number of classes may reveal how complex the set is and the number of outliers suggests the noisiness of the data [1], [5]. In a similar way, the configurations of different machine learning algorithms are of importance in the meta-learning process. Here, parameters range from general types of methods, e. g. Support Vector Machines or neural networks, to their specific, exact parameterisations, e. g. the type of kernel and values of associated parameters for Support Vector Machines or the weights stored in a neural network. Lastly, for each combination of considered task and configuration of machine learning technique, observations about the performance — measured by metrics like classification accuracy or regression errors in supervised settings, or received rewards in reinforcement learning — and the training process, e. g. training time or learning curves, are taken into account.

From these collected meta-data, a meta-model is then tasked with producing an appropriate model for a new, unseen task.

B. Transfer Learning

Transfer learning examines a domain containing a lot of data and transfers the procured insight of this field into a domain with less data [6]. Hence, a model trained on tasks of the former domain can be used as initialisation for constructing a model from the tasks of the latter field. The effectiveness of this process is depending on how profoundly the contents of both domains are related, i.e. in our case how similar one task is to the other. This can be achieved by choosing the new model to be as similar to the old models as possible, in terms of content, context, structure or in a other way. Transfer learning is very well applicable on neural networks, as such a model can easily be used as a pre-trained initialisation model that can be fine-tuned employing data from the new task [1]. Thus, fine-tuning is often used as transfer learning for deep learning [7], but to avert over- or under-fitting manual choices regarding learning rate and which layers to train have to be made, like Zhou et al. have suggested in their work [8]. In contrast to this, a meta-learner, such as those described in Section II-C, automates this process for multiple tasks at once. Moreover, the meta-learner should be proficient enough to rapidly learn from old tasks and automatically adapt to new, similar ones [1].

C. Meta-Learners for Neural Networks

In the context of neural networks, meta-learning approaches can be found that learn how to influence the initialisation and learning behaviour of base learners over a range of tasks, in order to quickly generalise to a new task. Two recent approaches are outlined below.

1) Model-Agnostic Meta-Learning (MAML): This algorithm for meta-learning introduced by Finn et al. [9] can be applied to any machine learning model that is trained with gradient descent and tackles the problem from a different perspective. The proposed algorithm effectively trains a machine learning model on a distribution of training tasks to be easy to fine-tune to new, unseen tasks. Specifically, this means that the algorithm learns to adjust the initial parameters of the underlying machine learning model in a way such that only a small number of gradient steps and training data are needed to achieve good performance on a task.

2) Meta-SGD: The Meta-SGD algorithm as introduced by Zhou et al. [10] can be seen as a continuation of the philosophy behind MAML. Compared to the former, Meta-SGD not only learns the model's initialisation parameters but also the update direction and learning rates. As is the case with MAML, Meta-SGD can be applied to both supervised learning tasks as well as reinforcement learning.

D. Few-Shot Learning

Few-shot learning describes the problem of training a machine learning model for a task that does not contain many examples [1]. Humans have an innate ability to do this — consider for example the following scenario of learning about unfamiliar, foreign, unknown types of fruit: Assuming a person does not know what a mangosteen, for instance, looks like, a

