Exploiting Transformer Models in Three-version Image Classification Systems

Shamima Afrin
Department of Computer Science
University of Tsukuba
Tsukuba, Japan
afrin.shamima@sd.cs.tsukuba.ac.jp

Fumio Machida
Department of Computer Science
University of Tsukuba
Tsukuba, Japan
machida@cs.tsukuba.ac.jp

***Abstract*—Machine learning (ML) models are extensively employed in a wide range of real-world applications, including safety-critical ones. The reliability of ML application systems is a critical concern, particularly in situations where incorrect system outputs lead to severe consequences. This paper aims to enhance the reliability of ML-based image classification systems by exploiting a transformer-based model with non-transformer models in two distinct architectural frameworks, specifically the Majority Voting (MV) and Recovery Block (RB). Instead of relying on a single ML prediction, the proposed architectures leverage multiple ML models, executed simultaneously for the same input, to improve system output reliability. Our experimental results on image classification tasks show that three-version systems employing transformer models exhibit reliability enhancements in both MV and RB architectures. Moreover, considering performance overhead imposed by transformer models, we evaluated the response times as a performance metric of ML systems. The evaluation results show that RB architecture proves to have shorter response times than MV architecture.**

***Index Terms*—Image classification, N-version ML system, Recovery block, Reliability, Transformers**

I. Introduction

Machine learning (ML) has been used extensively in a diverse range of practical fields, including healthcare, finance, transportation, and beyond. Nevertheless, the use of ML in safety-critical systems, such as self-driving cars, poses a challenge in high reliability and safety. ML models inherently introduce uncertainty into their outputs and are highly sensitive to changes in input data. In a safety-critical ML system, incorrect outputs can lead to detrimental consequences, such as accidents caused by automated driving [1]. To ensure the reliability of ML system in safety-critical applications, several techniques are introduced, including risk analysis, model validation, redundant configurations, and ML testing. ML testing is an approach to detecting defects between existing ML models and required conditions [2]. However, the test coverage on testing data does not always guarantee the correctness of the predictions for anomaly examples [3].

To make ML systems dependable, redundant architecture provides a simple and effective solution. As an approach to improve ML system reliability, we can leverage the traditional software fault-tolerant technique known as N-version programming (NVP) [3]. In contrast to NVP, the N-version ML system can exploit the diversity of input data to obtain different prediction results. Similar to ensemble ML models [4], combining multiple prediction results can improve decision reliability. The existing studies on N-version ML system using commonly-adopted ML models such as Convolution Neural Networks (CNNs) and Deep Neural Networks (DNNs) [5]. For computer vision tasks, recently, transformer-based vision models like Vision Transformer (ViT) have evolved and surpassed the most of conventional models [6]. Transformer is originally designed for natural language processing task, then it becomes a norm of many machine learning tasks including computer vision [7]. While inclusion of transformer models into N-version ML systems can benefit the improved reliability, the combination of transformer and existing non-transformer models have not been largely investigated.

In this paper, we aim to leverage a transformer-based model within redundant architectures for ML-based image classification systems. As transformer-based models, such as ViT, employs a different structure from the existing DNN models, the redundant architecture can benefit the diversity of predictions to improve the reliability of system outputs. Our objective is to investigate the potential for combining a transformer-based model and the conventional DNNs, referred to non-transformer models in this paper, in N-version ML architectures for image classification systems. Despite the competitive prediction accuracy, transformer models often require more computing resources and potentially decreases the overall performance of the ML system. The synchronization overhead for decision making may become a concern when transformer is used in parallel with other light-weight non-transformer models.

To address the issue, we propose alternative redundant ML system architectures inspired from the traditional software fault-tolerant technique, known as Recovery Block (RB). RB is a technique that achieves fault avoidance where multiple alternative solutions are executed in a sequence until an acceptable solution is found, as determined by an adjudicator [8]. This technique used in software design to enhance reliability [9]. The proposed RB architecture leverages the benefits of the transformer's superior accuracy while minimizing the potential drawbacks associated with its computational demands. In this architecture, non-transformer models are used in the primary block and determine the output when they agree with the inference results. When the non-transformer models fall in disagreements, the architecture uses a transformer model as an RB to determine the final output. We present two different modes of RB architectures. In the first mode, the output of the RB solely relies on the output of the transformer model. This mode is akin to the original RB approach and is named *Independent RB*. Alternatively, we can consider a system memorizes the outputs from non-transformer models and make the final decision by majority voting (MV) of all the prediction results at the RB. This mode of RB architecture is referred to as *Dependent RB* since the output of RB depends also on the output of the primary block. Although Dependent RB requires the memory and additional comparison process for making the final decision, the reliability of system output potentially further improves. To evaluate the effectiveness of the proposed architectures, we conducted experiments on the CIFAR-10 and CIFAR-100 datasets using a various combination of transformer and non-transformer models. Our experiment results show that incorporating transformer models in three-version ML systems achieves a higher reliability on both CIFAR-10 and CIFAR-100 datasets than a system solely relying on a transformer model. Notably, the combination of BiT, ViT, and ResNet achieves the highest scores, reaching 0.9867 for CIFAR-10 and 0.9373 for CIFAR-100, surpassing the reliability of all other combinations. We also evaluated the response times of two architectures, that are the elapsed time from the start of the processing an input to the completion of the processing. The evaluation results show that the RB approach yields shorter response times than the MV architecture.

