computation

Article

Incremental Learning-Based Algorithm for Anomaly Detection Using Computed Tomography Data

Hossam A. Gabbar *, Oluwabukola Grace Adegboro, Abderrazak Chahid and Jing Ren

Faculty of Engineering and Applied Science, Ontario Tech University (UOIT), Oshawa, ON L1G 0C5, Canada; jing.ren@ontariotechu.ca (J.R.)
* Correspondence: hossam.gaber@ontariotechu.ca

Abstract: In a nuclear power plant (NPP), the used tools are visually inspected to ensure their integrity before and after their use in the nuclear reactor. The manual inspection is usually performed by qualified technicians and takes a large amount of time (weeks up to months). In this work, we propose an automated tool inspection that uses a classification model for anomaly detection. The deep learning model classifies the computed tomography (CT) images as defective (with missing components) or defect-free. Moreover, the proposed algorithm enables incremental learning (IL) using a proposed thresholding technique to ensure a high prediction confidence by continuous online training of the deployed online anomaly detection model. The proposed algorithm is tested with existing state-of-the-art IL methods showing that it helps the model quickly learn the anomaly patterns. In addition, it enhances the classification model confidence while preserving a desired minimal performance.

Keywords: incremental learning; continual learning; computed tomography; anomaly detection; classification; industrial inspection

check for updates

Citation: Gabbar, H.A.; Adegboro, O.G.; Chahid, A.; Ren, J. Incremental Learning-Based Algorithm for Anomaly Detection Using Computed Tomography Data. *Computation* **2023**, *11*, 139. https://doi.org/10.3390/computation11070139

Academic Editor: Yudong Zhang

Received: 5 June 2023
Revised: 4 July 2023
Accepted: 5 July 2023
Published: 10 July 2023

1. Introduction

The term incremental learning (IL) is usually used interchangeably with continual, lifelong, or sequential learning. It refers to a machine learning paradigm that studies how model learning occurs as new data or example(s) emerge, even from an infinite data stream. In particular, it differs from the conventional machine learning approach, such that it assumes that the training examples appear progressively over time as opposed to having the entire training dataset initially [1,2]. Thus, the past knowledge from the previous data is acquired over time and can be extended for future learning and problem solving [3,4]. This concept of incremental learning is used in different applications in various areas such as intelligent robotics, unmanned aerial vehicles, and autonomous driving [5]. Because such an application deals with a dynamic environment, it uses the online or dynamic adaptation of the model to these changes. A major challenge of incremental learning is the catastrophic forgetting or interference that occurs when a model is trained with new information, and this affects the previously learned knowledge. When controlling catastrophic forgetting, the stability–plasticity dilemma comes into play. The stability–plasticity dilemma is the extent to which an incremental learning system is plastic enough to fuse new information and the extent to which it is stable enough to avoid catastrophic interference with consolidated knowledge [2].

Incremental or continual learning techniques can be categorized into three categories: regularization-based, dynamic architectures, and complementary learning systems (CLS) and memory replay approaches, as shown in Figure 1. The regularization-based approach includes the use of knowledge distillation to enforce the similarity between the network of previously learned tasks and the current task [6]. However, this kind of method, the so-called learning without forgetting (LwF), is highly dependent on task relevance and training time. The memory-aware synapses (MAS) method uses the sensitivity of the output function to assign significant weights to the network parameters and is not loss dependent [7]. The Elastic Weight Consolidation (EWC) method is useful in the supervised and reinforcement learning space [8]. The dynamic architecture approach includes the progressive network method, which blocks any change that occurs to the network trained on previous knowledge while expanding the network's architectural properties [9]. The incremental denoising autoencoder method adds neurons for high-loss samples [10]. The model evaluation on the MNIST [11] and CIFAR-10 [12] datasets showed better performance when compared to nonincremental denoising autoencoders [2]. The network structure and weight adaptation method balances the model complexity and empirical risk minimization through network structure and weight adaptation, and the necessary model complexity is learned adaptively by the algorithm [13]. Finally, the CLS and memory replay approach includes different methods based on dual memory and complementary learning systems theory to mitigate catastrophic forgetting. Eden Beloudah et al. [14] introduced incremental learning with dual memory (IL2M) and a fine-tuning-based approach. Srivastava et al. [15] used vector quantization as a replay-based scheme to overcome catastrophic forgetting and applied it to classify chest X-ray pathologies.

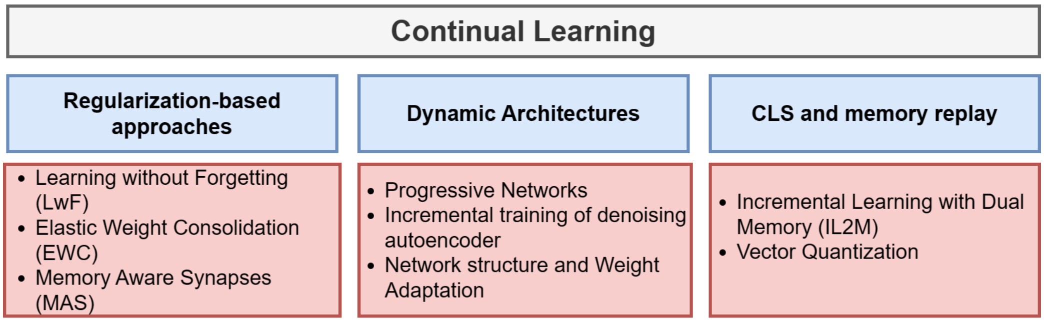


Figure 1. Illustration of the existing continual learning approaches.

