# Resilience-aware MLOps for resource-constrained AI-system

Viacheslav Moskalenko[1], Alona Moskalenko[1], Anton Kudryavtsev[1] and Yuriy Moskalenko[1]

[1] *Sumy State University, 116, Kharkivska st., 40007 Sumy, Ukraine*

**Abstract**

Artificial intelligence systems are increasingly used in security-critical applications with limited computing resources, which makes them vulnerable to such disturbances as adversarial attack noise, out-of-distribution data, and fault injections. To absorb disturbances and adapt the AI system during its life cycle, it is necessary to expand the Machine learning operations structure with stages related to the implementation of resilience mechanisms. In this case, increasing resilience in one form or another is associated with the introduction of redundancy in the form of additional resources to absorb disturbances and quickly recover performance. It is proposed to provide Affordable Resilience for resource-constrained AI systems by implementing a resilience optimization stage and adding add-ons with a small number of parameters that will allow for uncertainty calibration and rapid adaptation to labeled and unlabeled data. This approach is also intended to separate the work related to the development and deployment of the basic AI model that solves the applied problem and the work related to ensuring resilience in the Machine learning operations structure. The experiments were conducted on CIFAR-10 and CIFAR-100 datasets using the MobileViT network, which is a modern network architecture for visual image analysis in conditions of limited computing resources. We have experimentally confirmed the increase in the resilience of an AI system at different stages of its life cycle by implementing the stages of resilience optimization, uncertainty calibration, and Test-Time Adaptation. Post-hoc resilience optimization improved robustness to fault injection by 5% and robustness to adversarial attack by 7%. Moreover, tuning with 10% of the test data allowed for an additional 6% increase in robustness to fault injection and 7.1% increase in robustness to adversarial attack on the new data. Also, the use of post-hoc resilience optimization increased the integral indicator of resilience to task changes by 10.5%. Post-hoc uncertainty calibration makes it possible to further increase the robustness of fault injection models by an average of 4.4% and the robustness to adversarial attacks by an average of 1.3%. Test-Time Adaptation increases robustness to Fault Injection by 6.9% and robustness to Adversarial Attack by 4.72%.

**Keywords**

MLOps, Resilience, Artificial Intelligence, Confidence Calibration, Optimization, Test-Time Adaptation

## 1. Introduction

Artificial intelligence (AI) systems are increasingly deployed on safety-critical but resource-constrained devices (such as unmanned aerial vehicles, autonomous robots, autopilots). All AI systems to some extent are susceptible to data noise through adversarial attacks, novelty in data, concept drift, and the injection of faults into the memory of neural weight [1]. Moreover, AI systems with constrained resources are more vulnerable to disturbances. The ability to absorb disturbances (robustness), graceful degradation due to the impact of disturbances that could not be absorbed, and rapid adaptation to new disturbances are considered to be the key features of resilient system [2].

Traditional approaches in Machine Learning Operations (MLOps) predominantly emphasize data integrity, model efficiency, and security aspects but often fall short in tackling resilience issues as highlighted in [3]. Considering the critical decisions entrusted to AI systems, insufficient resilience may lead to severe implications, including unreliable performance and financial losses.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

v.moskalenko@cs.sumdu.edu.ua (V. Moskalenko); a.moskalenko@cs.sumdu.edu.ua (A. Moskalenko); (A. Kudryavtsev), yuriy.mosk@gmail.com (Y. Moskalenko)

0000-0001-6275-9803 (V. Moskalenko); 0000-0003-3443-3990 (A. Moskalenko); 0000-0003-0967-0185 (A. Kudryavtsev); 0009-0002-3635-3337 (Y. Moskalenko)

Therefore, this study aims to enrich MLOps methodology by incorporating resilience as a fundamental component.

**The object of the research** is MLOps process under resource constraints and the impact of various types of disturbances on the AI system. **The subjects of the research** are the architecture of MLOps and methods of ensuring the resilience of an AI system with limited resources to various types of disturbing influences. **The goal** is to develop a new MLOps methodology that ensures the resilience of resource-constrained AI-system to such negative factors as adversarial attacks, fault injections, drift, and out-of-distribution of data.

## 2. Related works

The concept of resilience in AI systems is not new and has been examined across various fields, including cybersecurity, manufacturing, and even autonomous vehicles. Techniques like adversarial training [4], fault-aware training [5], and uncertainty quantification [6] have been employed to improve resilience. The paper [7] proposes the concept of Secure Machine Learning Operations paradigm, but without proposals for effectively protecting the same AI system from different types of threats. The issue of ensuring compatibility and computational efficiency of combining different methods to provide resilience and efficiency of the AI system is not considered.

A number of works consider the aspects of the MLOps methodology for AI systems with resource constraints [8, 9]. In addition to typical MLOps aspects, more attention is focused on model optimization by quantization or pruning of weights and knowledge distillation. However, the online adaptation of resource-constrained AI systems to changes is considered only for shallow machine learning models or within the framework of federated learning, which requires network communication with distributed nodes. In [2, 4], various types of destructive disturbances for AI systems are considered, to which systems with limited resources are most vulnerable. However, most existing MLOps frameworks are designed to ensure efficient operation rather than resilience to various disturbances in resource-limited environments.

The papers [10, 11] consider the use of Parameter-Efficient Tuning methods as one of the effective approaches to increase computational efficiency and speed of adaptation of AI system to changes. The papers [12, 13] consider approaches to Test-Time Adaptation, which increases the efficiency of adaptation to novelty in the absence of labeled data. The papers [14, 15] consider meta-learning algorithms for increasing the robustness and efficiency of few-shot learning. These methods are promising to be combined in the MLOps methodology, but the possible configurations of their combination are not well studied.