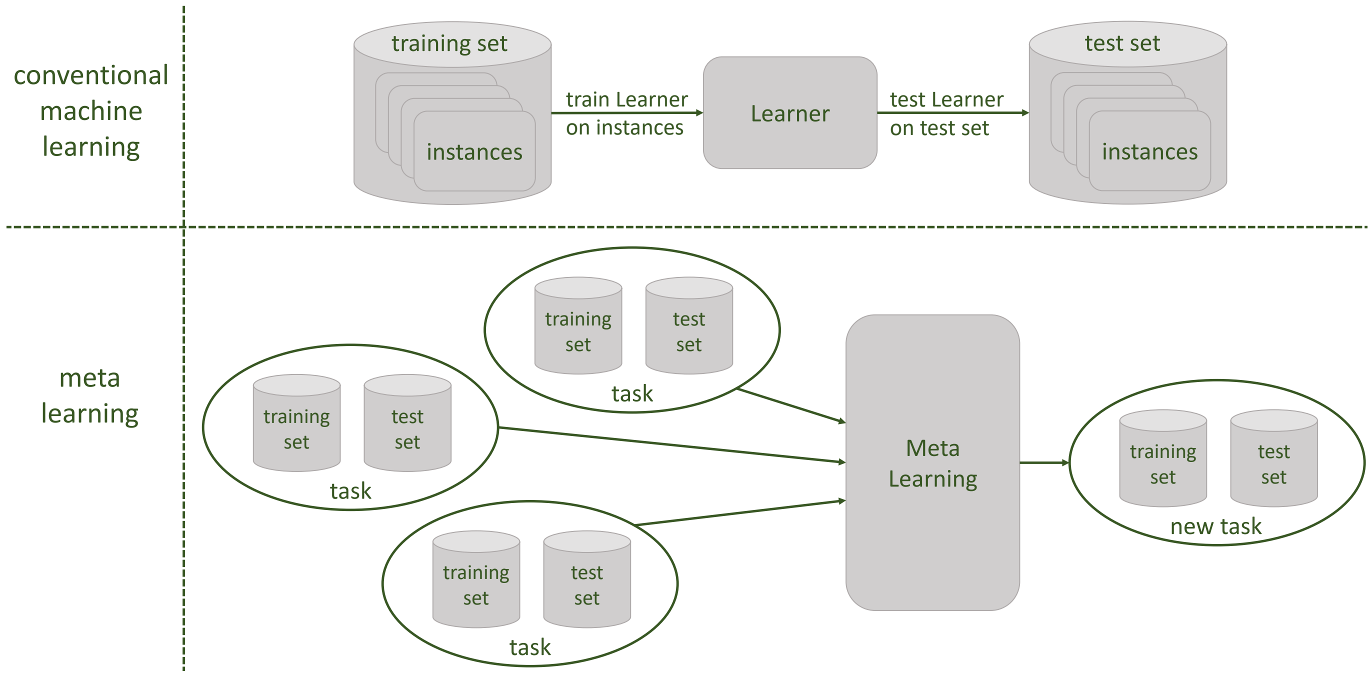

Fig. 1. In conventional machine learning (here supervised learning), instances of a so-called training set are used to train a Learner. This Learner is then tested on instances of a test set. In meta-learning, we inspect different, similar tasks, each composed of a training set containing (few) instances and a test set with other instances. Based on knowledge it attains from these tasks, the meta-learning process then aims to produce a machine learning method that performs well on a new, unseen task.

handful of pictures of this fruit will be shown to this person and afterwards they have to choose the mangosteen from a variety of images of different fruits. With this information, a person will easily be able to decide which fruit is the mangosteen. The underlying problem of few-shot learning is to recreate this instinctive learning process of humans by machine learning operations. The described task of few-shot image recognition is challenging — even for state-of-the-art machine learning models that achieve super-human performance on large-scale image recognition tasks as their deep architectures require large amounts of training data, cf. [2] and Section III. In this context, meta-learning tries to learn how to effectively train deep learning models on small amounts of training data, e. g. by learning effective representations over a set of training tasks and changing parameters of the underlying model accordingly to decrease training effort on a new task [1].

III. Deep Learning in Meta-Learning and Traffic Control

In the following, the field of deep learning is described in the context of meta-learning. Also, two deep learning approaches relevant to traffic control are described.

A. Deep Learning

Deep learning describes a popular sub-area of machine learning that uses large amounts of data to train various neural network architectures, often on raw sensory input data, like images, audio or video content. In this context, "deep" describes neural networks with a large number of layers [2]. Deep learning provides state-of-the-art results in fields like visual recognition [11]–[13] or speech and sound processing [14]–[16], to name a few. An important property of deep learning is its ability to derive useful abstracted representations of the raw input data during training. In the context of meta-learning, representations learned on a specific task can often be transferred to learning new, related tasks. For example, the feature representations learned by convolutional neural networks (CNNs) on the large-scale visual recognition corpus ImageNet [17] can serve as effective input features for other visual recognition tasks, often outperforming handcrafted features [18].