The rest of the paper is organized as follows. In Section II, we discuss the related work. In Section III, we introduce the background of our study. Section IV proposes the two variants of RB architectures. Section V presents the details of the experimental design. Section VI describes the experimental results. Finally, section VII gives our conclusion.

II. Related Work

N-version ML systems have been studied as a redundant architecture for improving the reliability of ML system outputs [21]. This approach leverages multiple input and diverse ML models to generate multiple predictions to determine a better output than the output from the system relying on a single model. The previous study has demonstrated that the outputs of ML modules can be diversified by using different versions of ML algorithms, neural network architectures, and perturbed input data [5]. From the theoretical perspective, the reliability models for two-version and three-version image classification ML architectures have been presented using diversity metrics [21] [10]. The theoretical results showed that the Triple-model with Triple-input architecture, which leverages both input and model diversities, has an advantage in the reliability of the three version ML systems [10]. While N-version ML systems enhance the reliability of system outputs through redundancy, they incur additional costs and processing overheads. To evaluate the performance overhead incurred multi-version ML systems, the performance of two-input MLSs has been theoretically investigated by queueing analysis [11]. However, the analysis relied on theoretical assumptions like exponentially distributed service time and Poisson job arrival, lacking empirical validation in real-world scenarios. In recent study [12], Two-input ML systems were implemented and empirically investigated their performance such as response time, throughput, and energy consumption. In contrast to these studies, this paper is the first to consider the performance of three-version systems employing transformer with non-transformer models.

The reliability and robustness of ML systems using transformer models is an open research challenge that requires further investigation. A transformer model is a type of deep learning model that processes input data in parallel, using self-attention and feed-forward neural networks. In contrast to traditional recurrent neural networks (RNNs) that process data sequentially, transformer models leverage parallel attention mechanisms. The first transformer model introduced in 2017 [6] uses a multi-head attention mechanism, which allows the model to attend to different parts of the input and output sequences simultaneously. ViT is the first transformer model for computer vision tasks proposed in [13]. In the domain of computer vision, input images are segmented into patches, each transformed into a vector and subsequently mapped to a smaller dimension using a linear operator. Many variants of ViT have been presented such as Swin Transformer [14], TimeSformer [15], and CaiT [16]. Although transformer-based models have achieved impressive accuracy on computer vision tasks, they are also reported to be fragile and vulnerable to adversarial attacks or input perturbations [17]. Various studies were conducted to improve the robustness of ViT against perturbations to inputs. The robustness of ViT models are evaluated by various measures in comparison with ResNet baselines has been conducted [18]. The robustness of ViT and convolutional neural networks (CNNs) models has been assessed by intentionally incorporating adversarial examples into the training dataset [17]. However, these studies aimed at increasing the robustness of the ML module do not guarantee complete system reliability.

A few existing studies consider the combination of a transformer model with a non-transformer model for making reliable ML systems. A recent study presented the integration of ViTs and CNNs in safety-critical systems against adversarial perturbations [19]. While ViT employs self-attention to learn relationships in images, suitable for tasks like classification and object detection, CNNs use convolution to capture spatial relationships in images, ideal for tasks like segmentation and edge detection. Another study explored CNNs, ViTs, and Data efficient image Transformers (DeiTs) for image classification, both individually and using ensemble learning to achieve state-of-the-art accuracy in specific domains like ecological datasets [20]. The experiment results show that the ensemble of DeiTs potentially achieve improved performance compared to individual models alone. Instead of ensembling multiple transformers, we focus on exploiting the complementary properties of transformer and non-transformer models as independent components. Furthermore, we explore their combined application and employ redundant configurations to achieve a reliable image classification system.

