Addressing Catastrophic Forgetting and Beyond: Key Challenges in Continual Learning

Rui Teng[1], Aihui Wang[1,*], Hengyi Li[1], Jinkang Dong[1], Yao Yao[1] and Xueying Hu[1]

[1]The School of Automation and Electrical Engineering, Zhongyuan University of Technology,450007 Zhengzhou, China

Abstract

Current artificial intelligence relies on a one-time training process based on a predefined data set, which remains static during the subsequent reasoning and operation stages. However, a true artificial intelligence system needs to demonstrate the ability to continual learning, that is, to dynamically adapt to the changing environment and new information and to continuously evolve. In the continual learning scenario, catastrophic forgetting is the core problem it encounters. Therefore, this paper first systematically investigates various methods to deal with catastrophic forgetting; secondly, it classifies various methods and deeply analyzes their theoretical basis, specific cases, advantages and disadvantages; finally, it proposes the key challenges and future development directions currently facing continual learning, laying a solid foundation for building an artificial intelligence system with adaptive and self-improvement capabilities.

Keywords

Artificial Intelligence, Continual learning, Catastrophic forgetting, Stability-Plasticity, Experience replay

1. Introduction

Artificial intelligence enables machines to simulate human intelligent behavior to perceive the environment, recognize information and make reasoning decisions [1]. As an important branch of artificial intelligence, deep learning enables automatic extraction of multi-level features directly from raw inputs by building and training multi-layer neural networks to achieve intelligent tasks such as pattern recognition, prediction and decision-making [2]. Deep learning has found extensive applications in natural language processing, image recognition, autonomous driving and other fields [3]. However, current deep learning usually performs one-time training in a static environment, which means that the model parameters are no longer updated and are unable to adapt to constantly changing dynamic scenarios [4]. In addition, model training demands extensive labeling of data samples, which makes its generalization ability for a small number of samples weaker [5]. To address these shortcomings, intelligent systems need to continuously acquire, update, accumulate and utilize knowledge during their life cycle. This ability is called continual learning [6].

The 7th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2025), September 10-11, 2025, Kusatsu, Shiga, Japan

*Corresponding Author

ruiteng@zut.edu.cn (R. Teng); a.wang@zut.edu.cn (A. Wang); lihengyi@zut.edu.cn (H. Li)

0009-0008-0022-0026 (R. Teng); 0000-0002-7531-6240 (A. Wang); 0000-0003-4112-7297 (H. Li)

The primary objective of continual learning is to design algorithms that are able to learn and adapt to continuous data streams [7]. However, when a model sequentially learns new tasks, it usually overwrites the parameters of previous tasks [8], leading to impaired performance on tasks learned earlier, a phenomenon often referred to as "catastrophic forgetting" [9]. This is mainly because in a multi-task environment, the same set of parameters needs to serve both new and previous tasks, resulting in conflicts between the optimal solutions for new and previous tasks when updating parameters [10].

To suppress catastrophic forgetting within the continual learning framework, Researchers have explored different tactics [11], mainly including dynamic architecture-based methods, regularization-based methods, and replay-based methods [12].

The dynamic architecture-based method is to separate the parameters of different tasks by expanding the model structure when faced with new tasks, to avoid parameter conflicts between the new and previous tasks [13]. As an illustration, Iman et al. proposed a continuous and progressive learning system for deep transfer learning - EXPANSE [14]. The regularization-based method prevents drastic parameter changes by adding penalty terms for important parameters of previous tasks in the loss function. For example, Wakelin et al. proposed an analysis of current continual learning algorithms when addressing the image classification problem [15]. Generative replay is to reconstruct the data of previous tasks by training generative models [16]. For example, Shin et al. proposed deep generative replay [17].

This paper systematically reviews methods for preventing catastrophic forgetting in continual learning, summarize the theoretical basis and specific cases of various methods, analyze their advantages and disadvantages, and finally discuss the future research direction of continual learning to provide practical references for promoting the stable application of artificial intelligence in dynamic environments.