In this paper, a new algorithm is proposed to enhance the accuracy of existing IL methods while ensuring stable training towards a desired prediction accuracy for supervised anomaly detection using a classification model. To our knowledge, this is the first work applying a notion of continual learning for classification-based supervised anomaly detection on industrial computed tomography (CT) scans. This paper is organized as follows. Section 2 describes the materials and methods featuring the proposed incremental learning schemes including the dataset and the proposed algorithms in use. Section 3 discusses the obtained results. Finally, the conclusions and future work are outlined in Section 4.

2. Materials and Methods

In this work, incremental learning is applied to an automated inspection framework developed for the X-ray CT inspection of industrial maintenance tools within the nuclear industry [16]. The maintenance tools need to be properly inspected for discrepancies such as scratches and missing components, before and after use for nuclear vault inspection. This manual inspection process can last from a few minutes to months depending on the complexity of the maintenance tools and the available human resources. It is important to note that while this manual tool inspection takes place, the nuclear reactor has to remain shut down to ensure that no missing component of the tool is left behind for safety reasons. However, this outage time of the nuclear reactor comes at a cost to the nuclear power plant operators, with a possible extended outage time resulting in an even higher cost. Hence, the manual tool inspection process can lead to a loss of time, money, and resources. Therefore, an automated inspection solution, which allows for increased reactor availability or decreased outage time and a lower cost of human resources due to manual labor, would play a significant role in this field. The proposed automated tool inspection solution aims to analyze the CT scan of the maintenance tool and predict whether it is defect-free, defective, or anomalous (i.e., with missing components such as spring holders). Moreover, in practical digital industrial applications, the data acquired are dynamically increasing over time. Therefore, incremental learning comes into play by allowing the trained model to learn continually without having to train from scratch each time new data arrive. There are three major types of incremental learning, including task [3], class, and domain incremental learning. In this paper, we focused on domain incremental learning, in which new data streams can appear in the previous or new classes, and the task boundaries are unknown [17] (see Figure 2). The domain incremental learning is used to train a model in a stream across various contexts and applications [18]. It preserves past domain knowledge by enabling the trained model to combine the knowledge learned across different tasks or domains during the training phase [19].

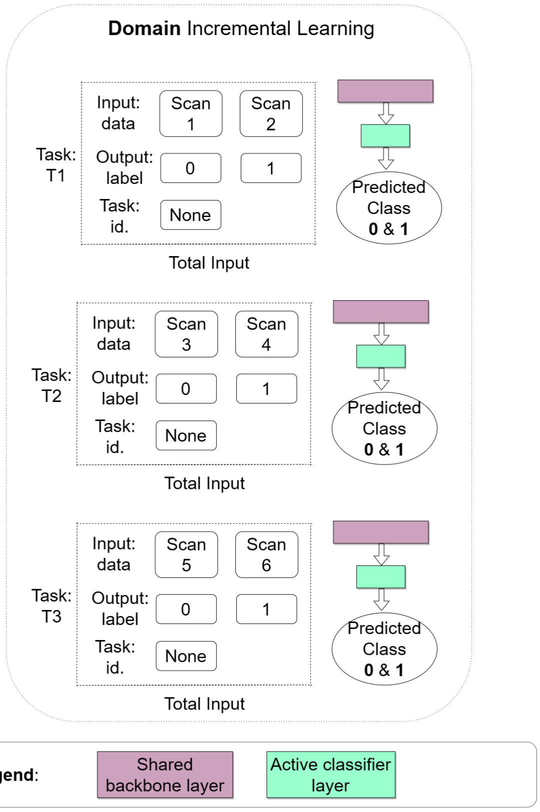


Figure 2. Illustration of the domain incremental learning. (Adapted from [20]).

The proposed incremental learning scheme presents a dynamic and automated training mechanism for stable prediction performance. The proposed scheme is integrated into existing continual learning frameworks for supervised anomaly detection using X-ray computed tomography images.

2.1. Experimental CT Dataset

The dataset used in this work consists of a set of CT images acquired from a case study tool used for nuclear power plant maintenance. The CT image scans were acquired using an industrial CT scanner by Diondo Gmbh [21]. The acquired dataset contained the 2D image projections that were transformed using 3D reconstructions to 2D cross-section or slice images used in the experimental dataset. Figure 3 shows an example of both the projection scan and the reconstruction slice of the nuclear power plant tool scanned in five parts due to the small size of the used scanner [22]. It is worth mentioning that because of the limited computational resources, the studied defects are introduced in the bottom part from which the training dataset is built. Each defect is scanned multiple times to consider the possible noise related to the scanning artifact (see Table 1). In addition, this allows us to have more data to feed the incremental learning tasks streamlining.

