Look At Me, No Replay!
SurpriseNet: Anomaly Detection Inspired Class Incremental Learning

Anton Lee*
Victoria University of Wellington
Wellington, New Zealand
anton.lee@vuw.ac.nz

Yaqian Zhang
University of Waikato
Hamilton, New Zealand
yaqian.zhang@waikato.ac.nz

Heitor Murilo Gomes
Victoria University of Wellington
Wellington, New Zealand
heitor.gomes@vuw.ac.nz

Albert Bifet†
University of Waikato
Hamilton, New Zealand
abifet@waikato.ac.nz

Bernhard Pfahringer
University of Waikato
Hamilton, New Zealand
bernhard@waikato.ac.nz

ABSTRACT

Continual learning aims to create artificial neural networks capable of accumulating knowledge and skills through incremental training on a sequence of tasks. The main challenge of continual learning is catastrophic interference, wherein new knowledge overrides or interferes with past knowledge, leading to forgetting. An associated issue is the problem of learning "cross-task knowledge," where models fail to acquire and retain knowledge that helps differentiate classes across task boundaries. A common solution to both problems is "replay," where a limited buffer of past instances is utilized to learn cross-task knowledge and mitigate catastrophic interference. However, a notable drawback of these methods is their tendency to overfit the limited replay buffer. In contrast, our proposed solution, SurpriseNet, addresses catastrophic interference by employing a parameter isolation method and learning cross-task knowledge using an auto-encoder inspired by anomaly detection. SurpriseNet is applicable to both structured and unstructured data, as it does not rely on image-specific inductive biases. We have conducted empirical experiments demonstrating the strengths of SurpriseNet on various traditional vision continual-learning benchmarks, as well as on structured data datasets. Source code made available at https://doi.org/10.5281/zenodo.8247906 and https://github.com/tachyonicClock/SurpriseNet-CIKM-23

CCS CONCEPTS

• **Computing methodologies → Lifelong machine learning**.

---

*Also with University of Waikato.

†Also with, LTCI, Telecom Paris Institute Polytechnique de Paris.

KEYWORDS

class-incremental continual learning; parameter isolation; anomaly detection; lifelong learning

1 INTRODUCTION

Learning, in the biological sense, is a continuous process of acquiring knowledge and skills, where ideas build upon each other over time. This acquired knowledge is adaptable and adjusts to changes in the world, such as a favorite restaurant relocating or the introduction of new words. Our world is dynamic and constantly changing. In comparison, deep learning is restricted to learning from data that has been homogenized through shuffling to conform to a stationary data distribution. **Continual learning** aims to free neural networks from interpreting data as stationary and enables them to learn continuously from a potentially infinite and evolving stream of data [24].

Continual learning encompasses various scenarios, such as **Task Incremental Learning (Task-IL)** and **Class Incremental Learning (Class-IL)** [29]. Continual learning problems typically involve a sequence of tasks, where the learner is trained sequentially on each task before being evaluated on all tasks collectively. When new knowledge is acquired through training, it degrades old representations, negatively impacting the network's performance on past tasks [6, 20]. Even small changes can have a catastrophic effect. This phenomenon is known as "catastrophic interference" or forgetting.

In the Task-IL scenario, the learner has access to task labels or IDs during both training and testing. However, in many real-world applications, assuming the availability of task labels is impractical. On the other hand, the Class-IL scenario removes the learner's knowledge of the task label during testing. Empirically, there exists a significant performance gap between the Task-IL and Class-IL scenarios. In Class-IL, the learner must differentiate between classes

across different tasks, rather than just within a single task as in Task-IL. Researchers refer to this challenge as "cross-task knowledge," "inter-task confusion," or "inter-task class separation" [11, 12, 19].

To effectively learn continually in the Class-IL scenario, reducing catastrophic interference and acquiring cross-task knowledge becomes essential [11].