III. Background

A. Transformer

Transformer models have revolutionized the field of natural language processing (NLP). The original transformer introduces a self-attention mechanism and replaces sequential processing with parallel attention mechanisms [10]. This enables efficient learning of long-range dependencies in texts and achieved state-of-the-art results in machine translation, surpassing RNNs that were the dominant approach at the time. After the first transformer-based computer vision model was presented [6], many variants of ViT have been investigated. In this study, we employ ViT and CaiT as representative transformer-based models as explained below.

1) ViT: ViT extends the transformer architecture to the domain of computer vision, where demonstrated remarkable success in image classification tasks. Directly applying the transformer model to computer vision tasks would require attention between every pair of pixels, which is not practical due to the quadratic cost in the number of pixels. The ViT model overcomes this limitation by reshaping an image into a sequence of flattened patches of size P × P, effectively reducing the sequence input length by P^2 times. Generally, the patch size P is chosen to be 16 or 32. By leveraging the power of self-attention, ViT achieves state-of-the-art performance on various computer vision tasks, including image classification, object detection, and segmentation, surpassing the performance of traditional CNN architectures on several benchmark datasets.

2) CaiT: CaiT (Class-Attention in Image Transformers) is a type of image classification architecture that uses a novel class-attention layer and builds upon the encoder-decoder architecture [23]. This architecture with specific class-attention offers a more effective processing of the class embedding. Unlike traditional ViT architectures where self-attention layers within the encoder simultaneously process both individual image patches and a class embedding, CaiT separates these tasks explicitly into distinct processing stages. In the initial stage, dedicated self-attention layers focus solely on processing individual image patches without class embedding. Subsequently, the class attention stage employs a dedicated set of layer to refine the class embedding by extracting the content/relevant information from the processed patches. This explicit separation mitigates potential conflicts and leads to better performance compared to traditional transformer architectures as evidenced by state-of-the-art results on benchmark datasets such as ImageNet [23]. CaiT achieves these performance gains even without reassessed labels or additional training data and leveraging class attention for superior accuracy.

B. ML Architecture: Majority Voting

In this study, we focus on ML-based image classifications that receive image data as input and predicts the label of the input image. The architecture of ML system can leverage various forms of diversity to determine a better output than a single ML model. A type of three-version ML architecture using three ML models with a single input is called Triple Model Single Input (TMSI) system [21]. As shown in Fig.1, the same input image is used to predict the class labels by three models and the final decision is made by a MV. A voting decision from diversified prediction results can correct errors and avoid an undesirable decision. Following a MV basis, the system outputs incorrect results when more than two modules output errors for the same input. In this study, we focus on a three-version image classification system. The architecture is essentially a variant of triple modular redundancy (TMR) whose reliability characteristic is well known when failure probabilities of modules are independent [24].

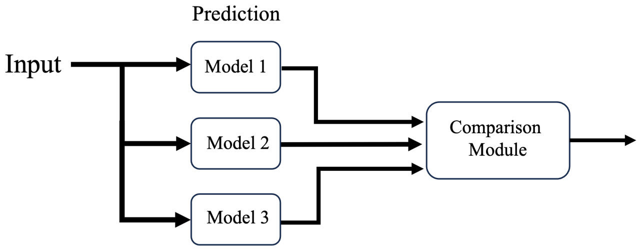

Fig. 1. Triple Model Architecture

C. Recovery block (RB)

RB is a well-known software fault tolerance technique which is a method developed by Randell [25]. Software fault tolerance is the ability for software to detect and recover from a fault that is happening or has already happened in either the software or hardware in the system in which the software is running to provide service by the specification.

In a system with RBs, the system view is broken down into fault recoverable blocks. Each block contains a primary and secondary case code along with an adjudicator. Upon first entering a unit, the primary module is run followed by the acceptance test (AT). If the module passes the AT, it is considered reliable; otherwise, it is considered faulty and then tries to roll back the state of the system and tries the secondary alternate.

One of the benefits of RB Compared with NVP is the invocation of RB is limited only when the AT fails at the primary block. This means computational resources necessary for executing the RB are conserved when the primary block performs correctly. Therefore, RB can be considered a resource efficient alternative for simple redundancy scheme like NVP.