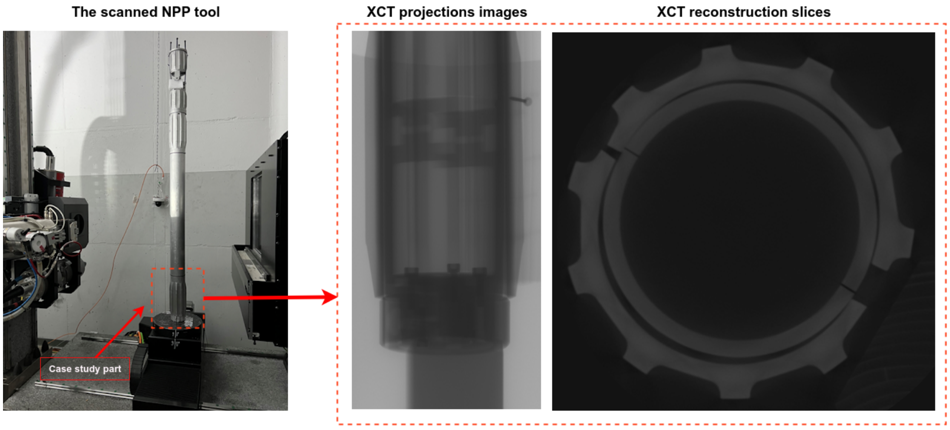


Figure 3. The used NPP tool (**left**) with samples of the 2D projection image (**middle**) and reconstruction slice (**right**).

The CT images of the tool were taken with complete and incomplete components. Each image had a resolution of 1500 × 750 pixels. The dataset consisted of 21 scans categorized as follows:

- **Defect-free scans** [M1 and M2] are two reference scans of a complete tool with no missing pieces.
- **Defective scans** [M3–M21] are scans with various missing components such as spring stoppers, an internal disk pin, spring holders, spring support components, and/or inner disk clips (see Table 1 for more details on the defect types).

Table 1. Summary of the experimental CT dataset provided by our NVS collaborators [22].

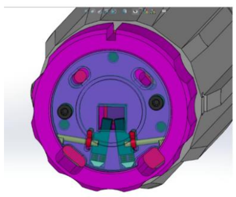
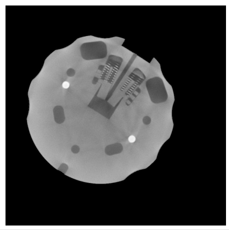
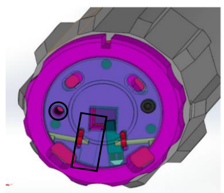
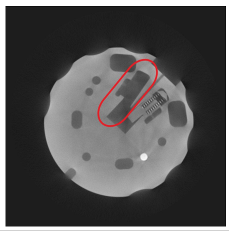
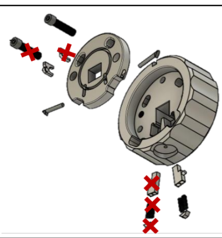
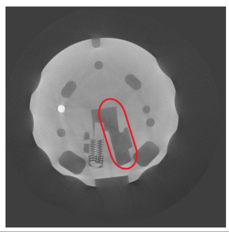
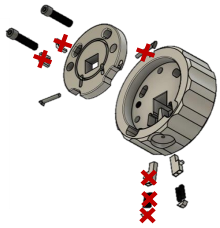
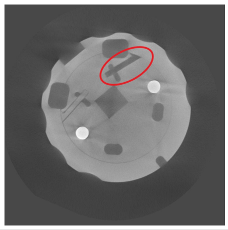
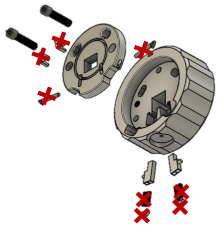
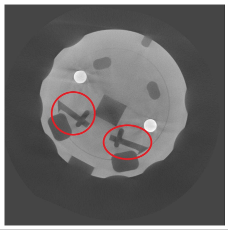
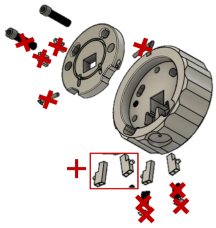
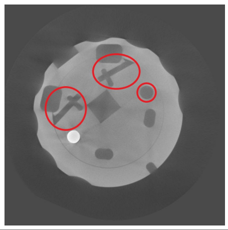
CT-Scan Name	Defect Picture	Image Examples
M1 and M2		
M3 and M4		
M6, M7, M8, and M9		

Table 1. *Cont.*

CT-Scan Name	Defect Picture	Image Examples
M10, M11, M12, and M13		
M14, M15, M16, and M17		
M18, M19, M20, and M21		

2.2. Anomaly Detection Model

For supervised-based anomaly detection, classification-based models are trained to distinguish between normal (associated with the defect-free class) and anomalous (associated with the defective class) slice images. The defect-free and defective slice images were derived from all the 18 scans of our custom dataset CT data types (M1, M2, and M6–M21) depending on the presence of the defect in the slice (see Table 1). The input images were preprocessed by resizing to a shape of 32×32, and normalization with a mean of 0.1000 and a standard deviation of 0.2752 were used. Two different backbone architectures, ResNet18 and Multilayer Perceptron (MLP), were used to evaluate and analyze the proposed classification-based incremental learning tasks. Figure 4 shows an illustration of the classification framework.