In a recent survey, Masana et al. [19] concluded that replay-free methods cannot compete with replay methods. The replay mechanism involves storing a subset of each task and continuously replaying instances to prevent forgetting and gain cross-task knowledge. Unfortunately, replay has some drawbacks. Firstly, replay methods tend to overfit on the replay buffer [4, 19]. Secondly, replaying data through a neural network is inefficient, as the replay instances consume compute resources. Consequently, finding a mechanism for continual learning without replay holds significant value.

In this work, we aim to bridge the gap between Class-IL and Task-IL by explicitly performing task identification with tools inspired by anomaly detection. Specifically, SurpriseNet employs an anomaly detection-inspired auto-encoder to infer tasks and applies parameter isolation to prevent catastrophic interference. It distinguishes itself from similar methods by being compatible with both structured and unstructured data, eliminating the requirement for image-specific inductive biases.

2 RELATED WORK

Continual Learning and Anomaly Detection. Kim et al. [10, 11] demonstrated the connection between task identification and anomaly detection. Their work led to the proposal of CLOM (Continual Learning based on OOD detection and Task Masking), a state-of-the-art replay-free method that combines anomaly detection and fixed network parameter isolation [10, 11]. CLOM uses image rotations and other data augmentations to serve as out-of-distribution training data during contrastive learning. These assumptions make CLOM inappropriate for some vision datasets without modification because different problems require non-trivial data-specific decisions.

In addition to CLOM, we compare SurpriseNet against replay, pseudo-replay, and regularization strategies. We choose the replay strategies: DER [1], A-GEM [2], iCaRL [25], CLOM-c [10], and experience replay. Replay methods reuse a limited buffer of past training data to avoid interference and gain cross-task knowledge, but vary in how samples are chosen and exploited. Our experiments use a small replay buffer of 500 to emphasize sample efficiency [1]. Apart from normal replay, we compare against pseudo-replay strategies: BIR [28] and GR [28]. These methods generate training data to resemble past data and approximate replay. Finally, we compare against regularization strategies EWC [14] and LwF [17], which aim to slow changes to important weights in the network to avoid interference. These methods typically lack a specific mechanism to learn cross-task knowledge and, consequently, perform poorly on class-IL scenarios.

Parameter isolation techniques isolate and protect parameters important to previous tasks, keeping them fixed while allowing other parameters to be learnt for new tasks. "Hard Attention to The Task" (HAT) protects past tasks by learning a mask controlled by a task label embedding [27]. Similarly, "PackNet" protects past tasks by creating a mask using weight magnitude as a heuristic for importance [18]. We do not compare against HAT or PackNet because they rely on a task ID to activate and deactivate the correct subsets of their neural networks.

In contrast to replay methods, SurpriseNet effectively avoids overfitting the replay buffer and enhances training efficiency by eliminating the need to replay samples. While regularization strategies are also replay-free, EWC and LwF encounter difficulties in the Class-IL setting. SurpriseNet shares the concept of fixed network parameter isolation with HAT and PackNet but is not limited to Task-IL since it can infer task IDs. This characteristic aligns SurpriseNet closely with CLOM in terms of theoretical similarity, but both approaches use different mechanisms for parameter isolation and anomaly detection. Our approach has an advantage over CLOM as it can be applied to a wider range of datasets and makes fewer assumptions about the data.

3 METHOD

SurpriseNet divides continual learning into two subproblems: preventing catastrophic forgetting and identifying task IDs. The issue of catastrophic forgetting is tackled by pruning un-important parameters and freezing the important ones, creating "task-specific subsets" for each task. These subsets can be selectively activated or deactivated to solve each task in isolation. The second problem, task identification, is addressed with an auto-encoder. This auto-encoder gauges the reconstruction quality of each task-specific subset, to identify the task. This is based on the assumption that the most accurate reconstruction corresponds to the instance's task. Integrating these techniques produces an effective continual learner.