## 3. Resilience-aware MLOps architecture

Key concepts in MLOps include the delineation of duties and the collaboration among different teams. Specialized platform-level solutions that bolster the resilience of any AI model assign the task of updates and maintenance to a dedicated team of AI resilience specialists. Additionally, to enhance the delineation of responsibilities, new stages in MLOps that focus on resilience should be introduced as subsequent (post-hoc) procedures.

Figure 1 shows a diagram of the proposed resilience-aware MLOps, which additionally includes the stages of Post-hoc Resilience Optimization, Post-hoc Uncertainty Calibration, Uncertainty Monitoring, Test-Time Adaptation and Graceful Degradation. In addition to Uncertainty Monitoring, the Explainable AI mechanism can be used to assist decision-making by the human to whom control is delegated in case of uncertainty. The article [16] questions the adequacy of existing methods of explaining decisions, so the explanation mechanism will be excluded from further consideration, but for generality, the diagram shows this MLOps stage.

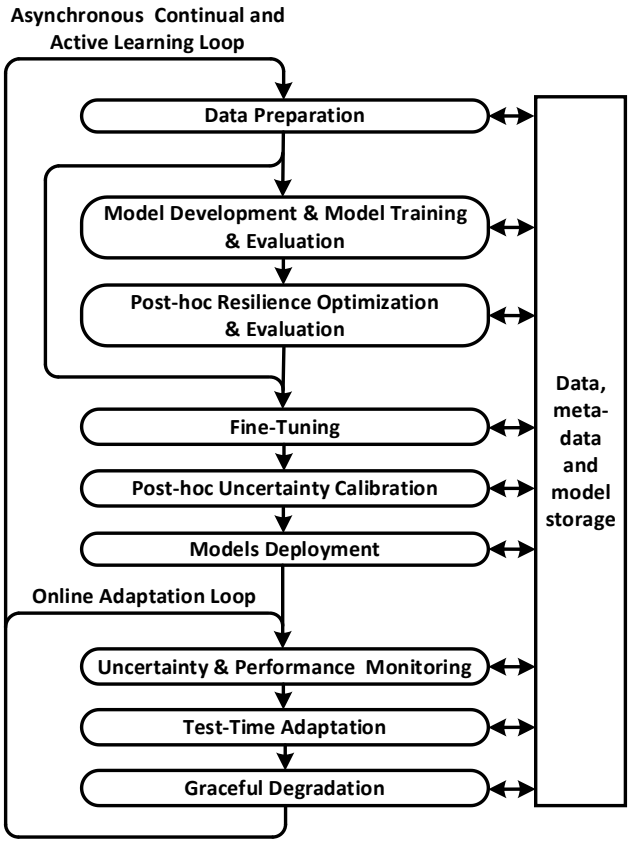
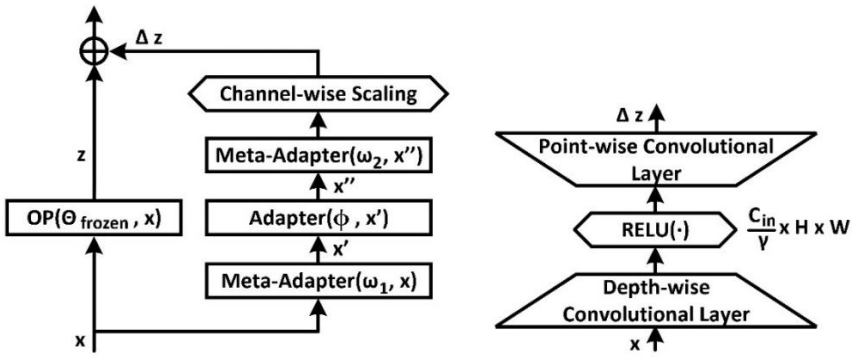



**Figure 1:** Basic stages of resilience-aware MLOps

In the phase focused on enhancing resilience, it is suggested to attach computationally efficient (meta-)adapters to the existing model to improve robustness and speed up fine-tuning [11]. During this enhancement, the weights of the original model are maintained unchanged. Typically, the original model comprises specific units or modules, such as a ResNet Block or MobileViT Block. To generalize, we will refer to these blocks as frozen operations and denote them as $OP(x)$. The parallel method of connecting an adapter to the frozen blocks of the model is the most convenient and versatile approach (Figure 2a). In this case, to ensure the properties of resilience, it is proposed to use three consecutive blocks of adapters at once, two of which are tuned during meta-training [10]. To balance between different modules, we introduce a channel-wise scaling.




a b

**Figure 2:** Design of Parallel Adapter. (a) Parallel correction scheme for the frozen block; (b) Model architecture of adapter or meta-adapter

The adapter designs shown in Figure 2b utilize convolutional layers. The convolutional adapter shown in Figure 2b can adjust channel compression by a factor of 1, 2, 4, or 8 using the γ hyperparameter.