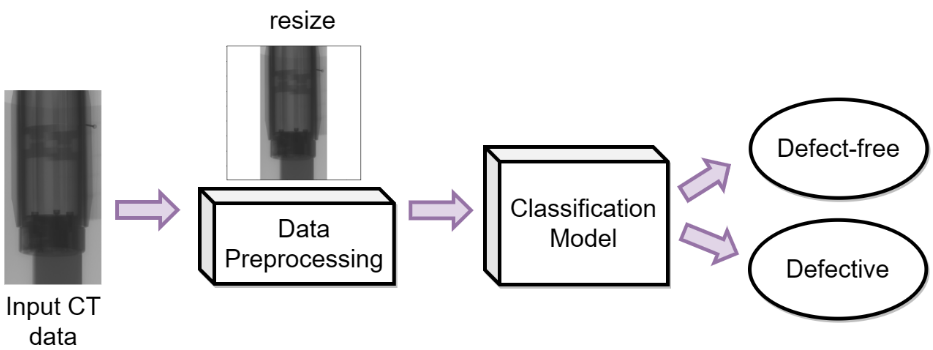


Figure 4. Overview of the classification-based method.

2.3. The Proposed Thresholding-Based Algorithms

In this work, we designed two different algorithms based on the thresholding technique to ensure the optimal possible dynamic training of the classification model. The proposed thresholding scheme has two main advantages. First, it can be used as an add-on to any existing baseline algorithms. In other words, this is similar to a plug-and-play setting whereby the proposed scheme is implemented on existing baseline continual learning methods making it easy to integrate. Lastly, as each incremental task is introduced and trained, the proposed soft thresholding scheme can often enhance the model accuracy of the existing baseline results by training the model following a desired performance, which is the confidence threshold (C_{th}). The two proposed thresholding schemes to be applied to existing IL methods to optimize model performance are defined as follows:

- **Soft-thresholding-based training**: The soft thresholding scheme uses all exemplars of the past data (that is, the data used previously for training) and combines them with the new data.
  The flowchart of the proposed scheme as shown in Figure 5 is summarized as follows. First, the new labelled scans of data are acquired. The next step is to determine whether old labelled scans of the data exist. If old data scans exist, the soft thresholding scheme will combine all the old data scans and new data scans and build a balanced dataset for training. Therefore, an equal number of image instances is represented per class for each training, validation, and testing set. If no old scans exist, only the new data scans will be used to build a balanced training set for training. If there is an existing model, the model is reloaded and used for training; otherwise, a new model is trained. The model is trained and validated using the training and validation sets, respectively, and the confidence score is obtained. The model retrains until the confidence score is greater than a specified confidence threshold (95%) or until the maximum number of times to repeat training (R_{max}) is achieved. The final model is saved. The entire process continues if new labeled scans of data are received.

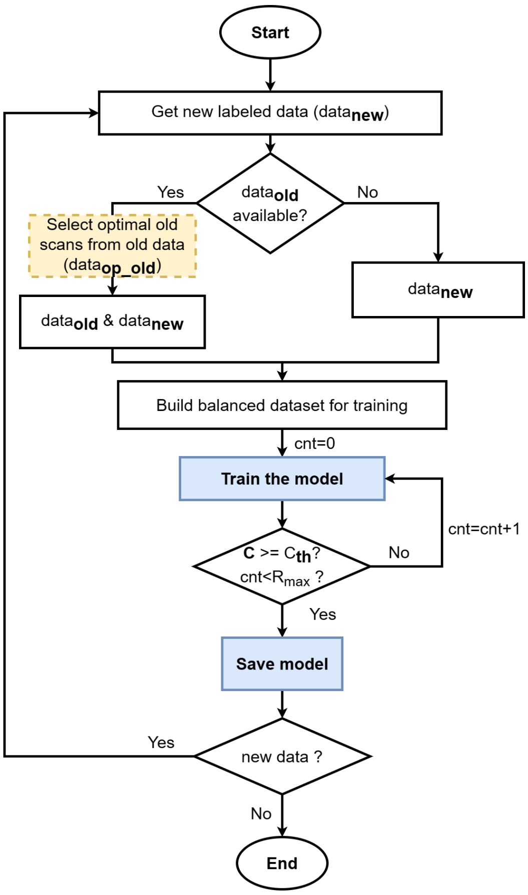


Figure 5. Illustration of the proposed thresholding scheme: soft thresholding (without the yellow box) and selective soft thresholding (with the yellow box).

  Algorithm 1 provides the proposed algorithm.
- **Selective soft-thresholding-based training**: The selective soft thresholding scheme uses the optimal training dataset selection process to select the old data or scans to be

combined with the new data during training. The optimal training dataset selection process involves selecting old scan data whose accuracy fell below the specified threshold of acc_{th}. If no previously trained data fall within the specified threshold interval $[acc_{th}, C_{th}]$, the threshold acc_{th} will automatically be increased gradually (e.g., by 5% each time) until at least one scan is selected to start the training process. If no optimal old data are returned, all old data are selected for training. After the selection of these old data, they are combined with the new data forming the training set of the task as shown in detail in Algorithm 2. This selection process is very important because it recognizes the previously trained data that falls outside the category of the specified accuracy threshold and selects them for further training with the model. Therefore, this saves some memory by selecting a subset of the old scan data to be trained further in combination with the new scan data. In addition, this also provides an opportunity to further train on the data in cases where the model performance was still below the desired threshold. The flowchart of the selective soft thresholding scheme is outlined in Figure 5.