B. Deep Meta-Learning

In their work, Zhou et al. [8] integrate the representational power of deep learning with meta-learning. Normally, meta-learning is performed on the input space, e. g. for image recognition tasks this can mean the raw pixel values of pictures, but, as described above, deep neural networks are able to learn higher-level, conceptual representations in a data-driven way. The authors describe a framework for the task of few-shot image recognition that exploits this property: Instead of training a meta-learner on the raw image input data, they instead employ a concept generator, e. g. a CNN, that is trained jointly with the meta-learner on a large external image recognition database to learn useful representations of the input data. This transfers the meta-learning problem from the complex, high-dimensional input space to a lower-dimensional, easier to grasp concept space. In their framework, both the concept generator and the meta-learning component are replaceable by arbitrary deep representation learners and meta-learners, respectively. They specifically also name the meta-learners described in Section II-C. Finally, they point out, that their framework could be applied in a lifelong learning scenario, as the concept generator will continue to evolve given new data and tasks.

C. Traffic Flow Prediction

The problem of *traffic flow prediction* can be effectively addressed with deep learning methods. Lv et al. [19] describe a deep stacked autoencoder architecture that learns useful, generic representations for predicting traffic flow while inherently considering the spatio-temporal aspects of the problem. They train their autoencoder in a greedy, layer-wise fashion to make short-term traffic flow predictions for different freeways from a specific number of past traffic flow observations. They prove the effectiveness of the learned feature representation by showing that their approach performs better than traditional methods without deep representational learning on the Caltrans Performance Measurement System database.

For the similar task of network transportation speed prediction, Ma et al. [20] use an image based representation of spatio-temporal traffic dynamics that is then used to train a CNN to predict network-wide traffic speeds with high accuracy. A transportation network's traffic speeds are represented by a two dimensional time-space matrix. The network is first decomposed into individual sections respecting their spatial relations, and average travelling speeds are measured in specific time intervals for each. The speed of a section i at time interval j is then considered as the intensity of the pixel $p_{i,j}$. Over a specific period of time, e. g. a day, this results in an image used for training a CNN to predict travelling speeds on specific sections. They compare their method to a number of other machine learning approaches, including stacked autoencoders, and report that it outperforms those algorithms. Finally, they argue that the CNN approach is also scalable to large transportation networks.

IV. Dynamic Runtime Exploitation of Knowledge Sources

Calma et al. [21] provide a perspective on meta-learning that is more closely related to the field of OC and puts a focus on how a self-adapting and self-learning system can exploit a diversity of knowledge sources during runtime. They extend the Observer/Controller approach from the OC domain [22] with another higher-level control loop, the meta-learning layer. In the basic Observer/Controller model, a System under Observation and Control (SuOC) represents the productive part of a system which perceives its environment with sensors and interacts with it with actuators. In the adaptation layer, an observer analyses the current status of the SuOC and uses experience and predictions to form a description of the situation that is then used by the controller to decide which adaptations are needed to improve the performance of the SuOC. The meta-learning layer now sits on top of the adaptation layer and should serve as a means for the behaviour of the adaptation layer to self-improve. This layer again consists of an Observer and a Controller component. The goal of the Observer component is to gain a broader understanding of the environmental setting through reflection on the behaviour of the adaptation layer and external knowledge sources. It then sends useful information to the controller component, which controls the adaptation layer based on learned models of external knowledge sources. It not only has to decide on how and which source model to query for information, but also when new information should be gathered from the knowledge sources. This decision process is guided by a learning mechanism.

Example knowledge sources the authors describe in their paper include humans (e. g. domain experts), other similar productive systems and free and open data, e. g. news broadcasts or social media. How efficiently these sources can be queried for information differs: The research field of active learning deals with how machine learning algorithms can and should request help from expert humans in order to improve their performance [23]. The online and self-adapting nature of OC systems can be problematic as human interaction is often only sparsely available. Exchanging knowledge between productive systems can either help those systems achieve their individual or a common goal. Querying free and open data can range from accessing information stored in a structured way in data bases to extracting useful information from large, unstructured data sources like the internet or social media in an intelligent way. Furthermore, these knowledge sources can also be queried and analysed with hybrid methods, combining sources or acquired results.