The base model is trained on the original dataset $D_{base} = \{D_{base}^{tr}; D_{base}^{val}\}$ to perform the main task under normal conditions. Resilience optimization requires generating a set of synthetic

disturbance implementations $\{\tau_i | \ i = \overline{1,N}\}$ [2]. As disturbances $\tau_i$ can be considered adversarial attacks, fault injection, or switching to a new task. In addition, it is necessary to provide datasets $D = \{D_k^{tr}; D_k^{val} | k = \overline{1,K}\}$, that solve other problems for $K$ few-shot learning tasks, where fine-tuning data $D_k^{tr}$ is used in the fine-tuning stage and validation set $D_k^{val}$ is used in the meta-update stage. There is given a set of parameters $\theta$, $\phi$, $\omega$ and $W$, where $\theta$ are parameters of a pretrained and frozen base AI model, $\phi$ and $\omega$ are adaptation parameters of AI model backbone, and $W$ are task specified head parameters. Head weights $W_{base}$ for the main task are pre-trained on the data $D_{base}$. To simplify the description of the problem, we denote the set of all parameters as $\Xi =< \theta, \phi, \omega, W >$. Then the meta-learning process for direct maximization of the expected resilience indicator can be described by the formula [2, 17]

$$\Xi^* = \underset{\Xi}{argmax} \underset{\tau_i \sim p(\tau)}{E} \left[R_{\tau_i}\left(U(\Xi, D)\right)\right] = \underset{\Xi}{argmax} F(\Xi), \quad (1)$$

where $U$ is an operator that combines disturbance generation and adaptation in T steps, which maps the current state of $\phi$ to the new state of $\phi$;

$R_{\tau_i}$ is a function that calculates the value of the resilience indicator on test sample $D_{\tau_i}^{val}$ for $\tau_i$ disturbance implementation over model parameters $\omega$ during its adaptation by formula

$$R_{\tau_i} = \frac{1}{P_0 T} \sum_{t=1}^{T} P_{\tau_i}\left(\theta, \omega, \phi_t, W_t, D_{\tau_i}^{val}\right), \quad (2)$$

where $P_{\tau_i}$ is a performance metric for current state of model parameters and evaluation data.

In the implementation of the operator U, it is proposed to use the SGD stochastic gradient descent algorithm with T steps. The results of adaptation according to the SGD algorithm are proposed to be used for meta-updating the gradient in the outer loop. The metagradient is estimated using a Gaussian-smoothed version of the outer loop objective and is calculated according to the formula [2, 13]

$$\nabla \underset{g \sim N(0,I)}{E} [F(\Xi + \sigma g)] = \frac{1}{2\sigma} E[R(\Xi + \sigma g) - R(\Xi - \sigma g)]. \quad (3)$$

A perturbation vector $g$ is generated for the meta-optimized parameters at the beginning of each meta-optimization iteration. Thus, the pseudocode of the proposed resilient-aware meta-learning algorithm is shown in Figure 3.

As shown in Figure 3, one type of destructive influence is used within one step of meta-adaptation. Each step of the meta-adaptation process starts with a random selection of the disturbance type. Then, $n$ implementations of the disturbance are generated, followed by a nested loop of adaptation to each disturbance. Mixing different disturbances at once might not be effective. For example, following the injection of faults into the weights of the neural network, we would obtain a model that is no longer relevant for conducting adversarial attacks.

The $Adv\_perturbation()$ function is used to generate adversarial samples. Adversarial training of differentiated models can be performed using FGSM attacks or other white-box attacks [15]. However, to test models, it is proposed to use attacks based on the covariance matrix adaptation evolution strategy (CMA-ES) algorithm, which are more universal for any model [18]. The amplitude of the perturbation is limited by the $L_\infty$-norm or $L_0$-norm. If the image is normalized by dividing the pixel brightness by 255, then the specified perturbation amplitude is also divided by 255.

The introduction of faults is carried out through the $Fault\_injection()$ function [19]. It is recommended to select the fault type that poses the greatest difficulty for absorption, which entails generating an inversion of a randomly chosen bit (bit-flip injection) within the model's weight. For models that exhibit differentiability, it is advisable to pass the test dataset through the network and compute the gradients, which can subsequently be sorted according to their absolute values. In the top-k weights exhibiting the highest gradient magnitudes, a single bit is inverted at a random position. The proportion of weights subjected to the inversion of a random bit can be denoted as the fault rate.

Changing tasks can be considered as a way to mimic the concept drift and out of distribution data. The selection of other tasks can be done either by randomizing the domain of the base task or by randomizing tasks of the same domain, or in combination.

Augmented training data can be used to improve calibration on in-distribution data. Data from other datasets with semantically different annotations can be used to generate out-of-

distribution data. Also, for experimental research, Soft Brownian Offset with an autoencoder can be used to generate out-of-distribution data [20].

**Require:** Distribution over disturbances $p(\tau)$; Step size hyperparameters $\alpha$, $\beta$;
Precision parameter $\sigma$; Number of adaptation steps $T$.