IV. Recovery Block (RB) Architecture

N-version ML systems, while aiming to enhance reliability through redundancy by employing multiple models in parallel, can incur significant computational overhead. Such an overhead becomes particularly concerning when considering the inclusion of resource-intensive models like transformers. To address this challenge and exploit the potential of transformers for improved reliability without compromising efficiency, We propose RB architectures for ML systems, to facilitate reliable decision-making with multiple ML models. This architecture consists of a primary block and a backup block (i.e., RB) by using multiple versions of ML module and an adjudicator to decide the correct output. The idea is to execute the primary block of the algorithm and check its output with an acceptance test. If the test passes, the output is returned. Nevertheless, If the test fails, the primary version is discarded and a backup version is executed. The process is repeated until either a valid output is obtained or the system drop the output. The adjudicator is a component that implements the acceptance test and the selection of the backup component. In contrast to the N-version ML system, where all redundant models are run concurrently and each processing yields a different result, in the RB technique, all redundant models are not run concurrently. Instead, these models are executed sequentially, following the primary and recovery block. Two approaches are considered to determine the output.

A. Independent RB

Fig 2 shows the RB architecture operates within the context of an image classification system, utilizing both transformer and non-transformer models. Given the computational demands of the transformer model, it serves as a backup mechanism, independently determining the final output. We refer to this architecture option as Independent RB. This architecture uses two non-transformer models (CL1 and CL2) as the primary processing block, with the transformer model (CL3) serving as a backup recovery block. In cases where the non-transformer models solely produce conflicting labels, the final output is determined by the transformer model.

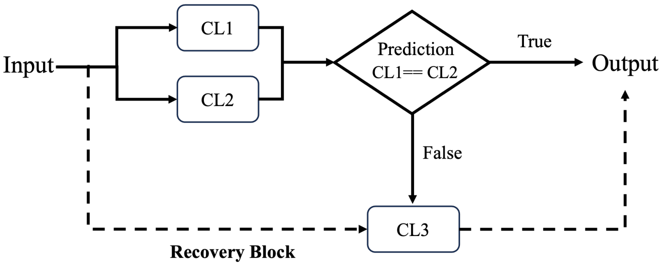

Fig. 2. Independent RB architecture

B. Dependent RB

Alternatively, we can consider a system depends on the outputs of non-transformer models and makes the final decision by MV at the recovery block. Fig. 3 shows the RB architecture referred to as Dependent RB. In this approach, the final output is determined not only from the prediction result at a recovery block but also the prediction results from the primary block. Although Dependent RB necessitates memory of prediction results and an additional comparison step for making a final decision, the approach potentially improves the reliability of the ML system. Like the MV architecture, if all three model inference results are different, the system drops the output to avoid unsafe decision.

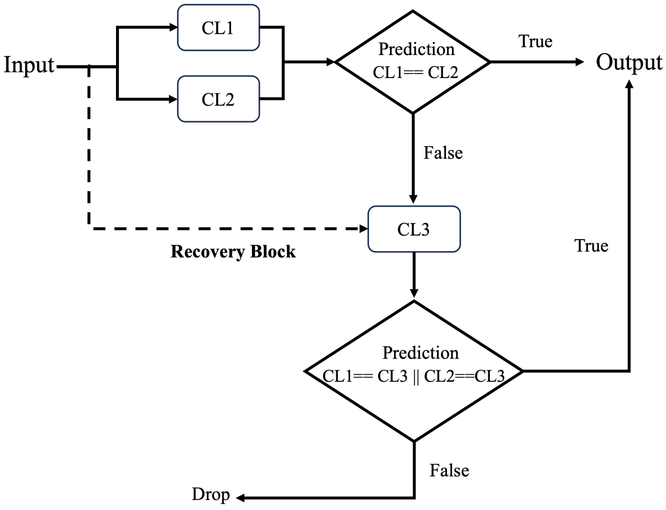

Fig. 3. Dedependent RB architecture

V. Experiment

To evaluate the effectiveness of the proposed ML system architectures we conducted performance experiments using image classification datasets.

A. Experiment configuration

In this experiment, we use five image classification models, ViT, CaiT, Residual Network (ResNet), Densely Connected Convolutional Networks (DenseNet) and Big Transfer (BiT) trained for image classification tasks on the CIFAR-10 and CIFAR-100 datasets [27]. The CIFAR-10 dataset consists of 60,000 color images of size 32x32, divided into 10 different classes, with 6,000 images per class. The CIFAR-100 dataset is similar to CIFAR-10, except that it has 100 classes containing 600 images each. Both datasets are split into 50,000 training images and 10,000 test images. For ViT, we use ViT_b16. This model uses 16x16 pixel patches and was initially pre-trained on ImageNet21K. The model is fine-tuned for CIFAR10 and CIFAR100, transitioning from the native 32x32 pixel resolution to 224x224. The ViT_b16 comprises 85 million trainable parameters. For CaiT, we use CaiT-S24 at resolution 224 with repeated augmentation originally pre-trained on ImageNet. The learning rate of the AdamW optimizer and weight decay is set to 0.00001. The S24 variation comprises 46 million trainable parameters. For non-transformer models, we use the ResNet152V2, DenseNet201 and BiT-M-R101×1 which were originally pre-trained on ImageNet and contained a total of 83, 42, and 42 million trainable parameters respectively. We evaluated the performance of three-version image classification systems in different architectures using these models.