Algorithm 1: The proposed soft thresholding training algorithm

Input : $data_{old}$: all old data
$data_{new}$: new data
$model_{path}$: optimal model
C_{th} : confidence threshold. Default = 95%
R_{max} : maximum number of training repeat
Output: C_{op} : The new model confidence
$model_{op}$: path to the optimal model

◇ Build the dataset
if $data_{old} = \emptyset$ **then**
  $dataset = [data_{new}]$
else
  ◇ Compile the dataset
  $dataset = [data_{old}, data_{new}]$
end

◇ Initializations
$cnt = 0; C_{op} = 0$
◇ Split the dataset into training and validation set
$train_{set}, validation_{set} = dataset$
while $cnt < R_{max}$ *or* $C_{op} < C_{th}$ **do**
  ◇ Train the model
  $model_{path}$ = train the model ($train_{set}$, $model_{path}$)
  $cnt += 1$
  ◇ Compute the model confidence
  $C_{op}, model_{op}$ = test the model ($validation_{set}$, $model_{path}$)
end

return $C_{op}, model_{op}$

Algorithm 2: Select optimal scans from old data

Input : $data_{old}$: data pipeline history
$model_{path}$: trained model
acc_{th}: maximal accuracy threshold
Output : $dataset_{op}$: optimal training dataset

Initialization
$data = \emptyset$
while $data = \emptyset$ **do**
  ⋄ Evaluate the model confidence
  $acc, data$ = evaluate ($model_{path}, data_{old}$)
  ⋄ Increase the accuracy threshold
  acc_{th} += 5
end
⋄ Re-select the optimal scans with $acc \leq acc_{th}$
if $data = \emptyset$ **then**
  $dataset_{op} = data_{old}$
else
  $dataset_{op} = data$
end
return $dataset_{op}$

3. Results and Discussion

The two proposed soft thresholding and selective soft thresholding algorithms were implemented using the PyTorch deep learning framework. For each experimental run, a pipeline of training sets of two classes (defect-free and defective) was built for the training of a given scan stream. These continuous training data are known as a task and consist of a batch of defect-free and defective slice images fed incrementally into the model for training. This kind of incremental learning follows the domain incremental-learning type, where the different batches of data are fed from the same predefined classes (see Figure 2). For each experimental run, the order of the task was shuffled and selected at random among the available CT scans list (see Table 1). This was carried out to observe whether the order of the task pipeline had any significant effect on the incrementally trained model performance. The reported results are the averaged performance of three experimental runs of the shuffled task pipeline for each algorithm as summarized in the different tables.

3.1. Comparison of the Proposed Soft Thresholding Schemes and the Nonincremental Scheme

To evaluate the improvement of the continual training compared to the static training, an experiment was conducted by training a classification model with incremental and nonincremental runs. The obtained performance was compared using the conventional testing accuracy and a confidence score metric defined as the minimum of the testing accuracy of the trained model on the last five tasks. Table 2 shows that the soft thresholding achieved higher accuracy as expected because the model was gradually trained compared to the nonincremental cases, where the model was tested on all the data at once. In addition, the selective soft thresholding scheme seemed to achieve better task-wise performance by obtainiing a higher confidence score of 88.07% compared to the soft thresholding scheme. The generalization of this assumption was evaluated by applying the proposed thresholding scheme to the existing incremental learning scheme method.

Table 2. The comparison of the average performance over three runs of the nonincremental and the proposed incremental schemes on 10 tasks.

Scheme	Number of Tasks	Accuracy (%)	Confidence Score (%)
Soft Thresholding	10	**100**	85.33
Selective Soft Thresholding	10	86	**88.07**
Nonincremental approach	NA	99.67	NA

3.2. Performance Sensitivity Analysis Using Existing Incremental Learning Methods

The proposed scheme was integrated with some existing baseline methods that consist of four regularization-based continual learning methods which are: Elastic Weight Consolidation (EWC), Online EWC, Synaptic Intelligence (SI), and Memory-aware Synapses (MAS) [20]. The four continual learning methods were tested using the ResNet18 and multilayer perceptron (MLP) model architectures. The proposed thresholding schemes handled the batch selection in the training process. The experiments included the baseline (i.e., the original existing methods), selective (i.e., the baseline + optimal training dataset selection), soft thresholding, and the selective soft thresholding scheme. The default training hyperparameters used for the experiments are outlined in Table 3. The default regularization coefficients defined by Hsu et al. [20] were used with 100, 700, 3000, and 10,000 for the EWC, Online EWC, SI, and MAS methods, respectively.

Table 3. Summary of the default hyperparameters used to implement the existing IL methods [20].