Pretrain $\phi$, $\omega$ on original data $D_{base}$
While not done do:
  Select type of disturbance from set { fault injection, evasion adversarial attack, task change}
  Sample disturbance implementations $\tau_i \sim p(\tau)$, $i = \overline{1,n}$
  Sample perturbation vectors $g_{\phi,\tau_i} \sim N(0,I)$, $i = \overline{1,n}$; $g_{\omega,\tau_i} \sim N(0,I)$, $i = \overline{1,n}$
  For i=1,2,…,n do:
    Clone the current parameters: $\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i}, \hat{\phi}_{\tau_i}, \hat{W}_{\tau_i} \leftarrow copy(\theta, \omega, \phi, W_{base})$
    $\hat{\phi}_{\tau_i+} \leftarrow \hat{\phi}_{\tau_i} + \sigma g_{\phi,\tau_i}$; $\hat{\phi}_{\tau_i-} \leftarrow \hat{\phi}_{\tau_i} - \sigma g_{\phi,\tau_i}$
    $\hat{\omega}_{\tau_i+} \leftarrow \hat{\omega}_{\tau_i} + \sigma g_{\omega,\tau_i}$; $\hat{\omega}_{\tau_i-} \leftarrow \hat{\omega}_{\tau_i} - \sigma g_{\omega,\tau_i}$
    If disturbance type is a task change:
      Sample the training and validation data $D_{\tau_i}^{tr}$, $D_{\tau_i}^{val}$ from new task
    else:
      Sample the training and validation data $D_{\tau_i}^{tr}$, $D_{\tau_i}^{val}$ from $D_{base}$
    If disturbance type is a fault injection:
      $\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i+}, \hat{\phi}_{\tau_i-} \leftarrow Fault\_injection(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i+}, \hat{\phi}_{\tau_i-})$
    If disturbance type is an evasion adversarial attack:
      $D_{\tau_i}^{tr}, D_{\tau_i}^{val} \leftarrow Adversarial\_perturbation(D_{\tau_i}^{tr}, D_{\tau_i}^{val})$
    $\{\hat{\phi}_{\tau_i+,t} | t = \overline{1,T}\} \leftarrow SGD_{\phi,W}(L_{\tau_i}(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\phi}_{\tau_i+}, \hat{W}_{\tau_i}, D_{\tau_i}^{tr}),\ T,\ \alpha)$
    $\{\hat{\phi}_{\tau_i-,t} | t = \overline{1,T}\} \leftarrow SGD_{\phi,W}(L_{\tau_i}(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i-}, \hat{W}_{\tau_i}, D_{\tau_i}^{tr}),\ T,\ \alpha)$
    $R_{+,\tau_i} \leftarrow R(\{P_t(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i+}, \hat{\phi}_{\tau_i+,t},\ D_{\tau_i}^{val})\})$
    $R_{-,\tau_i} \leftarrow R(\{P_t(\hat{\theta}_{\tau_i}, \hat{\omega}_{\tau_i-}, \hat{\phi}_{\tau_i-,t},\ D_{\tau_i}^{val})\})$
    $R_{\tau_i} \leftarrow \frac{1}{2}(R_{+,\tau_i} - R_{-,\tau_i})$
  $\omega \leftarrow \omega + \beta \frac{1}{\sigma n} \sum_{i=1}^{n} R_{\tau_i} g_{\omega,\tau_i}$
  $\phi \leftarrow \phi + \beta \frac{1}{\sigma n} \sum_{i=1}^{n} R_{\tau_i} g_{\phi,\tau_i}$

**Figure 3:** Pseudocode of model-agnostic meta-learning with evolution strategies for AI-system resilience optimization

The post-hoc confidence calibration algorithm necessitates the integration of certain supplementary components to the frozen model, which are tuned on the calibration data to minimize the discrepancy between the predicted confidence and the actual probability. Calibration enhancements for classification models encompass techniques such as Isotonic Regression, Histogram Binning, Bayesian neural networks, etc [21].

Despite the existence of preparatory stages in the form of resilience optimization and confidence calibration, unexpected errors in input or model weight, unexpected changes in the domain or other distributional shifts can always occur. A promising approach to mitigate such disturbances is to use the ideas and methods of Test-Time Augmentation and Test-Time Adaptation [12]. It is proposed to calculate the entropy of the marginal probability at the model output for its tuning over a certain number of iterations for low confidence predictions by analogy with scientific research [13]. But instead of tuning the entire model, it is proposed to tune only the adapters. The loss function for test-time adaptation to the input exemplar $x$ which may belong to a certain category from the set $Y$, will be

$$l(\theta, x) \approx H(\bar{p}_\theta(\cdot\,|x)) = \sum_{y \in Y} \bar{p}_\theta(y|x) log\, \bar{p}_\theta(y|x), \quad (4)$$

where $\bar{p}_\theta(y|x)$ is an estimate of the marginal probability at the model output calculated by the formula

$$\bar{p}_\theta(y|x) = \frac{1}{B}\sum_{i=1}^{B} p_\theta(y|\tilde{x}_i), \quad (5)$$

where $\tilde{x}_i$ is the $i$-th augmented version of the input exemplar.

If, after tuning, the entropy at the model output does not exceed the threshold value, it is necessary to switch to the graceful degradation mode. The easiest way to use graceful degradation is to hand over control to a human to fine-tune the system on labeled samples.

Active learning is part of the feedback loop in our MLOps diagram. Unlike traditional MLOps, it is not the base model that is fine-tuned, but the adapters and meta-adapters. Re-training of the base model and resilience optimization can be performed when a sufficient amount of new labeled data is accumulated.

Test-time adaptation can be performed autonomously in Online Loop, and its results can be stored in the local storage. Active Learning requests can be processed asynchronously or synchronously depending on the available service resources.

## 4. Experiments

Experimental research is performed using the CIFAR-10 and CIFAR-100 datasets that contain annotated images [22]. The CIFAR-10 dataset contains 10 classes and will be considered as the main task dataset. The CIFAR-100 dataset contains 100 classes and therefore can be used as a source of data describing additional tasks for few-shot learning in the meta-learning process. The few-shot learning task has 10 classes that are randomly selected from the set of available classes ($n_{way} = 10$). It is suggested to use T=40 iterations to adapt to disturbances, each iteration processing a mini-batch of 4 images (mini_batch_size=4). It is proposed to use 16 images for each class ($k_{\text{shot}}$=16) to balance the data during adaptation. The learning rate of the inner loop and the outer loop of the resilient-aware meta-learning algorithm is alpha=0.001 and beta=0.0001, respectively. The maximum number of iterations of the outer loop of the meta-learning algorithm is 300. However, the meta-learning can be stopped earlier if the average integral resilience criterion does not increase for 10 consecutive iterations. Experiments used one of the modern architectures of visual stranformers Mobile ViT, which is popular in machine vision tasks [23]. Adapters and meta-adapters are connected in parallel to each Mobile ViT Block. The following parameters are used in the Test-Time Adaptation algorithm: learning rate is 0.001; optimizer is SGD.