The following sections are arranged as follows: Section II classifies the typical methods for solving the problem of catastrophic forgetting; Section III delivers an extensive analysis of the principal challenges and prospective research directions in continual learning; Section IV summarizes the main results of this paper.

2. Taxonomy of Methods

To counteract catastrophic forgetting within continual learning frameworks, this section systematically reviews the primary approaches proposed in relative literature, analyzes their theoretical foundations along with representative cases (Table 1), and evaluates the strengths and weaknesses of each approach.

2.1. Dynamic Architecture-Based Methods

The dynamic architecture–based method modifies the neural network's structure to enable the model to adaptively learn new knowledge while ensuring that previously acquired knowledge remains intact, thereby alleviating the phenomenon of catastrophic forgetting [18]. This approach achieves its goal by designing networks on demand, assigning independent parameters to each task, introducing adaptive submodules, and dividing the model into shared and dedicated components [19].

Table 1
Taxonomy for alleviating catastrophic forgetting in continual learning

Method classification	Theoretical basis	Specific cases
Dynamic architecture-based methods	Assign exclusive parameters to each task, construct task adaptive submodules, and split the model into shared components and dedicated components	PNN achieves knowledge transfer by laterally connecting old column features through adapters within reinforcement learning domains such as Atari and 3D mazes.
Regularization-based methods	Use an additional penalty in the loss to keep previously important parameters from drifting too far	Estimate the importance of previous task parameters through EWC, SI and MAS, and constrain the parameters with high importance. Knowledge distillation: compress and convey the teacher's knowledge into a smaller student model.
Replay-based methods	Save a set of input-output pair samples to the memory module and then mix those samples with the incoming task dataset during model training	Experience replay: iCARL, for example, saves a portion of samples from learned categories and mixes them with new data when learning new categories. Generative replay: In the DRG framework, the Generator generates previous task samples based on a deep generative network and mixes them with new data to train Slover.

Progressive Neural Network (PNN) is the most common method based on dynamic architecture expansion. PNN has a multi-column architecture, and each new task corresponds to a separate network branch. When learning a new task, each layer within the newly added column reuses features extracted by the previous column through the adapter lateral connection to achieve knowledge transfer [20]. PNN starts with one base column, assume a deep neural network with M layers, hidden activations $h_i^{(1)} \in R^{ni}$, let n_i denote the neuron count of layer $i \leq M$. Its parameters $\theta^{(1)}$ are trained to convergence. At the initiation of a new task, the previous-task parameters $\theta^{(1)}$ are frozen and the new column's parameters $\theta^{(2)}$ are randomly initialized. The activation $h_i^{(2)}$ in layer i then takes input from its own previous layer in the column, $h_{i-1}^{(2)}$ and the corresponding layer in the preceding column, $h_{i-1}^{(1)}$ via lateral connections [21]. More generally, for the k-th task, the activation in layer i is given by,

$$h_i^{(k)} = f(W^{(k)} h_{i-1}^{(k)} + \sum_{j<k} U_i^{(k:j)} h_{i-1}^{j}) \tag{1}$$

where $W^{(k)} \in R^{n_i \times n_{i-1}}$ is the weight matrix of the column k of the i-th layer, $U_i^{(k:j)} \in R^{n_i \times n_j}$ denote the lateral links connecting layer $i-1$ of column j with layer i of column k, h_0 is the input of the network. Figure 1 is a schematic diagram of a three-column PNN. The two

columns on the left represent the training of tasks 1 and 2. The third column is dedicated to the final task, which can receive the features of all previously learned old task layers.