The experiments were conducted on the computer with the following configurations.

TABLE I
PERFORMANCE COMPARISON OF SINGLE VERSION SYSTEMS ON CIFAR-10 DATASETS: RELIABILITY, AND RESPONSE TIME

Model Type	Model	Reliability	RT (ms)
Non Transformer	BiT	0.9674	4.0176
	ResNet	0.9721	4.1142
	DenseNet	0.9687	4.2511
Transformer	ViT	0.9869	7.6287
	CaiT	0.9771	11.7696

TABLE II
PERFORMANCE COMPARISON OF SINGLE VERSION SYSTEMS ON CIFAR-100 DATASETS: RELIABILITY AND RESPONSE TIME

Model Type	Model	Reliability	RT (ms)
Non Transformer	BiT	0.8799	3.8960
	ResNet	0.8296	4.1106
	DenseNet	0.844	3.9880
Transformer	ViT	0.9175	8.3130
	CaiT	0.8559	11.7647

- GPU: NVIDIA GeForce RTX 3060 x 12GB GDDR6
- Processor: 11th Gen Intel(R) Core (TM) i7-11700 @ 2.50GHz 2.50 GHz
- RAM: 64GB
- Operating System: windows 11

B. Evaluation metrics

1) System output reliability: System output reliability is defined as the probability that the output of the system is correct in terms of ground truth in the real world. We assume that the correct answer is given by real application context. For example, in a medical diagnostic system, the identification of abnormal conditions such as tumors or irregular heart rhythms is crucial. Any misidentification of these critical conditions could lead to serious medical consequences and is considered an error. As we consider the image classification system in this study, the system output reliability is evaluated by the probability of correct label predictions over the total number of input images.

2) Task Drop Ratio: The drop ratio is calculated by dividing the number of dropped outputs by the total number of sample inputs. In N-version ML system with MV and Dependent RB architectures, a requested task is dropped when the inference results do not reach a consensus by majority voting. Since a higher task drop ratio decreases the system throughput, smaller task drop ratio is better.

3) Response Time (RT): The RT is the time between the start of processing the input within the model and the completion of processing, including the generation of the corresponding output for each individual input in the test dataset. A shorter RT can benefit user experience, especially in applications like image classification where immediate results are preferred. Moreover, a system with a shorter RT can process more inputs in a given time period, thereby increasing the overall efficiency of the system. This becomes crucial in scenarios such as autonomous driving, where real-time decision-making is vital. RT depends on several factors. The complexity of the model is a significant factor as more complex models, such as transformer models, may consume more resources and take a longer time to process inputs.

C. Four-version architecture

In order to investigate the impact of the number of versions, we also consider the evaluation of four-version architectures that use both transformer and non-transformer models. The four-version architecture also follow three decision models: MV, Independent RB and Dependent RB. In the MV architecture, four models are executed in parallel and the final output is determined by the majority voting. If the predictions from more than two models out of the four concur, their agreement is designated as the final output. This approach prioritizes agreement among the models, potentially enhancing reliability in scenarios where individual models might exhibit errors. In the RB architectures, the primary module employs three non-transformer models and the RB uses a transformer model. If all three non-transformer models predict the same class, their consensus is chosen as the final output. If no such unanimity exists, the transformer model in the RB is used. Similar to three-version architectures, Independent RB solely relies on the output of the transformer model, while Dependent RB applies MV of all prediction results at the RB.

VI. EXPERIMENTAL RESULT

This section shows the experiment results on three-version image classification systems in different architectures. For the comparative purpose, we also evaluated the single version system which relies on a single classifier model and four-version systems. The reliability, the task drop ratio, and the RT are evaluated on the test datasets for CIFAR-10 and CIFAR-100. A total of 10,000 test samples were used from each dataset in the evaluation. For RT, we take the average of response times for the input samples leading to the system outputs. The dropped samples are not counted in the calculation of the average RT.