The effect of each additional resilience-aware MLOps stage on the accuracy and speed of accuracy recovery is investigated by analyzing several MLOps configurations:

- Config 0 is a traditional MLOps with a fine-tuning stage based on active feedback loop data;;
- Config 1 is an improvement of Config 0 by adding a fault tolerance optimization stage;
- Config 2 is an improvement of Config 1 by adding a stage of forecast uncertainty calibration;
- Config 3 – upgraded Config 2 with Test-Time Adaptation stage.

CIFAR-10 datasets contain training, validation, and test subsamples. To simplify the experiment and analyze the results, we will divide it into 4 parts. Each part of the test data is needed to simulate a part of the AI model's life cycle. Let's consider 4 consecutive parts of the life cycle:

- Test 0 – training the AI model on the training dataset and testing the model on the first part of the test dataset, followed by selecting 10% of the test data points with the highest uncertainty;
- Test 1 – fine-tuning the AI model on the selected test data points from the previous test and testing the model on the second part of the test dataset under the disturbance, followed by selecting 10% of the test data points with the highest uncertainty;
- Test 2 – fine-tuning the AI model on selected data points from the previous test and testing the model on the third part of the test dataset under the disturbance, followed by selecting the 10% of test data points with the highest uncertainty;
- Test 3 – fine-tuning the AI model on selected test data points from the previous test and testing the model on the fourth part of the test dataset under increased disturbance intensity.

The graceful degradation mechanism is proposed to be implemented in the simplest form by rejecting a decision if the entropy threshold is exceeded. Therefore, the control is transferred to

a human or a larger and more powerful AI model. In this case, we consider the accuracy of the model, which is calculated in two ways:

- ACC1 is the accuracy of the AI system taking into account all test examples;
- ACC2 is the accuracy that does not take into account the examples for which the decision was rejected due to high uncertainty.

Conventional MLOps reject decisions based on predictive confidence, while resource-aware MLOps reject decisions based on uncertainty.

For training adapters with meta-adapters, fault injection is carried out by selecting weights with the largest absolute gradient values. The proportion of modified weights is fault_rate = 0.1. For testing the resulting model, fault injection will be performed by random bit-flips in randomly selected weights, the proportion of which (fault_rate) are equals to 0.1 or 0.15.

The training of the tuners and meta-tuners involves generating adversarial samples using the FGSM algorithm with perturbation_level according to $L\infty$ up to 3. However, to test the resulting model against adversarial attacks, the adversarial samples are generated using the CMA-ES algorithm with perturbation_level according to $L\infty$-norm are equals to 3 or 5. The number of solution generation in the CMA-ES algorithm is set to 10 to reduce the computational cost of conducting experiments.

Instead of directly modeling different types of concept drift or novelty in the data, it is proposed to model the ability to quickly adapt to task changes, as this can be interpreted as the most difficult case of real concept drift. The preparation for the experiment involved adding adapters and meta-adapters to the network, which had been trained on the CIFAR-10 dataset. During meta-training, these adapters performed adaptations to either attacks or a 10-class classification task, which was randomly generated from a selection of the CIFAR-100 set. Subsequently, to verify the capability for rapid adaptation to a new task change, the new task was considered either as a classification task with the full set of CIFAR-100 classes. The resilience curve is constructed over 40 mini-batch fine-tunings, from which the resilience criterion (2) is calculated.

Taking into account the elements of randomization, it is proposed to use their average values when assessing the accuracy of the model. To this end, 100 implementations of a certain type of disturbance are generated and applied to the same model or data.

Uncertainty calibration will be performed on a dataset containing augmented test samples and out-of-distribution samples generated by Soft Brownian Offset Sampling. 300 images per class are generated for in-distribution test set to calibrate the uncertainty. The total number of out-of-distribution images is the same as the in-distribution calibration set. In this case, the Soft Brownian Offset Sampling algorithm is used with the following parameter values: minimum distance to in-distribution data is equal to 25; offset distance is equal to 20; softness is equal to 0. Bayesian Binning into Quantiles with 10 bins was chosen as the calibration algorithm.

## 5. Results

Table 1 shows the average values of accuracy (ACC1) and accuracy excluding rejected solutions (ACC2) at different life cycle successive stages of the MobileViT model with add-ons under fault injection for each MLOps configuration. The average accuracies in Table 1 are calculated based on 100 repetitions of the experiments to reduce the effect of randomization.

Experimental data confirm the increase in fault tolerance for configuration 1 (with resilience optimization) compared to configuration 0 and configuration 2 (with uncertainty calibration) compared to configuration 1. The dynamics of accuracy growth during adaptation (Tes 1-Test 2) is higher for Config 1 and Config 2, and Config 3 is characterized by the highest accuracy values. In the last two configurations, even an increase in the fraction of damaged tensors does not lead to a significant drop in accuracy compared to the previous configurations. Also, when comparing ACC2 with ACC1, it is noticeable that ACC2 is always larger than ACC1. Config 3 ensures recovery and improved accuracy even as fault injection intensity increases. Note that the averaged values of ACC1 and ACC2 for the MobileViT-based model on Test 0-Test 3 test data with the corresponding fault injection rate without fine-tuning on 10% of human-labeled examples are 0.811 and 0.878, respectively. It proves the importance of using an active feedback loop for

adaptation. For the average accuracy values in Table 1, the margin of error does not exceed 1% at a 95% confidence level.