The authors also give two potential application scenarios of their proposed meta-learning concept: Urban traffic management and industry automation. The former is part of this paper's research question and therefore outlined in Section V.

V. Proposed Framework

In the following section, a general model for traffic control is described that integrates the concept of deep meta-learning with the alternative meta-learning perspective described in Section IV. The model makes use of those two paradigms at different levels and explains how they could be employed together for improving an OTC system.

A. Organic Traffic Control Model

The overall model is derived from the meta-learning model for OC systems described in Section IV and visualised in Figure 2.

The SuOC is now an intersection based, urban traffic control system. This approach is taken in the OTC system where traffic lights are optimised on-line at the intersection level to improve traffic flow [24]. Neighbouring intersection controllers learn to form progressive signal systems, coordinating their green phases. As this organic, distributed approach can only react to measured local traffic flow, ways to improve the system's

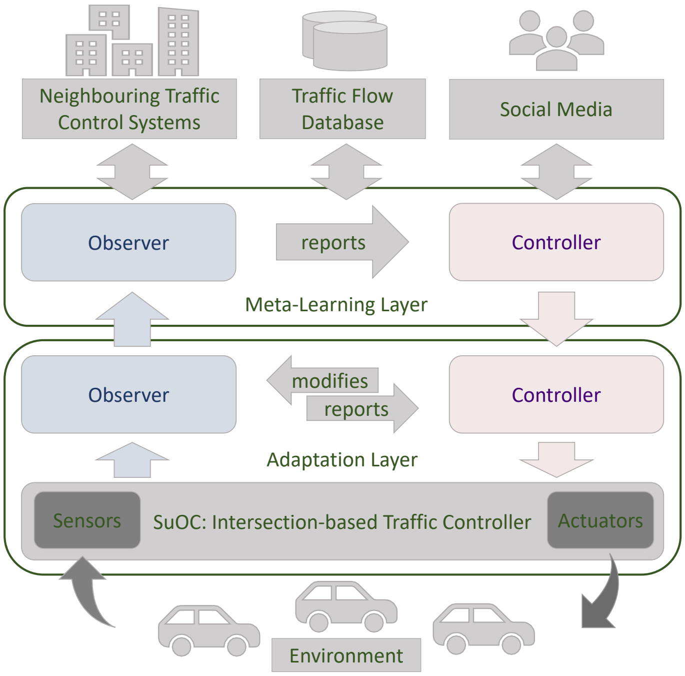

Fig. 2. A variant of the extended Observer/Controller model adapted from [21] for an urban traffic control system. Considered external knowledge sources are neighbouring traffic systems, large amounts of data about traffic flow stored in external databases, and social media channels.

behaviour on a broader scale and in special cases have been proposed. Tomforde et al. [25] extend the model by the introduction of a regional manager component which is a higher-level observer/controller layer that aggregates local knowledge to help with improving coordination on a more global level. For dealing with congested or blocked roads, a dynamic route guidance (DRG) system based on the *Distance Vector Routing protocol* [26] is applied by [27], [28]. Intersection level routing tables are maintained and updated with recommended routes to prominent destinations, like train stations. In a simulated scenario with blocked roads caused by a traffic incident, DRG shows to be especially beneficial in the immediate aftermath of the incident where the intersection controllers have not yet sufficiently adapted their signal plans.