A. Single version architecture

Tables I and II illustrate a trade-off between reliability and RT in single-version deep learning architectures for image classification on the CIFAR-10 and CIFAR-100, respectively. In the single-version systems, reliability is the same as the accuracy of the model on the test datasets because the system output solely relies on the model output. Transformer models (i.e., ViT and CaiT) consistently demonstrate superior reliability compared to non-transformer models (BiT, ResNet, DenseNet) in both datasets. ViT achieves the highest reliability in both datasets (0.9869 for CIFAR-10 and 0.9175 for CIFAR-100), underlining the transformer-based model's suitability for tasks where precise and reliable predictions are crucial. Non-transformer models exhibit varying reliability performance. While BiT demonstrates good reliability on both datasets (0.9674 in CIFAR-10 and 0.8799 in CIFAR-100), some models like BiT and ResNet show a significant drop in reliability within the CIFAR-100 dataset (0.9721 in CIFAR-10 versus

TABLE III
Performance Comparison of Three-version systems on CIFAR-10 datasets: Reliability, Drop Ratio, and Response Time

Transformer	Non-transformer	Architecture	Reliability	Drop ratio	Response (ms)
ViT	BiT,ResNet	MV	0.9867	0.30%	21.4032
		Independent RB	0.9854	–	12.9570
		Dependent RB	0.9867	0.30%	13.0409
	DenseNet,ResNet	MV	0.9852	0.37%	18.4293
		Independent RB	0.9839	–	9.7578
		Dependent RB	0.9852	0.37%	9.9268
	DenseNet,BiT	MV	0.9856	0.30%	20.2324
		Independent RB	0.9845	–	10.4539
		Dependent RB	0.9856	0.30%	12.4924
CaiT	BiT,ResNet	MV	0.9829	0.37%	24.2560
		Independent RB	0.9808	–	12.8259
		Dependent RB	0.9829	0.37%	12.7785
	DenseNet,ResNet	MV	0.9820	0.35%	21.5277
		Independent RB	0.9800	–	10.0118
		Dependent RB	0.9820	0.35%	9.9741
	DenseNet,BiT	MV	0.9826	0.37%	19.9956
		Independent RB	0.9800	–	10.4614
		Dependent RB	0.9826	0.37%	10.4361

TABLE IV
Performance Comparison of Three-version systems on CIFAR-100 datasets: Reliability, Drop Ratio, and Response Time

Transformer	Non-transformer	Architecture	Reliability	Drop ratio	Response (ms)
ViT	BiT,ResNet	MV	0.9336	3.87%	19.1933
		Independent RB	0.9165	–	11.2869
		Dependent RB	0.9336	3.87%	11.2898
	DenseNet,BiT	MV	0.9333	3.57%	18.2489
		Independent RB	0.9177	–	10.7083
		Dependent RB	0.9333	3.57%	10.7308
	DenseNet,ResNet	MV	0.9169	4.37%	19.0103
		Independent RB	0.9029	–	11.5010
		Dependent RB	0.9169	4.37%	11.3458
CaiT	BiT,ResNet	MV	0.9162	4.50%	22.0928
		Independent RB	0.8859	–	11.7372
		Dependent RB	0.9162	4.50%	11.6875
	DenseNet,BiT	MV	0.9152	4.02%	21.1232
		Independent RB	0.8881	–	10.8585
		Dependent RB	0.9152	4.02%	10.9293
	DenseNet,ResNet	MV	0.9022	4.60%	22.3474
		Independent RB	0.8754	–	11.2746
		Dependent RB	0.9022	4.60%	11.1437

TABLE V
Performance Comparison of Four-version systems on CIFAR-10 datasets: Reliability, Drop Ratio, and Response Time

Transformer	Non-transformer	Architecture	Reliability	Drop ratio	Response (ms)
ViT	BiT,ResNet,DenseNet	MV	0.9918	1.81%	25.2376
		Independent RB	0.9865	–	17.2348
		Dependent RB	0.9918	1.81%	17.2917
CaiT	BiT,ResNet,DenseNet	MV	0.9899	1.95%	30.0104
		Independent RB	0.9805	–	17.8531
		Dependent RB	0.9899	1.95%	17.8993

TABLE VI
Performance Comparison of Four-version systems on CIFAR-100 datasets: Reliability, Drop Ratio, and Response Time

Transformer	Non-transformer	Architecture	Reliability	Drop ratio	Response (ms)
ViT	BiT,ResNet,DenseNet	MV	0.9549	10.99%	23.5282
		Independent RB	0.9199	–	16.0732
		Dependent RB	0.9549	10.99%	15.9308
CaiT	BiT,ResNet,DenseNet	MV	0.9440	11.30%	21.1340
		Independent RB	0.8819	–	14.2206
		Dependent RB	0.9440	11.30%	14.0065

0.8296 in CIFAR-100). In terms of RT, transformer models incurred significantly longer RTs compared to non-transformer models when processing a single input. CaiT consistently exhibited the longest RT across both datasets, followed by ViT. Conversely, non-transformer models offered faster RTs, with BiT consistently demonstrating the shortest RT in both datasets, followed by other non-transformer models.