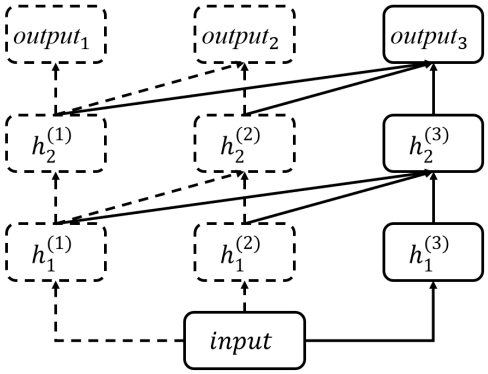

Figure 1: Schematic of a three-column progressive neural network

PNN has achieved continual learning capability with "zero forgetting" in multiple reinforcement learning tasks. For example, in the Atari game experiment, every time PNN learns a new game, it adds a network column and reuses the convolutional features and strategies of the existing pathways through lateral connections. This design not only helps to achieve cross-task knowledge transfer and sharing, but also effectively prevents information interference between different tasks, thereby maintaining a clear separation between tasks.

The advantage of PNN is that it is able to learn in an orderly manner during multi-task training, and it has flexible knowledge transfer capabilities to avoid forgetting previous knowledge [22]. This approach suffers from the drawback that parameter size expands proportionally with the growth in task number. This results in a significant increase in computing resources and storage requirements, which poses challenges to practical applications in environments with numerous tasks or limited resources.

2.2. Regularization-Based Methods

Through the incorporation of regularization terms into the loss function, regularization-based methods constrain alterations to key parameters, thus reducing forgetting [23]. This approach is classified into weight regularization and knowledge distillation.

Weight regularization aims to regularize the model parameters associated with the previous task differently according to their significance [24]. Parameters deemed highly important are constrained during new task training to avoid significant changes, thereby preventing the forgetting of knowledge from earlier tasks. The methods for estimating parameter importance include Elastic Weight Consolidation (EWC), Synaptic Intelligence (SI), and Memory Aware Synapses (MAS) [25].

EWC uses the Fisher Information Matrix to quantify the relevance of network parameters to earlier tasks. The Matrix is defined as follows,

$$F_i = \mathbb{E}_{x\sim\mathcal{D}}\left[\left(\frac{\partial}{\partial\theta_i}\log p(y|x,\theta)\right)^2\right] \tag{2}$$

Here, θ_i denotes the parameter values obtained following training on the prior task, and $\mathbb{E}$ denotes the expectation over the data distribution. $p(y|x,\theta)$ represents the model's output probability distribution.

Once the parameter importance is estimated, a weighted regularization term is embedded within the original loss function during new task training to restrict changes in crucial parameters. The EWC loss function is defined as follows,

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{new}}(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta^*_{old,\,i})^2 \tag{3}$$

$\mathcal{L}_{\text{new}}(\theta)$ denotes the conventional loss used for training the new task, λ denotes the regularization strength hyperparameter, F_i indicates the significance of parameter i estimated via the Fisher Inform, $\theta^*_{old,\,i}$ signifies the i-th parameter value after the prior task's training phase, and θ_i denotes the current value of the i-th parameter.

Throughout the training phase, SI dynamically evaluates the importance of parameters by evaluating the marginal impact of each parameter update to the loss reduction and integrating it along the training path. It then protects these important parameters through weighted regularization terms to reduce the forgetting of previous knowledge. The path integral is expressed by the following formula,

$$\Omega_i = \sum_{k=1}^{T-1} \frac{w_{k,i}}{(\Delta_{k,i})^2 + \epsilon} \tag{4}$$

where w_k reflects each parameter's significance to the loss function, calculated as the product, $w_k = \frac{\partial\mathcal{L}(\vec{x}_k)}{\partial\theta_k}\frac{\partial\theta_k}{\partial\, t}$. $\Delta_k = \theta_k(T) - \theta_k(0)$ indicates the extent of parameter shift after T iterations of training on the k-th task. w_k is updated in each iteration, while Δ_k is updated only after T iterations. ϵ represents a numerical stability term, which is used to prevent the denominator from being too small and causing a numerical explosion. It is generally set to $\epsilon = 0.01$.

