Reliability Models and Analysis for Triple-model with Triple-input Machine Learning Systems

Qiang Wen and Fumio Machida
Department of Computer Science
University of Tsukuba
Tsukuba, Japan
wen.qiang@sd.cs.tsukuba.ac.jp, machida@.cs.tsukuba.ac.jp

***Abstract*—Machine learning (ML) models have been widely applied to real-world systems. However, outputs of ML models are generally uncertain and sensitive to real input data, which is a big challenge in designing highly reliable ML-based software systems. Our study aims to improve the ML system reliability through a software architecture approach inspired by N-version programming. N-version ML architectures considered in our study combine multiple input data sets with multiple versions of ML models to determine the final system output by consensus. In this paper, we focus on three-version ML architectures and propose the reliability models for analyzing the system reliability by using diversity metrics for ML models and input data sets. The proposed model allows us to compare the reliability of a triple-model with triple-input (TMTI) architecture with other variants of three-version and two-version architectures. Through the numerical analysis of the proposed models, we find that i) the reliability of TMTI architecture is higher than other three-version architectures, but interestingly ii) it is generally lower than the reliability of double model with double input system (DMDI). Furthermore, we also find that a larger variance of model diversities negatively impacts the TMTI reliability, while a larger variance of input diversity has opposed impacts.**

Keywords—reliability, diversity, machine learning, N-version programming

I. INTRODUCTION

Machine learning (ML) is becoming an important and common component of intelligent software systems in recent years. ML-based intelligent software is widely used in our daily lives, for instance, in speech and image recognition, strategy games, and robot applications. In spite of the widespread use of ML applications, ML system reliability is still a major concern in safety-critical applications such as automated driving. Outputs of ML models are inherently uncertain and sensitive to the input data. For safety-critical ML systems, any incorrect outputs from ML models may cause undesirable consequences such as traffic accidents by automated driving [2].

To improve the reliability of ML-based software systems, various software engineering techniques can be leveraged. ML testing is an approach to detect defects between existing ML models and required conditions [1]. Several recent studies exploit metamorphic relations to tackle the oracle problems encountered in ML application testing [3][4][5]. ML testing techniques effectively reduce the dormant defects of ML components that potentially cause serious consequences in operation. However, the test coverage on testing data does not always guarantee the correctness of the predictions for wild examples [20]. To ensure the safe operation of ML systems at runtime, it is necessary to incorporate additional safety mechanisms such as data validation [20], safety monitors [22], and redundant architecture [13][16][19]. Data validation requires a white box model for deep neural networks, which is not generally applicable to other types of ML models. Safety monitors can be used for more general ML models but need to be trained together with the ML model in advance and hence may lack the flexibility. Redundant architecture approaches are rather simple yet can be applied to any combination of ML models that are trained independently for the same task.

As a redundant architecture approach to improve ML system reliability, we leverage the traditional software fault-tolerant technique, commonly known as N-version programming (NVP). NVP is defined as the independent generation of N (≥ 2) functionally equivalent programs from the same initial specification [11]. The core idea of NVP is to leverage the diversity of independently generated program components. Experimental studies have shown immense benefits of using NVP for ML components [12]. In contrast to NVP, the N-version ML system can exploit the diversity of input data to further increase the diversity of predictions. Similar to ensemble ML models [21], the output decision based on multiple predictions results could improve accuracy.

In this paper, we propose a system reliability architecture made of N-version ML modules to improve the reliability of ML-based software systems. We consider an ML-based classifier and evaluate the system reliability by the probability that the system does not output errors for ground truth. In an N-version ML system, multiple ML models and sensor inputs are used to determine the final system output. The main factors of improved reliability by the N-version ML system are the exploitation of input diversity and model diversity [13]. For input diversity, it can be achieved with samples from different input data sources (e.g., different sensors). Even when input data such as an image or a voice sentence causes incorrect ML output, different input data may produce correct ML output. On the other hand, for model diversity, ML models trained by different algorithms and/or training data sets can be used simultaneously. The benefit of such diversities has been studied for two-version architecture systems [13]. Nevertheless, the reliability gained by three or more version architectures that use majority voting for deciding the final outputs is still underexplored. In this paper, we focus on three-version architecture systems and propose the reliability model for a triple-model with triple-input machine learning system (TMTI) that combines three different inputs with three different ML models. To show the advantages and disadvantages of the three-version architecture, we conduct numerical experiments on the proposed reliability model and compare the reliabilities of different three-version and two-version architectures. Our numerical results uncover that the reliability of TMTI is