Parameter	Number of Tasks
Epoch	100 (per task)
Batch size	128
Model architecture	MLP and ResNet18
Loss function	Cross entropy
Optimizer	Adam
Learning rate	0.001

Figures 6 and 7 show the obtained testing accuracy of the incrementally trained model using the MLP and Resnet18 models, respectively. The figures show the performance of the different IL baseline algorithms compared to the integrated proposed algorithms. The selective implementation denoted as (S) refers to the use of the optimal training dataset selection, and the soft thresholding scheme is denoted as (+). For instance, the different implementations of the EWC method are denoted as the baseline EWC, selective baseline EWC(S), soft thresholding EWC(+), and selective soft thresholding EWC(S+). Table 4 shows that the nonselective scheme, which comprises the baseline and the soft thresholding scheme, achieved a higher accuracy performance by the last task (T18) compared to the selective scheme for both model architectures. In addition, the soft thresholding algorithm improved most of the baseline IL methods. Overall, it achieved the highest last accuracy of 99.29% using the Resnet18 architecture.

Table 5 summarizes the best achievable average accuracy and standard deviation computed from tasks T14–T18 (i.e., the average and standard deviation of the last five tasks) across all IL methods. It demonstrates that the EWC(+) method outperformed the other methods in terms of the average accuracy.

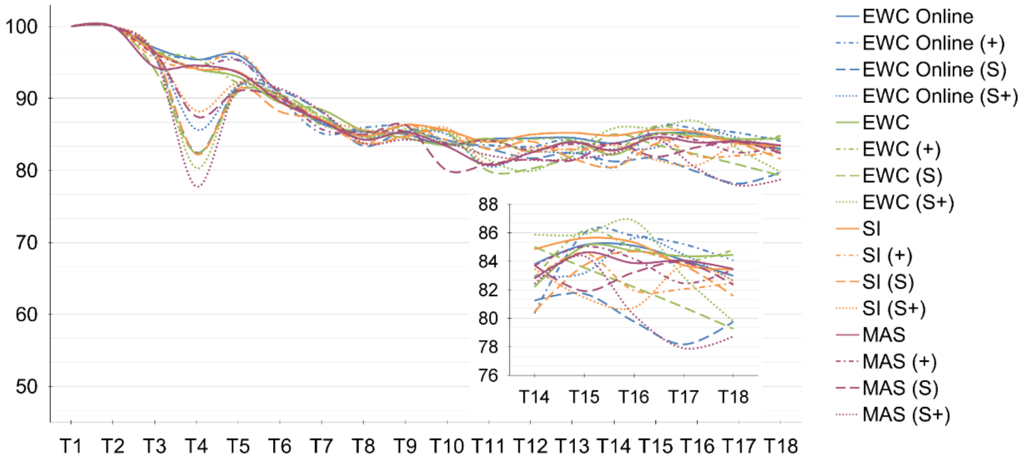


Figure 6. Performance comparison on custom data using the MLP architecture.

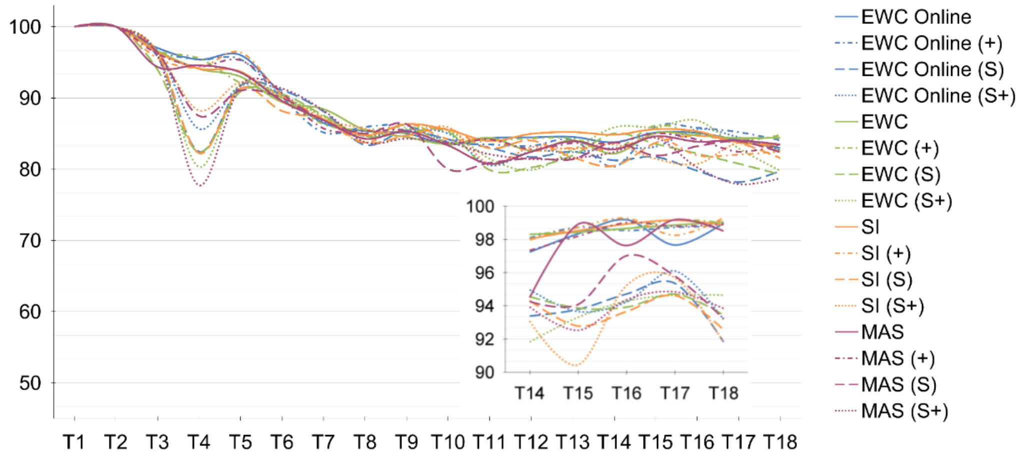


Figure 7. Performance comparison on custom data using the ResNet18 architecture.

Table 4. The averaged last accuracy performance comparison of the three averaged experimental runs of 18 tasks.