The loss function for SI is given as follows,

$$\mathcal{L}_{\text{SI}} = \mathcal{L}_{\text{new}}(\theta) + \lambda\sum_i(\Omega_i(\theta_i - \theta^*_{old,\,i})^2) \tag{5}$$

$\theta^*_{old,\,i}$ denotes the parameter values after training on the previous task, Ω_i is the importance weight of parameter θ_i. λ is a hyperparameter.

MAS evaluates parameter importance by analyzing the autocorrelation of feature activations [26]. We use parameter gradient variance to measure its sensitivity. Gradient variance serves as an indicator of a parameter's importance, with larger values reflecting stronger influence on the model's output. The calculation formula is as follows,

$$\Lambda_i = \mathbb{E}_{x\sim D_{old}}\left[\left|\frac{\partial\|q_x\|^2}{\partial\theta_i}\right|\right] \tag{6}$$

$\mathbb{E}_{x\sim D_{old}}$ represents the expectation of the preceding task dataset, $\|q_x\|$ refers to the output vector of the network for input x in the final layer, θ_i represents the parameter values after training.

The loss function for MAS is given as follows,

$$\mathcal{L}_{\text{MAS}} = \mathcal{L}_{\text{new}}(\theta) + \lambda \sum_i (\Lambda_i(\theta_i - \theta^*_{old,\,i})^2) \tag{7}$$

$\theta^*_{old,\,i}$ denotes the parameter values after training on the previous task, Λ_i is the importance weight of parameter θ_i. λ is a hyperparameter.

In addition to preventing previous knowledge from being covered by adding regularization terms to limit changes in important parameters, there is also a commonly used method called knowledge distillation. Knowledge distillation regularization is different from weight regularization. It transfers the constraint object from the parameter space to the output space, and pays more attention to whether the model preserves the output behavior consistency of earlier tasks as it learns new tasks. Knowledge distillation in general terms is a large neural network (teacher model) in the knowledge condensed and refined to the small neural network (student model), that is, to carry out the migration of knowledge, according to different transfer mechanisms, knowledge distillation is divided into two paradigms: target distillation and feature distillation [27].

Target distillation refers to directly let the student model to imitate the teacher model in the final output layer of the prediction results, The commonly used loss is mainly Kullback-Leibler (KL) Divergence [28] or Cross Entropy [29].

When the KL divergence is employed to quantify the discrepancy between the output distributions of the teacher and student models, the loss can be written as,

$$L_{KD} = KL(P_T(x)||P_S(x)) \tag{8}$$

Where, $P_T(x)$ and $P_S(x)$ represent the probability distribution of the teacher model and the student model for input x output respectively. In addition, the cross-entropy also serves directly as the distillation loss. The calculation formula is as follows,

$$L_{KD} = -\sum_{i=1}^{C} P_i^T(x) log P_i^S(x) \tag{9}$$

where, $P_i^T(x)$ and $P_i^S(x)$ refer to the prediction probability of the i-th category after softmax of the input x by the teacher and the student model respectively. C is the total number of categories.

The output process of feature distillation is different from that of target distillation. It focuses more on the consistency of internal representation rather than aligning only at the output layer [30]. Its output process is usually based on Euclidean distance loss rather than KL divergence. The Euclidean distance loss formula is as follows,

$$L_{KD} = ||\hat{z} - z||_2 \tag{10}$$

where, $\hat{z}$ denotes the logits produced by the prior model, z indicates the new model's logit outputs.

In practical applications, feature distillation is often combined with target distillation, which simultaneously optimizes output consistency and internal feature similarity. This approach is very suitable for model compression, network acceleration, and transfer learning scenarios.

Compared with dynamic architecture-based methods, regularization-based methods are not required to add new network columns when learning new tasks. They only need to impose constraints on important parameters of previous tasks in the loss function. Therefore, they have low computational overhead and simple implementation, but it is difficult to achieve "zero forgetting".

2.3. Replay-Based Methods

The replay-based method is also one of the typical methods to solve "catastrophic forgetting" problem. It saves a set of input-output pair samples into the memory module, and then incorporates these samples with data from the current task for model training [31]. The implementation methods of this method include experience replay and generative replay [32].