higher than other three-version architectures in most ranges of model/input diversity values. Interestingly, however, when compared with a two-version architecture, the reliability of a double-model with double-input system (DMDI) is generally better than the reliability of TMTI because of the difference in the output decision logic. Any three-version architectures employing a majority voting to determine the final output need to reach a consensus, which causes a disadvantage against two-version architectures. Furthermore, we also find that a larger variance of model diversities positively impacts the TMTI reliability, while a larger variance of input diversity negatively impacts the reliability.

The remainder of the paper is organized as follows. Section II explains related work. Section III introduces some background, including the definition of N-version ML architectures. In Section IV, we propose the reliability model for three-version ML systems using diversity metrics. Section V shows the numerical analysis results. Finally, Section VI gives the conclusions and briefly explains the future work.

II. Related Work

The reliability of ML components is considered by some approaches, such as ML testing techniques and adversarial ML. Evaluation measures (e.g., code coverage and mutation testing) are adopted in ML testing scenarios to quantify the reliability of ML components [6][7][8]. Besides, ML testing by category is a direction for improving the reliability of ML-based systems. Building robust ML models against adversarial examples is also important to reduce the error probability of ML models [9][10]. These approaches all mainly focus on the ML models, while we propose a multi-version ML approach for the ML system reliability.

Many recent studies introduce multi-version ML approaches to improve ML system reliability. It is revealed that NVP improves the reliability of ML components by a significant margin [12]. The approximate reliability of DNN is used to calculate the reliability of ML components. PolygraphMR composes several convolutional neural networks (CNNs) to improve the reliability of classification tasks by reducing high confidence wrong answers [16]. NV-DNN is presented as a system containing N independently developed models and a decision procedure that aims to enhance the fault-tolerant ability of a deep learning system [19]. While these studies assume that a single input at a time, our study considers multi-input to exploit the input diversity besides multi-version ML components. Considering the performance of multi-version ML systems, queueing models have been proposed for multi-model multi-input ML systems in two-version architectures [17]. The study focuses on the throughput of simple two-version architectures in which the system can use at most two different ML models and two data sources. The reliability impacts of multi-input for N-version ML systems have been modeled and analyzed by introducing the measure of the input diversity [13]. Moreover, experiments with diverse machine learning models with perturbated input datasets are conducted on image classification tasks of MNIST data set, and Belgian Traffic Sign data set [18]. The study shows that the three-version architecture can achieve higher system reliability than a single machine learning module. Compared to the existing studies, our work first formulates the reliability of three-version ML architectures with diversity metrics and analytically shows the potential reliability improvements by TMTI architecture. In particular, we clarify that the reliability of TMTI may not be better than the reliability of DMDI under the majority voting decision scheme.

III. Background

This section introduces some backgrounds of the study. First, we introduce our target safety-critical ML system that needs high reliability of ML prediction outputs. Next, we introduce the N-version architecture and its property. As specific instances of N-version architectures, we explain the details of two-version and three-version architectures in the following sections.

A. Safety-critical ML system

As the accuracy of ML-based classification tasks has greatly improved recently, ML components are also being employed in safety-critical domains. In this paper, we consider such a safety-critical ML system that classifies the real-world input data such as images, sounds, and voices. Automated driving is a representative example of such a system as it needs to recognize the surroundings by classifying the images captured by the sensors. If the system controls the vehicle based on the error classification outputs, vehicle accidents can happen, and pedestrians' lives will be threatened. However, even using extremely-accurate ML models, it is impractical to guarantee 100% accuracy for real-world samples. We have to assume that error outputs from ML models are inevitable, but we still need to deal with the issue so that system can avoid hazardous consequences. Redundant architecture can be employed for such ML systems by using multiple ML components in parallel. Individual ML components may install different ML models and use different data sources to diversify the prediction results as shown in Fig 1. In automated driving, when recognizing the traffic sign like 'STOP' using different sensors, even though some of the ML models output errors like 'GO STRAIGHT,' a voting decision from diversified prediction results can correct the error and avoid an undesirable decision. In the N-version architecture introduced in the following section, we can benefit from this redundant scheme to improve the reliability of the system even with a single ML model by taking inputs from different data sources.