Compared to the approaches above, the method described in Section IV does not solely rely on data available in the OTC system, but has access to external knowledge. Specifically for the use case of traffic control, the meta-learning layer is tasked with optimising the behaviour of the adaptation layer by a higher-level understanding of the traffic environment. In addition to observing the behaviour of the system on a larger time scale, it learns to dynamically exploit external knowledge sources. Relevant sources can be databases containing information about traffic flow in the area, social media channels or the traffic control systems of neighbouring cities. These sources have different associated costs for access, storage and processing [21]: A traffic flow database containing observations over a long period of time requires a large amount of storage space, for example on external servers, while the exploitation of freely accessible social media streams is difficult due to their unstructured nature. When dealing with neighbouring traffic systems, the meta-learning layer is also tasked with finding effective methods for identifying and transferring helpful information. Combining these knowledge sources can also be of help, e. g. detecting large incidents in neighbouring cities through social media channels could indicate that a closer cooperation with the traffic control system of this city might be needed in the immediate future.

B. Traffic Flow Predictor Driven by Deep Meta-Learning

This paper extends the overall model described above by the introduction of a traffic flow prediction component, placed in the meta-learning layer. This predictor can be seen as a source model for the knowledge of predicting network-wide travelling speeds. A broader and more holistic analysis of the traffic flow in the whole environment of the OTC system may be a useful component in guiding the adaptation layer towards more effective learning behaviour. From the viewpoint of dynamically exploiting this knowledge source during runtime, the predictor should also be fast to adapt to new, more specific tasks with few samples of training data. Possible scenarios might be to predict the traffic flow in a specific area after an unpredictable incident has happened, e. g. a car crash that is now blocking specific roads, or evaluating chosen rerouting strategies. In this way, the flow predictor could also be used for the purposes of cooperating with traffic control systems of neighbouring cities and be dynamically employed when certain events are detected from analysis of social media, e. g. protests or large sports events.

To fulfil these properties, the concept of deep meta-learning could be used in learning and updating this component. The predictor component jointly trains a concept generator on large amounts of historic traffic flow data, e. g. found in external databases or collected over the lifespan of the system, while a meta-learner learns to train effective prediction models from tasks of small amounts of specific traffic flow observations that may arise in extraordinary settings. The general structure of the component is visualised in Figure 3. In this model, the concept generator could, for example, be part of one of the two deep learning systems for traffic flow prediction outlined in Section III-C while the meta-learner might be replaced by one described in Section II-C. The meta-learner further does not act on the raw input of the traffic flow observations, but uses the feature representations produced by the concept generator.

The usage of deep representations for the traffic flow data could have two major advantages for the component and the system as a whole: For one, the task of the meta-learner might be made easier (see Section III-B), leading to more accurate prediction models trained on only small data sets. Secondly, the trained representations can be seen as capturing historical and conceptual information about the overall traffic flow and also about specific incidents, acting as a sort of memory component for the overall system.

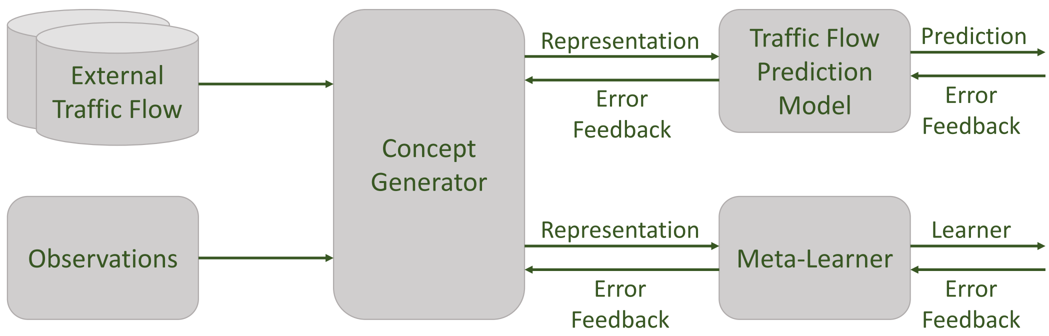

Fig. 3. The traffic flow predictor component of the system proposed in this paper. The framework in [8] is adapted from few-shot image recognition to the task of few-shot traffic flow prediction. Meta-learning is used to help with the fast adaptation and learning of an effective model from small sets of traffic observations that may arise in an extraordinary setting, such as a car crash. The meta-learner acts on a deep representation produced by a concept generator component which is jointly trained on a large external database of traffic flow data.