SurpriseNet builds upon PackNet [18], inheriting the pruning procedure and parameter importance heuristic. However, it incorporates additional features such as task inference, a new pruning schedule, and a hybrid architecture. Figure 1(A-E) provides a summary of the PackNet procedure as utilized in SurpriseNet. Most importantly, unlike PackNet, SurpriseNet is not confined to Task-IL as it is capable of inferring task IDs.

SurpriseNet introduces a novel pruning schedule (**EqPrune**), which allocates an equal proportion of the network to each task. Our experiments reveal that EqPrune consistently performs well when compared to pruning with a sub-optimal λ, although it is often outperformed by a well-chosen λ. EqPrune is a viable default when, hyper-parameter tuning is not possible.

Unlike PackNet, SurpriseNet is a hybrid supervised and unsupervised learner. It consists of an encoder $z = E_\theta(x)$, a decoder $\hat{x} = D_\theta(z)$, and a classifier $\hat{y} = C_\theta(z)$. The encoder and decoder form an auto-encoder (AE) where $\hat{x} = D_\theta(E_\theta(x))$. The classifier connects to the encoder, $\hat{y} = C_\theta(E_\theta(x))$. SurpriseNet is trained to achieve dual objectives. The first objective reconstructs the input in the output using mean squared error as the loss function $L_{rec} = \frac{1}{n} \sum_{i=1}^{n} ||x_i - D_\theta(E_\theta(x_i))||^2$. The second objective classifies instances with cross-entropy loss $L_{cls} = -\frac{1}{n} \sum_{i=1}^{n} y_i \log(C_\theta(E_\theta(x_i)))$. Here, (x_i, y_i) represent an instance in a batch of size n originating from the active task. Adding the losses together forms a hybrid model outputting both reconstructions and a classification at each forward step. As an alternative to AE, we experimented with variational auto-encoders (VAE) [13], which regularize the latent space to follow a probability
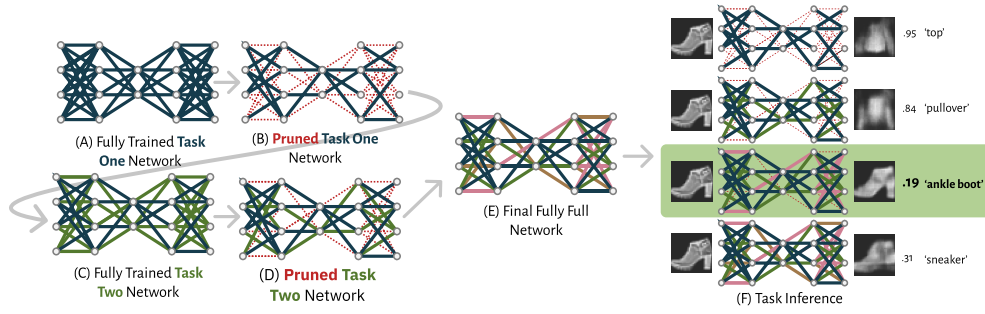
Look At Me, No Replay!
SurpriseNet: Anomaly Detection Inspired Class Incremental Learning

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

Figure 1: (A) Train a neural network on the classes in the first task. (B/D) *Prune* **the network by some proportion i.e.** $\lambda = 50\%$**. (C) The blue weights are** *frozen* **to avoid interference, but pruned weights are reused. (E) "Train, prune, retrain and freeze" repeats until tasks are exhausted. (F) During evaluation, SurpriseNet infers the task by passing each instance through task-specific subsets in the network, generating multiple outputs. The task is identified by comparing the reconstruction quality.**

distribution. The intention was to learn a more structured internal representation.

The auto-encoder infers the task, using a technique inspired by anomaly detection. The auto-encoder model contains a bottleneck that must learn a compressed representation, containing regularities of nominal training data. Consequently, anomalous data belonging to other tasks, is harder to reconstruct [23]. Essentially, the best-suited task-specific subset can be found by measuring the lowest reconstruction loss (e.g $||x - \hat{x}||^2$) of each task-specific subset. Figure 1 (E-F) further describes the SurpriseNet task inference procedure.