B. Three-version architectures

Tables III and IV compare the performance of three-version systems in different architectures on the CIFAR-10 and CIFAR-100 datasets, respectively. These systems combine different pairs of transformer and non-transformer models along with three different architectures: MV, Independent RB, and Dependent RB. The analysis of reliability shows how the choice of model combination and architecture interact and influence the outcome. For the CIFAR-10 dataset, the combination of BiT, ViT and ResNet consistently achieves the highest reliability across all architectures. Interestingly, with the BiT, ResNet and CaiT combination, the reliability remains identical to 0.9829 between MV and RB architectures despite the difference in RT. However, in the CIFAR-100 dataset, this combination exhibits lower reliability of 0.9162 across all architectures compared to the BiT, ViT and ResNet combination. Compared to the reliability of single-version architecture, three-version architectures generally achieve higher reliability on both the CIFAR-10 and CIFAR-100 datasets.

For the task drop ratio, both MV and Dependent RB architectures encounter task drops in a small fraction of cases (0.3% for CIFAR-10 and 4.6% for CIFAR-100), which are caused by disagreement of three prediction results. Note that Independent RB does not have task drops entirely in both datasets because it relies solely on the transformer model output, without comparison to non-transformer models or requiring a consensus through voting. Single-version architectures inherently absent task drops because they solely rely on the prediction from a single model, eliminating the possibility of disagreement.

For the RT, we observe that CaiT has the longest RT across both datasets and all architectures, due to deeper architecture compared to other models. Notably, RB architectures demonstrate a significant reduction in RT compared to conventional MV architectures and demonstrate the potential for improving system performance through reduced processing overhead without compromising reliability. Interestingly, there is little difference in RT between Independent RB and Dependent RB in most cases. Despite Dependent RB's inherent overhead due to memory access and an additional comparison step, the overhead does not impact the total RT significantly.

For a reliable and efficient image classification system, Dependent RB potentially achieves high reliability and a fast RT in both datasets. While MV boasts the highest reliability, its slow RT renders it impractical for real-time applications. Dependent RB, on the other hand, exhibits a good balance between reliability and RT but falls slightly short of Independent RB in both aspects.

C. Four-version architectures

While the main focus on this paper is comparison of three-version architectures, it is also an interesting question on how reliability can be further enhanced with more versions. To answer this question, we construct a four-version system, consisting of three non-transformer models with one transformer model. Tables V and VI compare the performance of these systems on the CIFAR-10 and CIFAR-100 datasets, respectively. First, when we look at the reliability, four-version systems tend to achieve higher reliability compared to the top-performing three-version configurations (presented in Table V and VI) across all architectures and datasets. For instance, the four-version system with ViT achieves the highest reliability on CIFAR-10. Four-version architectures generally display comparable or marginally superior reliability compared to their three-version architecture. The results show that inclusion of an additional diverse model has the potential to enhance the reliability of the ML system.

For the task drop ratio, four-version architecture of both MV and Dependent RB show significantly higher drop ratios compared to the three-version counterparts. Specifically, the three-version systems exhibit drop ratios in a small fraction of cases 0.3% for CIFAR-10 and 4.6% for CIFAR-100, whereas the four-version systems range between 1.81% and 1.95% for CIFAR-10 and 10.99% and 11.30% for CIFAR-100. Independent RB avoids task drops similar to that three-version architecture.

For the RT, four-version architectures consistently demonstrate longer RTs compared to their three-version counterparts in both CIFAR-10 and CIFAR-100 datasets. In the four-version architecture, the RTs are notably higher, ranging from approximately 15.93 ms to 30.01 ms for CIFAR-10 and from around 14.01 ms to 23.53 ms for CIFAR-100.

The evaluation results of four-version systems show that MV architectures generally achieve high reliability. However, this benefit comes at the cost of increased RT. In contrast, Independent RB prioritizes faster RTs but may potentially sacrifice some reliability compared to other architectures. Dependent RB strikes a balance, offering competitive response times alongside good reliability. Therefore, the optimal architecture selection depends on the specific application's priorities. In applications where absolute reliability is paramount, MV might be the preferred choice. Conversely, when faster response times are critical, Independent RB could be a better option, particularly for computationally intensive tasks. Additionally, Dependent RB emerges as a versatile alternative, well-suited for scenarios demanding a balance between efficiency and reliability.