VI. Conclusion and Future Work

This paper gave a short overview of the broad field of meta-learning and further inspected two different approaches in the field. One is concerned with improving deep learning models, e. g. for performing difficult few-shot learning tasks, and the other perspective embeds meta-learning into the framework of OC systems to exploit external knowledge sources in order to improve a system's performance. An idea of how both approaches could be combined was presented in the form of a deep, meta-learned traffic predictor integrated into an urban traffic control system that also makes use of external knowledge sources, like neighbouring traffic control systems and social media channels. While general ideas of how the interaction in the system might take place were outlined in this paper, a detailed modelling is left to future work. Furthermore, finding a representation of traffic flow that can be used as input across multiple tasks in a deep learning setting could be challenging. Here, the image-based representation mentioned in Section III-C could serve as a starting point.

References

[1] J. Vanschoren, "Meta-learning: A survey," *arXiv preprint arXiv:1810.03548*, 2018.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[3] S. Tomforde, J. Hähner, H. Seebach, W. Reif, B. Sick, A. Wacker, and I. Scholtes, "Engineering and mastering interwoven systems," in *Proceedings of the Workshop on Architecture of Computing Systems (ARCS)*. Vernetzt Digital Elektrisch (VDE), 2014, pp. 1–8.

[4] C. Müller-Schloer and S. Tomforde, *Organic Computing – Technical Systems for Survival in the Real World*. Springer, 2017.

[5] P. J. Rousseeuw and M. Hubert, "Robust statistics for outlier detection," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 73–79, 2011.

[6] S. J. Pan, Q. Yang *et al.*, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[8] F. Zhou, B. Wu, and Z. Li, "Deep meta-learning: Learning to learn in the concept space," *arXiv preprint arXiv:1802.03596*, 2018.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

[10] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[12] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 2625–2634.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[15] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.

[16] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proceedings of the 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255.

[18] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, 2014, pp. 806–813.

[19] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang *et al.*, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[20] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[21] A. Calma, D. Kottke, B. Sick, and S. Tomforde, "Learning to learn: Dynamic runtime exploitation of various knowledge sources and machine learning paradigms," in *Proceedings of the 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. IEEE, 2017, pp. 109–116.

[22] S. Tomforde, H. Prothmann, J. Branke, J. Hähner, M. Mnif, C. Müller-Schloer, U. Richter, and H. Schmeck, "Observation and control of organic systems," in *Organic Computing – A Paradigm Shift for Complex Systems*. Springer, 2011, pp. 325–338.

[23] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, no. 3, p. 4, 2006.

[24] H. Prothmann, J. Branke, H. Schmeck, S. Tomforde, F. Rochner, J. Hahner, and C. Muller-Schloer, "Organic traffic light control for urban road networks," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 2, no. 3, pp. 203–225, 2009.

[25] S. Tomforde, H. Prothmann, J. Branke, J. Hähner, C. Müller-Schloer, and H. Schmeck, "Possibilities and limitations of decentralised traffic control systems," in *International Joint Conference on Neural Networks, IJCNN 2010, Barcelona, Spain, 18-23 July, 2010*, 2010, pp. 1–9. [Online]. Available: https://doi.org/10.1109/IJCNN.2010.5596886

[26] A. S. Tanenbaum, *Computer Networks*, 4th ed. Pearson, 2002.

[27] H. Prothmann, H. Schmeck, S. Tomforde, J. Lyda, J. Hähner, C. Müller-Schloer, and J. Branke, "Decentralised route guidance in organic traffic control," in *5th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2011, Ann Arbor, MI, USA, October 3-7, 2011*, 2011, pp. 219–220. [Online]. Available: https://doi.org/10.1109/SASO.2011.34

[28] H. Prothmann, S. Tomforde, J. Lyda, J. Branke, J. Hähner, C. Müller-Schloer, and H. Schmeck, "Self-organised routing for road networks," in *International Workshop on Self-Organizing Systems*. Springer, 2012, pp. 48–59.