SurpriseNet's ability to perform anomaly detection and infer tasks depends on the capabilities of AE and VAE. However, anomaly detection remains a challenging problem, particularly for high-dimensional data such as images. In certain cases, deep generative models assign higher likelihood to anomalous data i.e. lower reconstruction loss [22]. This has been attributed to the emphasis of generative modeling on low-level features [7]. To overcome this limitation, we utilize a pre-trained ResNet18 to extract features and reduce dimensionality, enabling our model to work with higher-level features. van de Ven et al. [28] argues that the outer regions of the human mind, which undergo minimal changes throughout life, perform compression and dimensionality reduction, providing a biological basis for a similar procedure. Our approach, incorporating a pre-trained ResNet18, is designated **SurpriseNetE**.

3.1 Experimental Design

To improve the comparison with other methods, we adhere to the desirable qualities of a continual learning experimental setting outlined by Farquhar and Gal [5]. These qualities include cross-task resemblances, a shared output head, no assumption of task labels at test time, no unconstrained retraining on old tasks, and scenarios involving more than two tasks. Furthermore, we randomly sample the task order to showcase the robustness of our method to task ordering.

SurpriseNet is demonstrated on vision datasets with classes split into multiple tasks: **S-FMNIST** (epochs=20) [31], **S-CIFAR10** (epochs=50) [15], and **S-CIFAR100** (epochs=100) [15]. S-CIFAR100 is particularly popular among researchers [4, 12, 21, 28].

In addition to vision continual learning benchmarks, SurpriseNet is benchmarked on structured data datasets, specifically human activity recognition datasets (HAR). HAR is a desirable domain for continual learning because new unique activities will eventually occur [16]. Jha et al. [9] provides the example of COVID-19, the outbreak significantly changed human behavior, something a continual learning system could adapt to without also forgetting normal behavior. **S-DSADS** (Epoch Budget=200) [26] splits the Daily and Sports Activities dataset of sensor data for 19 activities randomly into 9 tasks of 2 activities each (the remaining activity is ignored). Participants 7 and 8 are used in the test set. **S-PAMAP2** (Epoch Budget=200) [3] splits the Physical Activity Monitoring dataset, containing data of 12 different physical activities, into 6 tasks of 2 activities each. Participants 7 and 8 are used in the test set. The variant of the dataset we utilize is provided alongside our code and uses a sliding window to generate features from raw accelerometer data for both S-DSADS and S-PAMAP2 [8, 9, 30, 32].

To fairly represent each method hyper-parameters must be optimized. Grid searches were conducted for EWC, LWF, ICARL, AGEM, DER, and ER, exploring learning rates and additional strategy-specific hyper-parameters where applicable. CLOM, BIR, and GR, use their default configurations (applying MNIST's for FMNIST), but epochs were reduced to fall within our epoch budget. Minor modifications were made to BIR and GR to ensure compatibility with S-DSADS and S-PAMAP2. Grid searches were also performed to select optimal values for fc-units, latent-dims, and learning-rate for both BIR and GR on those datasets. Regarding SurpriseNet, a grid search was conducted to determine the best prune-proportion, while keeping the learning rates constant. Specifically, S-FMNIST, S-CIFAR10, and S-CIFAR100 employed Adam with a learning rate of 0.0001, while S-DSADS and S-PAMAP2 used Adam with a learning rate of 0.0008. SurpriseNet is a ResNet like auto-encoder for vision datasets and a multilayer perceptron auto-encoder on non-vision datasets. In vision datasets using SurpriseNetE a feature extractor feeds into an auto-encoder multilayer perceptron. Results from grid search and hyper-parameter selections can be found alongside the source code.