Experience replay stores a subset of past task samples in a replay buffer and interleaves these with new-task data during training, ensuring simultaneous learning of current and previous tasks to mitigate forgetting. This method uses real data and has high model stability [33]. Incremental Classifier and Representation Learning(iCaRL) is an incremental learning method based on experience replay. The core group process of iCARL includes three steps. First, classification is performed using the nearest-mean-of-exemplars (NME) rule. Second, exemplars are selected and prioritized with the herding algorithm. Third, representation learning integrates knowledge distillation with prototype rehearsal. This approach enables continuous learning without access to all historical data.

For generative replay, the initial stage consists of the training of a generative model for approximating the data distribution of the previous task, and then the generated samples are incorporated into the training set of the current task to maintain the memory of the previous distribution and alleviate forgetting. Since there is no need to store the original data directly, this method is very suitable in some scenarios where privacy needs to be guaranteed, but the stability of the model is heavily influenced by the effectiveness of the generative model. If the generative component fails to reliably represent the essential characteristics of previous tasks, it may lead to memory degradation or even incorrect transfer, thus affecting the learning stability and performance of the entire system.

At present, the research on replay-based methods mainly focuses on three aspects. It mainly includes improving sample storage efficiency through some core sample selection strategies [34], enhancing the quality of generative models by improving generative model architectures such as diffusion models and Transformer-based generators, and improving the robustness of models by integrating other technologies such as meta-learning and self-supervision [35].

In comparison to the first two methods, the replay-based method has a stronger ability to resist forgetting, but it is largely influenced by how well the generative model performs. If the generated samples are very different from the real data, the effect of alleviating catastrophic forgetting will also decrease.

3. Key Challenges and Future Prospects

Although there are many methods that are capable of easing the problem of catastrophic forgetting to a certain extent, a range of key problems remains to be overcome when facing complex situations in actual application scenarios.

The stability-plasticity dilemma remains an essential challenge [36]. Traditional single strategies often fail to maintain an optimal balance between stability and plasticity in complex scenarios. Therefore, recent research has gradually shifted to hybrid strategies to address this issue. For example, methods combining replay and regularization (such as DER++) not only replay historical samples through a buffer but also use knowledge distillation to constrain the output distribution of the current model to be consistent with that of the historical model on the same input. This allows parameter updates to leverage both hard and soft label information, reducing representation drift and enhancing stability without significantly compromising plasticity. Strategies combining architecture expansion with meta-learning ensure model capacity through dynamic network expansion or sub-network allocation, while leveraging initialization priors or adaptive optimizers derived from meta-learning to rapidly adapt to new tasks while minimizing interference with existing tasks. Future research should continue to explore new hybrid strategies that, through multi-dimensional synergy, enable the system to better balance memory retention and new knowledge learning in complex task flows.

In practical applications, some advanced algorithms face heavy training requirements and are difficult to deploy to edge devices, which makes the model run inefficiently. For instance, an IoT-enabled smart doorbell tasked with real-time pedestrian detection and anomalous behavior recognition must operate under limited computational and memory resources, which restricts the use of large-scale models. Therefore, in the future, it is therefore necessary to investigate efficient algorithms and models capable of running effectively under constrained computational power and storage capacity. To meet the actual needs in edge computing environments.

In the fields of medical and industrial fields, the system is required to maintain strong performance using limited labeled data and to have the ability to protect user privacy. However, too little labeled data will result in few supervisory signals available for incremental tasks. This limits its effective update in the incremental learning process. Therefore, in the future, breakthroughs are needed in incremental learning of a small number of sample categories [37] and unsupervised continual learning [38]. In addition, combining privacy protection mechanisms such as federated learning will also be an important research direction for achieving scalable, secure, and efficient learning systems.