		Nonselective Scheme		Selective Scheme	
Model	**Method**	**Baseline**	**Soft Threshold**	**Selective Baseline**	**Selective Soft Threshold**
MLP	EWC	84.44 ± 0.91	**84.79 ± 1.38**	79.26 ± 5.87	79.79 ± 1.88
	Online EWC	82.97 ± 0.35	**84.05 ± 1.71**	79.76 ± 0.83	82.61 ± 1.16
	SI	**83.43 ± 0.74**	82.48 ± 1.65	81.57 ± 1.48	82.63 ± 0.13
	MAS	**83.45 ± 2.93**	83.44 ± 1.54	82.34 ± 0.95	78.73 ± 2.85
ResNet	EWC	**99.06 ± 0.23**	98.98 ± 0.28	93.44 ± 0.98	94.64 ± 1.56
	Online EWC	**98.99 ± 0.16**	98.79 ± 0.12	91.83 ± 2.64	93.18 ± 1.51
	SI	98.87 ± 0.28	**99.29 ± 0.05**	92.55 ± 1.34	91.78 ± 1.23
	MAS	98.50 ± 0.57	**98.91 ± 0.05**	93.22 ± 0.50	93.84 ± 0.60

Table 5. The summary of the best achievable accuracy performance comparison of the three averaged experimental runs of the T14–T18 (last five) tasks.

	Experiment	MLP		ResNet	
Average Accuracy (%)	*Baseline*	SI	**84.58 ± 0.86**	SI	98.70 ± 0.39
	Selective	MAS (S)	83.02 ± 0.78	MAS (S)	94.87 ± 1.35
	Soft Thresholding	EWC (+)	84.53 ± 1.10	**EWC (+)**	**98.72 ± 0.32**
	Selective Soft Thresholding	EWC (S+)	84.25 ± 2.59	EWC Online (S+)	94.42 ± 1.03

For the confidence score, Table 6 shows the obtained performance for different incremental learning schemes. The table demonstrates that the EWC method outperformed other methods specifically and achieved the highest confidence score of 98.30% using the ResNet architecture with the soft thresholding EWC(+) and baseline EWC schemes. However, Figure 8 demonstrates a drop in the performance of the EWC compared to the EWC(+) at task T6. Figure 8 shows that the EWC(+) scheme attained a more stable performance by T5, unlike the EWC scheme, which encountered a drop in performance at task 5 (T5) but recovered afterward. Overall, the incrementally trained model using the proposed framework helps in recognizing new defect patterns from new data streams while training the model only when needed. In addition, it adapts to dynamic data characteristics and size by using the selective soft thresholding scheme when the accumulated data are outside the computational resources limits. The proposed schemes help to improve the quality control and effectiveness of the industrial inspection applications [23].

Table 6. Confidence score for each scheme using the MLP and ResNet model architecture for three experimental runs.

	Experiment	MLP		ResNet	
Confidence Score	*Baseline*	SI	**83.43**	EWC	**98.30**
	Selective	MAS (S)	81.91	EWC (S)	93.44
	Soft Thresholding	EWC (+)	82.81	**EWC (+)**	**98.30**
	Selective Soft Thresholding	EWC Online (S+)	82.61	EWC Online (S+)	93.18

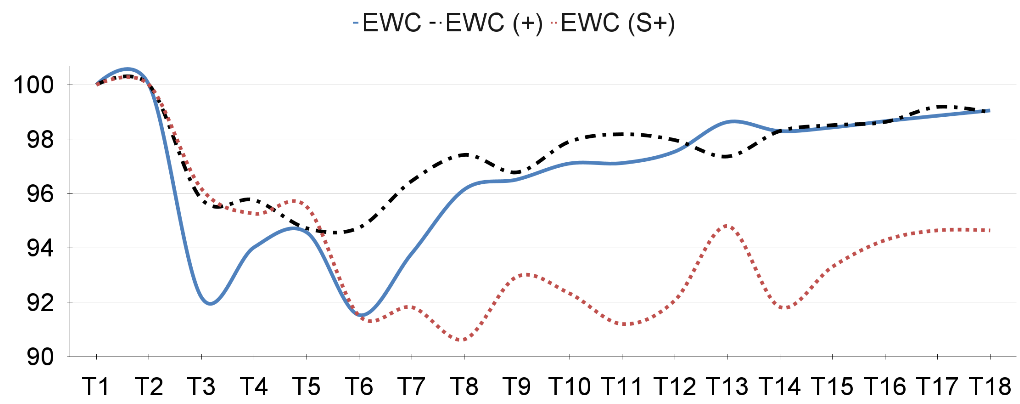


Figure 8. EWC performance comparison on custom data using ResNet18.

3.3. Limitations

The performance comparison with existing continual learning baselines shows that the proposed soft thresholding method seems to be more adaptable for dynamic and optimal training to enhance the model prediction. However, the major limitation is the need for high-quality online data annotation preferably verified by a human in the loop framework. Finally, the algorithm has an extended (long) training time as more sequences of data or tasks are trained incrementally. Therefore, this might cause the model to overfit. A possible solution is to decrease the number of retraining repetitions to enable shorter and early stopping that exits the training once the learning starts to saturate.

4. Conclusions and Future Work

Incremental learning, also known as continual learning or lifelong learning, is an adaptive algorithm that learns progressively over time from a continuous stream of information. During this process, new knowledge is learned while keeping the previously learned experiences. In this work, a new soft thresholding scheme was introduced to optimize the model prediction of existing incremental learning frameworks. The obtained results show that the proposed algorithm could achieve a steady performance around the desired prediction accuracy for supervised-based anomaly detection using CT images. In the future, the proposed schemes will be tested for multiclass and/or multilabel classification tasks. Finally, they could also be adapted for reinforcement learning, where a robot learns incrementally a specific manipulation using parameter estimation based on data-driven models.