Table 1: Mean Final Accuracy (%) ± One Standard Deviation, from 5 runs

Strategy		S-DSADS	S-PAMAP2	S-FMNIST	S-CIFAR10	S-CIFAR100
		(2 classes for 9 tasks)	(2 classes for 6 tasks)	(2 classes for 5 tasks)	(2 classes for 5 tasks)	(10 classes for 10 tasks)
Joint (Non-CL)		75.08 ± 3.33	88.12 ± 1.52	87.86 ± 0.72	90.35 ± 0.83	72.62 ± 0.58
		Replay Method n = Buffer Size				
Experience Replay	n=500	74.40 ± 3.29	**81.30 ± 4.42**	76.08 ± 2.07	44.24 ± 7.51	21.21 ± 0.87
A-GEM [2]	n=500	29.34 ± 6.83	48.41 ± 11.21	53.27 ± 7.86	24.05 ± 5.30	9.58 ± 0.55
DER [1]	n=500	65.02 ± 5.30	69.86 ± 9.19	80.71 ± 0.59	57.41 ± 4.46	25.18 ± 7.35
iCaRL [25]	n=500	43.74 ± 17.97	71.18 ± 2.29	57.22 ± 17.64	53.72 ± 7.00	41.05 ± 0.46
CLOM-c [10]	n=(20 per class)	NA	NA	37.88 ± 9.19	**81.37 ± 0.52**	**71.90 ± 1.53**
		Replay Free				
Naive SGD		9.81 ± 2.15	15.02 ± 2.85	19.85 ± 0.33	18.53 ± 1.36	8.80 ± 0.39
BIR [28]		14.68 ± 4.08	78.01 ± 1.65	76.37 ± 1.87	NA	21.2 ± 1.06
LwF [17]		10.77 ± 0.86	16.38 ± 0.08	20.07 ± 0.54	18.85 ± 0.30	8.80 ± 0.39
EWC [14]		11.13 ± 0.43	16.52 ± 0.11	22.27 ± 5.55	18.60 ± 1.09	7.90 ± 0.72
GR [28]		13.02 ± 2.68	63.40 ± 4.04	76.37 ± 1.87	18.93 ± 1.06	6.92 ± 0.54
CLOM [10]		NA	NA	54.51 ± 4.04	79.78 ± 1.16	71.50 ± 1.61
(ours) SurpriseNet	VAE	76.95 ± 1.08	79.76 ± 1.61	79.20 ± 1.72	37.21 ± 1.66	15.85 ± 0.28
(ours) SurpriseNet	AE	**78.21 ± 3.83**	80.48 ± 1.59	**82.16 ± 0.39**	38.29 ± 2.01	16.49 ± 0.58
(ours) SurpriseNet	VAE EqPrune	77.56 ± 1.22	79.45 ± 1.93	79.47 ± 1.95	34.86 ± 2.00	14.31 ± 0.54
(ours) SurpriseNetE	VAE EqPrune	NA	NA	80.52 ± 2.06	76.04 ± 1.42	48.15 ± 0.68
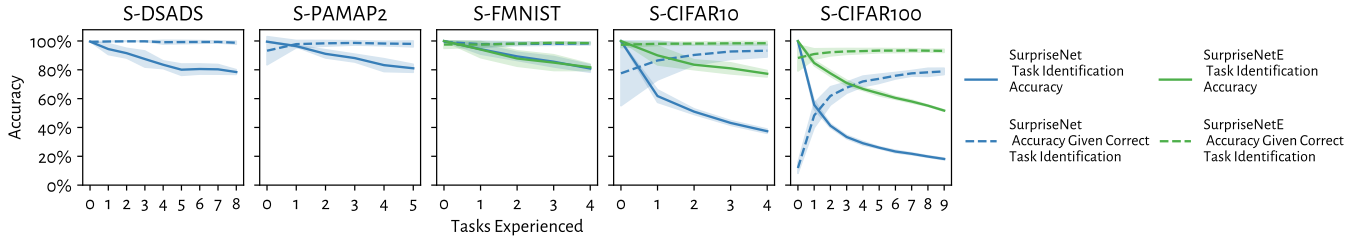
Figure 2: To classify correctly SurpriseNet must first identify a task, then classify within the task. "SurpriseNet VAE EqPrune" and "SurpriseNetE VAE EqPrune" configurations are shown above where each approach is applicable.