**Table 1**
**Accuracy of the MobileViT-based model under the influence of fault injection during the life cycle depending on the MLOps configuration**

| MLOps | Test 0 | | Test 1 | | Test 2 | | Test 3 | |
|---|---|---|---|---|---|---|---|---|
| configuration | ACC1 | ACC2 | ACC1 | ACC2 | ACC1 | ACC2 | ACC1 | ACC2 |
| Config 0 | 0.925 | 0.929 | 0.802 | 0.855 | 0.837 | 0.887 | 0.829 | 0.881 |
| Config 1 | 0.929 | 0.933 | 0.852 | 0.870 | 0.891 | 0.922 | 0.889 | 0.920 |
| Config 2 | 0.938 | 0.941 | 0.860 | 0.890 | 0.917 | 0.922 | 0.903 | 0.918 |
| Config 3 | 0.938 | 0.947 | 0.920 | 0.925 | 0.930 | 0.943 | 0.929 | 0.941 |

Table 2 shows the average values of accuracy (ACC1) and accuracy excluding rejected solutions (ACC2) at different life cycle successive stages of the MobileViT model with add-ons under adversarial evasion attacks for each MLOps configuration. The average accuracies in Table 1 are calculated based on 100 repetitions of the experiments to reduce the effect of randomization.

**Table 2**
**Accuracy values of the MobileViT-based model under adversarial attack during the life cycle depending on the MLOps configuration**

| MLOps | Test 0 | | Test 1 | | Test 2 | | Test 3 | |
|---|---|---|---|---|---|---|---|---|
| configuration | ACC1 | ACC2 | ACC1 | ACC2 | ACC1 | ACC2 | ACC1 | ACC2 |
| Config 0 | 0.925 | 0.929 | 0.720 | 0.775 | 0.787 | 0.817 | 0.819 | 0.821 |
| Config 1 | 0.929 | 0.933 | 0.782 | 0.820 | 0.891 | 0.922 | 0.889 | 0.920 |
| Config 2 | 0.938 | 0.941 | 0.805 | 0.830 | 0.917 | 0.922 | 0.903 | 0.918 |
| Config 3 | 0.938 | 0.947 | 0.850 | 0.915 | 0.890 | 0.923 | 0.923 | 0.929 |

Experimental data confirm the increase in robustness for configuration 1 (with resilience optimization) compared to configuration 0 and configuration 2 (with uncertainty calibration) compared to configuration 1. The dynamics of accuracy growth during adaptation (Tes 1-Test 2) is higher for Config 1 and Config 2, and Config 3 is characterized by the highest accuracy values. Traditional MLOps (config 0) also showed the ability to quick adaptation during fine-tuning (results of Test 1 and Test 2), but it was not successful in performance recovery. Config 1 - Config 3 show a noticeable recovery in accuracy. Increasing the magnitude of the perturbation (test 3) leads to a decrease in accuracy in all configurations, while config 1 - config 3 demonstrate greater resilience compared to configuration 0. Config 3 also provides recovery and improved accuracy even as adversarial attack intensity increases. Note that the averaged values of ACC1 and ACC2 on perturbed test data from Test 0-Test 3 stages without fine-tuning on 10% of human-labeled examples are 0.791 and 0.802, respectively. It also proves the importance of using an active feedback loop for adaptation. For the average accuracy values in Table 2, the margin of error does not exceed 1.2% at a 95% confidence level.

To evaluate the robustness and speed of adaptation of a pre-configured AI system to concept drift, it is proposed to calculate the integral resilience criterion (2) in fine-tuning mode (few-shot learning) on 10 class subset of CIFAR-100 set (Table 3).

**Table 3**
**The value of the integral resilience criterion (2) to the change of the medical image analysis task depending on the MLOps configuration**

| MLOps configuration | $\bar{R}$ |
|---|---|
| Config 0 | 0.751 |
| Config 1 | 0.830 |

Analysis of Table 3 shows that adding a resilience optimization stage to MLOps increases resilience to concept drift, i.e., robustness and speed of adaptation. For the average accuracy values in Table 3, the margin of error does not exceed 1.1% at a 95% confidence level.

According to Table 1, the resilience optimization increased the model's robustness to fault injections (inversion of one randomly selected bit in each of 10% of randomly selected weight tensors) by 5% on average. After tuning by 10% in the first part of the test data, a 6% increase in robustness to fault injections is demonstrated on the next part of the test data, even with an increase in the intensity of fault injections (15% of randomly selected weight tensors are damaged) compared to the configuration without resilience optimization. Similarly, according to Table 2, resilience optimization increases the model's robustness to adversarial attacks (maximum amplitude of 3 according to $L_\infty$-norm) by 7%. After tuning on 10% of the first part of the test data, a 7.1% increase in robustness is demonstrated even after increasing the disturbance intensity (maximum amplitude of 5 according to $L_\infty$-norm) compared to a configuration without using resiliency optimization. The results of Table 3 show that the use of resilience optimization increases the integral resilience indicator during adaptation to task changes by 10.5% compared to the configuration without resilience optimization.

The analysis of Table 1 and Table 2 shows that the application of Post-hoc Uncertainty Calibration makes it possible to further improve the model's accuracy under the influence of fault injection by an average of 4.4%, and to improve the model's accuracy under the influence of adversarial attack by an average of 1.3%. The analysis of Table 1 shows that The Test-Time Adaptation improved the model's accuracy under the influence of fault injections by an average of 6.9%. Even with an increase in the intensity of fault injections, the obtained classification accuracy using Test-Time Adaptation is approximately equal to the accuracy of the model under normal conditions without additional add-ons. Similarly, the analysis of Table 2 shows that The Test-Time Adaptation increased the model's accuracy under adversarial attack by an average of 4.72%. Moreover, even with an increase in the intensity of the attack, the model's accuracy is close to its accuracy without the influence of disturbances and without the use of add-ons.