4. Conclusion

This review focuses on catastrophic forgetting, a fundamental challenge in continual learning, and provides a systematic analysis of recent advances in addressing this issue. The methods under review are grouped into three distinct classes, namely dynamic architecture-based methods, regularization-based methods, and replay-based methods. For each category, the paper examines their theoretical foundations, representative techniques, and respective advantages and limitations. Looking ahead, future research on continual learning should address pressing

challenges, including the stability-plasticity dilemma, computational and storage overhead, limited labeled data, and privacy concerns. To this end, integrating multiple strategies, developing efficient algorithms and model architectures, and exploring incremental and unsupervised continual learning, particularly in low-data regimes, are crucial steps toward realizing truly lifelong learning in artificial intelligence systems.

Acknowledgments

The present work was funded by the Henan Province Key Research and Development Program (Grants No. 241111312000), the Henan Province Key International Science and Technology Cooperation Project (Grants No. 251111520400, 252102521009), the Henan Province Key Technologies Research and Development Project (Grants No. 252102211106, 252102320281, 252102221054), the Young Backbone Teacher Program of Zhongyuan University of Technology (Grants No. 2023XQG15).

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] L. Wang, X. Zhang, H. Su, J. Zhu, A comprehensive survey of continual learning: Theory, method and application, IEEE Trans. on Pattern Analysis and Machine Intelligence (2024).

[2] Y. Ge, Z. Li, L. Meng, Yolo-msd: a robust industrial surface defect detection model via multi-scale feature fusion, Applied Intelligence 55 (2025) 1–18.

[3] M. A. Hassan, C.-G. Lee, Forget to learn (f2l): Circumventing plasticity–stability trade-off in continuous unsupervised domain adaptation, Pattern Recognition 159 (2025) 111139.

[4] H. Li, L. Meng, Hardware-aware approach to deep neural network optimization, Neurocomputing 559 (2023) 126808.

[5] Y. Ge, Z. Li, X. Yue, H. Li, L. Meng, Dataset purification-driven lightweight deep learning model construction for empty-dish recycling robot, IEEE Transactions on Emerging Topics in Computational Intelligence (2025).

[6] A. Douillard, A. Ramé, G. Couairon, M. Cord, Dytox: Transformers for continual learning with dynamic token expansion, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 9285–9295.

[7] A. Ashfahani, M. Pratama, Autonomous deep learning: Continual learning approach for dynamic environments, in: Proceedings of the 2019 SIAM international conference on data mining, SIAM, 2019, pp. 666–674.

[8] M. Hasan, A. K. Roy-Chowdhury, A continuous learning framework for activity recognition using deep hybrid feature models, IEEE Transactions on Multimedia 17 (2015) 1909–1922.

[9] G. M. van de Ven, N. Soures, D. Kudithipudi, Continual learning and catastrophic forgetting, arXiv preprint arXiv:2403.05175 (2024).