4 RESULTS

Table 1 compares SurpriseNet with other strategies. SurpriseNet proves to be effective, surpassing replay methods (n=500) and other replay-free approaches when applied to lower-dimensional datasets such as S-DSADS, S-PAMAP2, and S-FMNIST. However, SurpriseNet's effectiveness diminishes when dealing with higher-dimensional data, as observed in S-CIFAR10 and S-CIFAR100. This limitation can be attributed to the fact that deep generative models are not well-suited for anomaly detection tasks involving high-dimensional data [22].

To address this issue, SurpriseNetE use a pre-trained network to reduce dimensionality. Figure 2 demonstrates the improvement in task identification. SurpriseNetE continues to outperform most other methods in our experiments, with the exception of CLOM, given our buffer budget and epoch budget.

The variances observed in Table 1 exceed those reported by other researchers in similar experiments due to our experimental protocol's inclusion of class order shuffling. To illustrate the effect of this factor, we repeated our iCaRL experiments on S-FMNIST

without shuffling class ordering. The sample's standard deviation decreased significantly from 57.2 ± 17.6 to 65.3 ± 1.2. We strongly argue that shuffling class orders is necessary to evaluate a model's robustness against task composition and to avoid biases in default class orderings.

5 CONCLUSION

SurpriseNet utilizes parameter isolation to stabilize against forgetting and an auto-encoder to learn cross-task knowledge and identify tasks. High-dimensional data challenges our method but we explore a dimensionality reduction technique to mitigate the limitation. Future work could focus on improving the pruning procedure to enable more efficient utilization of capacity or enhancing the anomaly detection capabilities. Although most continual learning works focus on image data, we present a more flexible method. SurpriseNet is a flexible, effective, and replay-free method competitive against replay methods, in some scenarios.

Look At Me, No Replay!
SurpriseNet: Anomaly Detection Inspired Class Incremental Learning

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

REFERENCES

[1] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html

[2] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed El-hoseiny. 2019. Efficient Lifelong Learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Hkf2_sC5FX

[3] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digu-marti, Gerhard Tröster, José del R. Millán, and Daniel Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognit. Lett.* 34, 15 (2013), 2033–2042. https://doi.org/10.1016/j.patrec.2012.12.014

[4] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3057446

[5] Sebastian Farquhar and Yarin Gal. 2019. Towards Robust Evaluations of Continual Learning. arXiv:1805.09733 (Jun 2019). http://arxiv.org/abs/1805.09733 arXiv:1805.09733 [cs, stat].

[6] Robert M French. [n. d.]. Catastrophic Forgetting in Connectionist Networks. 3, 4 ([n. d.]), 8.

[7] Jakob Drachmann Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. 2021. Hierarchical VAEs Know What They Don't Know. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 4117–4128. http://proceedings.mlr.press/v139/havtorn21a.html

[8] Saurav Jha, Martin Schiemer, and Juan Ye. 2020. Continual Learning in Human Activity Recognition: an Empirical Analysis of Regularization. *CoRR* abs/2007.03032 (2020). arXiv:2007.03032 https://arxiv.org/abs/2007.03032

[9] Saurav Jha, Martin Schiemer, Franco Zambonelli, and Juan Ye. 2021. Continual learning in sensor-based human activity recognition: An empirical benchmark analysis. *Inf. Sci.* 575 (2021), 1–21. https://doi.org/10.1016/j.ins.2021.04.062