## 6. Discussion

The proposed framework for resilience-aware MLOps facilitates the implementation of diverse specific solutions for its distinct stages and mechanisms. The central concept revolves around segregating the responsibilities of developers focused on crafting the foundational AI model, optimized for nominal operating conditions, and experts tasked with ensuring the intelligent system's resilience against disturbances and changes. Developers of the core AI model are typically burdened with accounting for data specifics, data collection methodologies, and the application itself to tackle the applied data analysis challenge. Addressing issues pertaining to AI resilience, encompassing security, trustworthiness, robustness, and rapid adaptation to changes, relies on specialized expertise detached from a particular application domain [4]. The primary obstacle in separating these tasks stems from the lack of universality in resilience-ensuring methods and an incomplete comprehension of the compatibility among methods that cater to different aspects of resilience and protection against diverse types of disturbances [2]. Determining the compatibility of these methods and combining them judiciously could augment flexibility and resilience in accordance with requirements and constraints.

The proposed implementation of Post-hoc Resilience Optimization represents merely one of the viable solutions that demonstrates the fundamental feasibility of segregating the development stage of a basic AI model tailored for normal conditions from the supplementary components aimed at ensuring resilience against disturbances and changes. The significance of employing the proposed Post-hoc Uncertainty Calibration stage has been experimentally substantiated. This stage enables, firstly, the detection of disturbances and, secondly, the accurate assessment and tolerance of uncertainty. The Test-Time Adaptation stage allows for real-time enhancement of robustness to various minor changes in input data or weights.

Unlike many existing MLOps methodologies, this approach adapts to changes at the level of adapters. In this case, adapters can be tuned both on labeled data during the fine-tuning stage and on unlabeled data during the Test-Time Adaptations stage. This ensures the continuity of the adaptation process regardless of the frequency of feedback. The size and architecture of the

adapters can vary depending on the architecture of the base model and available resources. The key is that the small size of the adapters allows for adaptation on constrained resources. The preparatory stage of resiliency optimization is able to configure meta-adapters in such a way that adaptation using adapters is accelerated.

# 7. Conclusion

## 7.1. Summary

The structure of resilience-aware MLOps for resource-constrained AI-systems has been proposed. The main novelty lies in the separate work on creating a basic model for normal operating conditions and work on ensuring its resilience. This is significant for the many industries, as the developer of the basic model should devote more time to comprehending applied field at hand, rather than specializing in a specific area of resilient systems. Therefore, Post-hoc Resilience Optimization, Post-hoc Predictive Uncertainty Calibration, Uncertainty Monitoring, Test-Time Adaptation and Graceful Degradation are used as additional stages of MLOps.

Resilience optimization aims to maximize robustness to disturbances and the ability to adapt quickly. Fault injection attack, adversarial evasion attack, and concept drift are considered as disturbances to the AI system. Additional add-ons in the form of adapters and meta-adapters are used to optimize the resilience of the AI system. Meta-adapters are updated based on meta-gradient calculated on results of adaptation to synthetic disturbances. Add-on for post-hoc calibration of predictive uncertainty can be tuned on in-distribution and out-of-distribution data. Calibrated confidence values at the output of the AI system make it possible to discard a part of unabsorbed disturbances to mitigate their impact. It has also been experimentally confirmed that the Test-Time Adaptation stage allows to increase the robustness to various small changes in input data or weights.

The experiments were performed on the CIFAR-10 and CIFAR-100 datasets. The use of post-hoc resilience optimization increased robustness to fault injection by 5% and robustness to adversarial attack by 7%. Moreover, tuning on 10% of the test data increased robustness to fault injection by 6% and robustness to adversarial attack by 7.1%. In addition, the use of post-hoc resilience optimization increased the integral indicator of resilience to task changes by 10.5%. Post-hoc uncertainty calibration increases the robustness to fault injection models by an average of 4.4% and the robustness to adversarial attacks by an average of 1.3%. The use of Test-Time Adaptation additionally increases robustness to Fault Injection by 6.9% and robustness to Adversarial Attack by 4.72%. Even an increase in the intensity of attacks does not lead to a noticeable decrease in accuracy.

Thus, experimentally confirmed increase of robustness and speed of adaptation for image recognition system during several intervals of the system's life cycle due to the use of resilience optimization, uncertainty calibration and Test-Time Adaptation stages.

## 7.2. Limitations

This research is illustrated through a case study of an image classification system and does not detail the application of resilience-aware MLOps to self-supervised or reinforcement learning systems. However, the overarching structure of resilience-aware MLOps is applicable to all kinds of intelligent systems. Additional limitation may be associated with attempts to generalize the information found, which could influence the completeness of the literature review.

The Graceful Degradation stage is excluded from the detailed analysis of their impact on resilience. The article focuses on the analysis of the MLOp structure with regard to resilience, as well as the peculiarities of implementing the stages of resilience optimization, calibration of predictive uncertainty, and test-time adaptation.

The specifics of implementing each phase on specific IoT, Edge, and other resource-constrained platforms were not considered. The focus was not on the type of target platform, but on identifying new stages of MLOps aimed at ensuring resilience.

### 7.3. Future research work

Future research should focus on the development new flexible adapter and meta-adapter architectures as addons for AI system resilience. Special attention should also be paid to the question of providing resilience for self-supervised and reinforcement learning systems. Another important area of research should be the investigation of methods to ensure resilience to new types of attacks on AI systems.

## Acknowledgements

The research was concluded in the Intellectual Systems Laboratory of Computer Science Department at Sumy State University with the financial support of the Ministry of Education and Science of Ukraine in the framework of state budget scientific and research work of DR No. 0124U000548 “Information Technology for Ensuring the Resilience of the Intelligent Onboard System of Small-Sized Aircraft”.

## References

[1] F. Khalid, M. A. Hanif, M. Shafique, Exploiting Vulnerabilities in Deep Neural Networks: Adversarial and Fault-Injection Attacks, in: Proceedings of the Fifth International Conference on Cyber-Technologies and Cyber-Systems, 2020, pp. 24–29. http://hdl.handle.net/20.500.12708/55602.