[10] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, B.-T. Zhang, Overcoming catastrophic forgetting by incremental moment matching, Advances in neural information processing systems 30 (2017).
[11] D. Cheng, Y. Hu, N. Wang, D. Zhang, X. Gao, Achieving plasticity-stability trade-off in continual learning through adaptive orthogonal projection, IEEE Transactions on Circuits and Systems for Video Technology (2025).
[12] F. Wiewel, A. Brendle, B. Yang, Continual learning through one-class classification using vae, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 3307–3311.
[13] S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, R. S. Sutton, Loss of plasticity in deep continual learning, Nature 632 (2024) 768–774.
[14] M. Iman, K. Rasheed, H. R. Arabnia, Expanse, a continual deep learning system; research proposal, in: 2021 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, 2021, pp. 190–192.
[15] J. Wakelin, N. A. Mohammedali, An analysis of current continual learning algorithms in an image classification context, in: 2022 6th International Symposium on Computer Science and Intelligent Control (ISCSIC), IEEE, 2022, pp. 34–39.
[16] S. Kim, L. Noci, A. Orvieto, T. Hofmann, Achieving a better stability-plasticity trade-off via auxiliary networks in continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 11930–11939.
[17] H. Shin, J. K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, Advances in neural information processing systems 30 (2017).
[18] X. Yue, H. Li, L. Meng, An ultralightweight object detection network for empty-dish recycling robots, IEEE Transactions on Instrumentation and Measurement 72 (2023) 1–12.
[19] Y. Yuan, Y. Du, G. Cheng, Class incremental website fingerprinting attack based on dynamic expansion architecture, IEEE Transactions on Network and Service Management (2025).
[20] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, arXiv preprint arXiv:1606.04671 (2016).
[21] T. Moriya, R. Masumura, T. Asami, Y. Shinohara, M. Delcroix, Y. Yamaguchi, Y. Aono, Progressive neural network-based knowledge transfer in acoustic models, in: 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 2018, pp. 998–1002.
[22] Y. Lin, Progressive neural network for multi-horizon time series forecasting, Information Sciences 661 (2024) 120112.
[23] Y. Zhang, Z. Lin, Y. Sun, F. Yin, C. Fritsche, Regularization-based efficient continual learning in deep state-space models, in: 2024 27th International Conference on Information Fusion (FUSION), IEEE, 2024, pp. 1–8.
[24] S. Nokhwal, N. Kumar, Rtra: Rapid training of regularization-based approaches in continual learning, in: 2023 10th International Conference on Soft Computing & Machine Intelligence (ISCMI), IEEE, 2023, pp. 188–192.
[25] X. Zhao, H. Wang, W. Huang, W. Lin, A statistical theory of regularization-based continual learning, arXiv preprint arXiv:2406.06213 (2024).
[26] H. Tercan, P. Deibert, T. Meisen, Continual learning of neural networks for quality

prediction in production using memory aware synapses and weight transfer, Journal of Intelligent Manufacturing 33 (2022) 283–292.
[27] S. Hassan, N. Rasheed, M. A. Qureshi, A new regularization-based continual learning framework, in: 2024 Horizons of Information Technology and Engineering (HITE), IEEE, 2024, pp. 1–5.
[28] J. Gou, L. Sun, B. Yu, L. Du, K. Ramamohanarao, D. Tao, Collaborative knowledge distillation via multiknowledge transfer, IEEE Transactions on Neural Networks and Learning Systems (2022).
[29] H. Bai, J. Wu, I. King, M. Lyu, Few shot network compression via cross distillation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 3203–3210.
[30] S. Li, M. Lin, Y. Wang, Y. Wu, Y. Tian, L. Shao, R. Ji, Distilling a powerful student model via online knowledge distillation, IEEE transactions on neural networks and learning systems 34 (2022) 8743–8752.
[31] S. Ho, M. Liu, L. Du, L. Gao, Y. Xiang, Prototype-guided memory replay for continual learning, IEEE transactions on neural networks and learning systems (2023).
[32] G. Xu, W. Guo, Y. Wang, Memory enhanced replay for continual learning, in: 2022 16th IEEE International Conference on Signal Processing (ICSP), volume 1, IEEE, 2022, pp. 218–222.
[33] T. L. Hayes, N. D. Cahill, C. Kanan, Memory efficient experience replay for streaming learning, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 9769–9776.
[34] H. Hassani, S. Nikan, A. Shami, Improved exploration–exploitation trade-off through adaptive prioritized experience replay, Neurocomputing 614 (2025) 128836.
[35] T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey, Meta-learning in neural networks: A survey, IEEE transactions on pattern analysis and machine intelligence 44 (2021) 5149–5169.
[36] D. Cheng, Y. Hu, N. Wang, D. Zhang, X. Gao, Achieving plasticity-stability trade-off in continual learning through adaptive orthogonal projection, IEEE Transactions on Circuits and Systems for Video Technology (2025).
[37] J. Zhang, L. Liu, O. Silvén, M. Pietikäinen, D. Hu, Few-shot class-incremental learning for classification and object detection: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2025).
[38] M. A. Ma'sum, M. Pratama, R. Savitha, L. Liu, R. Kowalczyk, et al., Unsupervised few-shot continual learning for remote sensing image scene classification, IEEE Transactions on Geoscience and Remote Sensing (2024).