[10] Gyuhak Kim, Sepideh Esmaeilpour, Changnan Xiao, and Bing Liu. 2022. Continual Learning Based on OOD Detection and Task Masking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*. IEEE, 3855–3865. https://doi.org/10.1109/CVPRW56347.2022.00431

[11] Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. 2022. A Theoretical Study on Solving Continual Learning. *CoRR* abs/2211.02633 (2022). https://doi.org/10.48550/arXiv.2211.02633 arXiv:2211.02633

[12] Jong-Yeong Kim and Dong-Wan Choi. 2021. Split-and-Bridge: Adaptable Class Incremental Learning within a Single Neural Network. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 9 (May 2021), 8137–8145. https://doi.org/10.1609/aaai.v35i9.16991

[13] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *CoRR* abs/1312.6114 (2013).

[14] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR* abs/1612.00796 (2016). arXiv:1612.00796 http://arxiv.org/abs/1612.00796

[15] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. (2009), 60.

[16] Clayton Frederick Souza Leite and Yu Xiao. 2022. Resource-Efficient Continual Learning for Sensor-Based Human Activity Recognition. *ACM Trans. Embed.*

[17] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.

[18] Arun Mallya and Svetlana Lazebnik. 2018. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, 7765–7773. https://doi.org/10.1109/CVPR.2018.00810

[19] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. 2022. Class-Incremental Learning: Survey and Performance Evaluation on Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 1–20. https://doi.org/10.1109/TPAMI.2022.3213473

[20] Michael McCloskey and Neal J. Cohen. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation* 24 (1989), 109–165. https://doi.org/10.1016/S0079-7421(08)60536-8

[21] Martin Mundt, Yongwon Hong, Iuliia Pliushch, and Visvanathan Ramesh. 2023. A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. *Neural Networks* 160 (2023), 306–336. https://doi.org/10.1016/j.neunet.2023.01.014

[22] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2018. Do Deep Generative Models Know What They Don't Know? (Oct 2018). https://arxiv.org/abs/1810.09136v3

[23] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2022. Deep Learning for Anomaly Detection: A Review. *Comput. Surveys* 54, 2 (Mar 2022), 1–38. https://doi.org/10.1145/3439950

[24] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113 (May 2019), 54–71. https://doi.org/10.1016/j.neunet.2019.01.012

[25] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 5533–5542. https://doi.org/10.1109/CVPR.2017.587

[26] Attila Reiss and Didier Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. In *16th International Symposium on Wearable Computers, ISWC 2012, Newcastle, United Kingdom, June 18-22, 2012*. IEEE Computer Society, 108–109. https://doi.org/10.1109/ISWC.2012.13

[27] Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming Catastrophic Forgetting with Hard Attention to the Task. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 4555–4564. http://proceedings.mlr.press/v80/serra18a.html

[28] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. 2020. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications* 11 (2020), 4069.

[29] Gido M. van de Ven and Andreas S. Tolias. 2019. Three scenarios for continual learning. *CoRR* abs/1904.07734 (2019). arXiv:1904.07734 http://arxiv.org/abs/1904.07734

[30] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and Philip S. Yu. 2018. Stratified Transfer Learning for Cross-domain Activity Recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications, PerCom 2018, Athens, Greece, March 19-23, 2018*. IEEE Computer Society, 1–10. https://doi.org/10.1109/PERCOM.2018.8444572

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv:cs.LG/1708.07747 [cs.LG]

[32] Juan Ye, Pakawat Nakwijit, Martin Schiemer, Saurav Jha, and Franco Zambonelli. 2021. Continual Activity Recognition with Generative Adversarial Networks. *ACM Trans. Internet Things* 2, 2 (2021), 9:1–9:25. https://doi.org/10.1145/3440036

Comput. Syst. 21, 6 (2022), 85:1–85:25. https://doi.org/10.1145/3530910