[2] V. Moskalenko, V. Kharchenko, Resilience-aware MLOps for AI-based medical diagnostic system, Frontiers in Public Health, 12 (2024). doi: 10.3389/fpubh.2024.1342937.

[3] M. Testi, M. Ballabio, E. Frontoni, G. Iannello, S. Moccia, P. Soda, G. Vessio, MLOps: A Taxonomy and a Methodology, IEEE Access, 10 (2022) pp. 63606–63618. doi:10.1109/access.2022.3181730.

[4] F.O. Olowononi, D.B. Rawat, C. Liu, Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS, IEEE Communications Surveys & Tutorials, 23 (1) (2021) pp. 524–552. doi:10.1109/comst.2020.3036778..

[5] J. V. Duddu, N. Rajesh Pillai, D.V. Rao, V.E. Balas, Fault tolerance of neural networks in adversarial settings, Journal of Intelligent & Fuzzy Systems, 38 (5) (2020) pp. 5897–5907. doi:10.3233/jifs-179677.

[6] S. Yang, T. Fevens, Uncertainty Quantification and Estimation in Medical Image Classification, in: Lecture Notes in Computer Science, 2021, pp. 671–683. doi:10.1007/978-3-030-86365-4_5.

[7] X. Zhang, J. Jaskolka, Conceptualizing the Secure Machine Learning Operations (SecMLOps) Paradigm, in: 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS), IEEE, 2022. doi:10.1109/qrs57517.2022.00023

[8] V. J. Reddi, A. Elium, S. Hymel, D. Tischler, D. Situnayake, C. Ward, L. Moreau, J. Plunkett, M. Kelcey, M. Baaijens, et al. Edge impulse: An MLOps platform for tiny machine learning, in: Proceedings of Machine Learning and Systems, 2023. doi:10.48550/arXiv.2212.03332.

[9] S. Leroux, P. Simoens, M. Lootus, K. Thakore, & A. Sharma. TinyMLOps: Operational Challenges for Widespread Edge AI Adoption, in: 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2022. doi:10.1109/ipdpsw55747.2022.00160

[10] V.V. Moskalenko, Model-agnostic meta-learning for resilience optimization of artificial intelligence system, Radio Electronics, Computer Science, Control, (2) (2023), 79. https://doi.org/10.15588/1607-3274-2023-2-9.

[11] W. Hou, Y. Wang, S. Gao, T. Shinozaki, Meta-Adapter: Efficient Cross-Lingual Adaptation With Meta-Learning, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021. doi:10.1109/icassp39728.2021.9414959.

[12] K. Bhardwaj, J. Diffenderfer, B. Kailkhura, M. Gokhale, Benchmarking Test-Time Unsupervised Deep Neural Network Adaptation on Edge Devices, in: 2022 IEEE International

Symposium on Performance Analysis of Systems and Software (ISPASS), IEEE, 2022. doi:10.1109/ispass55109.2022.00033.
[13] M. Zhang, S. Levine, C. Finn, MEMO: Test Time Robustness via Adaptation and Augmentation, arXiv, Version 3, 2021. https://doi.org/10.48550/ARXIV.2110.09506.
[14] X. Yang, J. Xu. Few-shot Classification with First-order Task Agnostic Meta-learning, in: 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA). doi:10.1109/cvidliccea56201.2022.9824307.
[15] Wang R., Xu K. , Liu S., Chen Pin-Yu et al. On Fast Adversarial Robustness Adaptation in Model-Agnostic Meta-Learning, in: 9th International Conference on Learning Representations, ICLR 2021. doi:10.48550/arXiv.2102.10454.
[16] M.W. Shen, Trust in AI: Interpretability is not necessary or sufficient, while black-box interaction is necessary and sufficient, arXiv, 2022. DOI:10.48550/ARXIV.2202.05302.
[17] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, J. Tan, Rapidly Adaptable Legged Robots via Evolutionary Meta-Learning, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020. doi:10.1109/iros45743.2020.9341571.
[18] S. Kotyan, D.V. Vargas, Adversarial robustness assessment: Why in evaluation both L0 and L∞ attacks are necessary, PLOS ONE, 17 (4) (2022) e0265723. doi:10.1371/journal.pone.0265723.
[19] G. Li, K. Pattabiraman, N. DeBardeleben, TensorFI: A Configurable Fault Injector for TensorFlow Applications, in: 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2018. doi:10.1109/issrew.2018.00024.
[20] P. Peng, J. Lu, T. Xie, S. Tao, H. Wang, H. Zhang, Open-Set Fault Diagnosis via Supervised Contrastive Learning With Negative Out-of-Distribution Data Augmentation, IEEE Transactions on Industrial Informatics, 19 (3) (2023) pp. 2463–2473. doi:10.1109/tii.2022.3149935.
[21] D. Karimi, A. Gholipour, Improving Calibration and Out-of-Distribution Detection in Deep Models for Medical Image Segmentation, IEEE Trans. Artif. Intell. (2022) 1. doi:10.1109/tai.2022.3159510.
[22] R. Doon, T. Kumar Rawat, S. Gautam, Cifar-10 Classification using Deep Convolutional Neural Network, in: 2018 IEEE Punecon, IEEE, 2018. https://doi.org/10.1109/punecon.2018.8745428.
[23] Liu, H. Chen, W. Zhou, Improved MobileViT: A More Efficient Light-weight Convolution and Vision Transformer Hybrid Model, in: Journal of Physics: Conference Series, Vol. 2562, Issue 1, IOP Publishing, 2023, p. 012012. https://doi.org/10.1088/1742-6596/2562/1/012012.