VII. Conclusion

We proposed N-version ML system architectures combining transformer and non-transformer models to improve the system output reliability. We particularly presented a new architecture inspired by RB to address the high computational cost of transformer models. Conventional N-version ML systems often suffer from high computational costs due to the parallel execution of multiple models. Our architecture incorporates two modes: Independent RB and Dependent RB. While Independent RB directly utilizes the output from the transformer model, Dependent RB stores outputs from non-transformer models and performs a majority vote to determine the final output. The proposed architecture is evaluated by image classification on CIFAR-10 and CIFAR-100 datasets. Notably, MV and Dependent RB architectures exhibit the same level of system reliability, even though MV entails longer response times than Dependent RB across all combinations. Three-version architectures leverage the combination of three different models BiT, ViT, and ResNet through MV and RB techniques. The combinations of different models through redundant architecture can potentially enhance the reliability of image classification systems.

Future studies will explore to further improve and generalize these findings. First, we can evaluate other architectures with versions more than four involving multiple transformer models and explore the desirable balance between reliability and overheads. Second, we can evaluate the architectures on diverse tasks beyond image classification. Third, we may delve deeper into robustness against various failures. Finally, it is also an interesting challenge to explore the potential of combining the N-version ML system with other reliability-enhancing techniques to enhance system reliability.

Acknowledgment

This work was partly supported by JSPS KAKENHI Grant Number 22K17871.

References

[1] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 1-18, 2017.

[2] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: Survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 1-36, 2020.

[3] W. Wu, H. Xu, S. Zhong, M. R. Lyu, and I. King, "Deep validation: Toward detecting real-world corner cases for deep neural networks," *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 125-137, 2019.

[4] Z. Zhou, *Ensemble methods: Foundations and algorithms*. CRC press, 2012.

[5] F. Machida, "On the diversity of machine learning models for system reliability," *IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 276-27609, 2019.

[6] A. Vaswani et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] R. Bommasani et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021.

[8] H. Kopka and P. W. Daly, "Software Fault Tolerance," [Online]. Available: https://users.ece.cmu.edu/~koopman/des_s99/sw_fault_tolerance/. Accessed on February 22, 2024.

[9] T. Anderson and R. Kerr, *Recovery blocks in action: A system supporting high reliability*. Springer, 1985.

[10] Q. Wen and F. Machida, "Reliability Models and Analysis for Triple-model with Triple-input Machine Learning Systems," *IEEE Conference on Dependable and Secure Computing (DSC)*, pp. 1-8, 2022.

[11] Y. Makino, T. Phung-Duc, and F. Machida, "A queueing analysis of multi-model multi-input machine learning systems," *51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 141-149, 2021.

[12] K. Wakigami, F. Machida, and T. Phung-Duc, "Reliability and Performance Evaluation of Two-input Machine Learning Systems," *IEEE 28th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 278-286, 2023.

[13] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012-10022, 2021.

[15] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," in International Conference on Machine Learning (ICML), vol. 2, no. 3, pp. 4, 2021.

[16] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 32–42.

[17] K. Mahmood, R. Mahmood, and M. Van Dijk, "On the robustness of vision transformers to adversarial examples," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7838-7847, 2021.

[18] S. Bhojanapalli et al., "Understanding robustness of transformers for image classification," *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10231-10241, 2021.

[19] M. Filipiuk and V. Singh, "Comparing Vision Transformers and Convolutional Nets for Safety Critical Systems," in *SafeAI@ AAAI*, 2022.

[20] S. P. Kyathanahally et al., "Ensembles of data-efficient vision transformers as a new paradigm for automated classification in ecology," *Scientific Reports*, vol. 12, no. 1, p. 18590, 2022, Nature Publishing Group UK London.

[21] F. Machida, "N-version machine learning models for safety critical systems," *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 48-51, 2019.

[22] F. Almalik, M. Yaqub, and K. Nandakumar, "Self-ensembling vision transformer (sevit) for robust medical image classification," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 376-386, 2022.

[23] H. Touvron et al., "Going Deeper With Image Transformers," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32-42, 2021.

[24] K. S. Trivedi, Probability & Statistics with Reliability, Queuing and Computer Science Applications, John Wiley & Sons, 2008.

[25] R. S. Hanmer, *Patterns for fault tolerant software*. John Wiley & Sons, 2013.

[26] M. R. Lyu, *Software fault tolerance*. John Wiley & Sons, Inc., 1995.

[27] A. Krizhevsky and G. Hinton, "CIFAR-10 and CIFAR-100 Datasets," 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html.