Author Contributions: H.A.G. was the principal investigator. H.A.G. and J.R. designed the system idea for the CT technology. O.G.A. and A.C. designed and developed the incremental learning algorithm. O.G.A. conducted and implemented the ablation studies for the incremental/continual learning experiments. O.G.A., A.C., H.A.G. and J.R. contributed to the paper writing. All authors read and approved the final manuscript.

Funding: This research was funded by Mitacs Grant Number 215457.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: This work was supported by New Vision Systems Canada Inc. (NVS), Mitacs, and Google. The authors would also like to thank the collaborators from Diondo and Fraunhover, Germany.

Conflicts of Interest: The authors have declared that no competing interests exist.

Abbreviations

The following abbreviations are used in this manuscript:

CT	Computed Tomography
EWC	Elastic Weight Consolidation
IL	Incremental Learning
MAS	Memory-aware Synapses
MLP	Multilayer Perceptron
NDT	Nondestructive Testing
SI	Synaptic Intelligence

References

1. Geng, X.; Smith-Miles, K. Incremental Learning. In *Encyclopedia of Biometrics*, Li, S.Z., Jain, A., Eds.; Springer: Boston, MA, USA, 2009; pp. 731–735. [CrossRef]
2. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual Lifelong Learning with Neural Networks: A Review. *Neural Netw.* **2019**, *113*, 54–71. [CrossRef] [PubMed]
3. Chen, Z.; Liu, B. Lifelong Machine Learning. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*; Springer: Cham, Switzerland, 2018; p. 187. [CrossRef]
4. Delange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3366–3385. [CrossRef] [PubMed]
5. Luo, Y.; Yin, L.; Bai, W.; Mao, K. An Appraisal of Incremental Learning Methods. *Entropy* **2020**, *22*, 1190. [CrossRef] [PubMed]
6. Li, Z.; Hoiem, D. Learning without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef] [PubMed]
7. Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; Tuytelaars, T. Memory Aware Synapses: Learning What (not) to Forget. In *Proceedings of the Computer Vision–ECCV 2018, Munich, Germany, 8 Septemeber 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Cham, Switzerland, 2018; Volume 11207, pp. 144–161. [CrossRef]

8. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming Catastrophic Forgetting in Neural Networks. *arXiv* **2017**, arXiv:1612.00796.
9. Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive Neural Networks. *arXiv* **2022**, arXiv:1606.04671.
10. Zhou, G.; Sohn, K.; Lee, H. Online Incremental Feature Learning with Denoising Autoencoders. In Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, 21 April 2012; Volume 22, pp. 1453–1461.
11. Mehrotra, K.; Mohan, C.; Ranka, S. *Elements of Artificial Neural Networks*; The MIT Press: Cambridge, MA, USA, 1996. [CrossRef]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
13. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 1 October 2014; pp. 1746–1751.
14. Belouadah, E.; Popescu, A. IL2M: Class Incremental Learning with Dual Memory. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October 2019; pp. 583–592. [CrossRef]
15. Srivastava, S.; Yaqub, M.; Nandakumar, K.; Ge, Z.; Mahapatra, D. Continual Domain Incremental Learning for Chest X-ray Classification in Low-Resource Clinical Settings. In *Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health*; Springer: Cham, Switzerland, 2021; Volume 12968, pp. 226–238. [CrossRef]
16. Gabbar, H.A.; Chahid, A.; Khan, M.J.A.; Adegboro, O.G.; Samson, M.I. CTIMS: Automated Defect Detection Framework Using Computed Tomography. *Appl. Sci.* **2022**, *12*, 2175. [CrossRef]
17. Belouadah, E. Large-Scale Deep Class-Incremental Learning. Computer Vision and Pattern Recognition [cs.CV]. Thesis, Ecole Nationale Supérieure Mines-Télécom Atlantique, Paris, France, 2021.
18. Van de Ven, G.M.; Tuytelaars, T.; Tolias, A.S. Three Types of Incremental Learning. *Nat. Mach. Intell.* **2022**, *4*, 1185–1197. [CrossRef] [PubMed]
19. Wei, X.; Liu, S.; Xiang, Y.; Duan, Z.; Zhao, C.; Lu, Y. Incremental Learning based Multi-Domain Adaptation for Object Detection. *Knowl.-Based Syst.* **2020**, *210*, 106420. [CrossRef]
20. Hsu, Y.-C.; Liu, Y.-C.; Ramasamy, A.; Kira, Z. Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines. *arXiv* **2019**, arXiv:1810.12488.
21. Diondo X-ray Systems and Services. Available online: https://www.diondo.com/ (accessed on 2 July 2023).
22. New Vision Systems Canada Inc. Available online: https://nvscanada.ca/ (accessed on 2 July 2023).
23. Schimanek, R.; Bilge, P.; Dietrich, F. Inspection in High-Mix and High-Throughput Handling with Skeptical and Incremental Learning. *TechRxiv* **2023**, techrxiv:23284049.