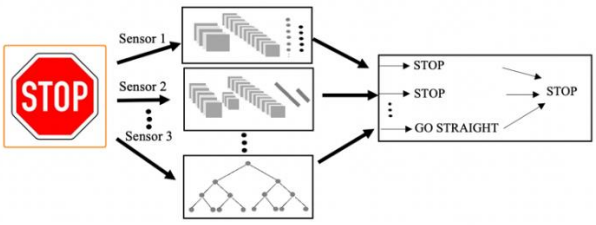

Figure 1: An example of the N-version ML system design

B. N-version ML architecture

N-version ML architecture is based on NVP and contains N (≥ 2) different versions of ML components that work for the same task in parallel. In the N-version ML architecture, we can use multiple inputs and ML models in a system. Different algorithms and training data sets are used to generate multiple versions of ML models. The diversity of ML models can mitigate the risk of erroneous system output by mutually validating individual outputs. ML components are also sensitive to input data. An ML

model may produce different results for slightly different input data (e.g., a sample image with a one-pixel change). By taking samples from different sources, we can leverage the input diversity for ML-based predictions so that the system can detect or correct the error outputs from ML models. Every ML model in N-version architecture can produce a prediction result, while the final result is determined by a specific decision rule. For a three-version architecture system, it is reasonable to employ a majority voting rule to determine the final outputs by consensus of individual ML components. Under the majority voting decision, the system can tolerate one classification error from one ML component if the other two components correctly predict the class labels. Our reliability models and analysis focus on binary classification tasks for ML components, although other types of ML tasks can also benefit from N-version architectures. As specific instances of N-version architectures, two-version and three-version ML architectures are explained next.

C. Two-version architectures

Fig. 2 shows the types of two-version architecture that may use two inputs and/or two ML models. x_1 and x_2 represent the inputs to ML models from different data sources (e.g., sensors), while m_a and m_b represent different ML models dealing with the same task. In two-version architectures, when both running models output the same error, the system results in the error output. A double-model with single-input system (DMSI) has two ML models with the same input. Any errors in one model can be detected when the other model outputs correctly. The reliability of the DMSI system is affected by model diversity. A single-model with double-input system (SMDI) has one ML model with two inputs. The input diversity is exploited in the SMDI system. Furthermore, a double-model with double-input system (DMDI) has two ML models with two different inputs. Since two ML models work with the different inputs in the DMDI system, both model diversity and input diversity affect the system reliability.

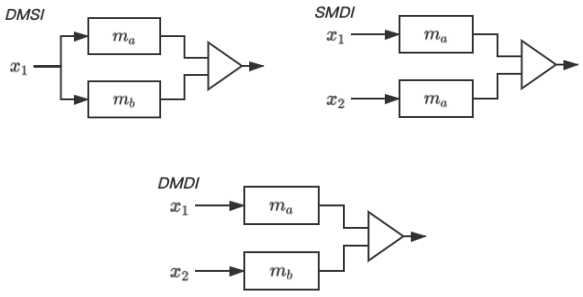

Figure 2: Two-version architecture

D. Three-version architectures

Fig. 3 shows the different three-version architectures. Systems can contain three inputs and/or three ML models. We add input x_3 and ML model m_c in three-version systems. Under the majority voting rule, the system outputs an error when more than two of the running ML models agree on an incorrect output. A triple-model with single-input system (TMSI) has three ML models with the same input. TMSI has the same structure as DMSI, but one more ML model is added. Besides, a single-model with triple-input system (SMTI) has one ML model with three inputs, which is an extension of an SMDI system. Finally, TMTI has three ML models with three inputs, which is an extension of a DMDI system. We analyze the reliability of TMTI through the diversity metrics introduced in the next section.

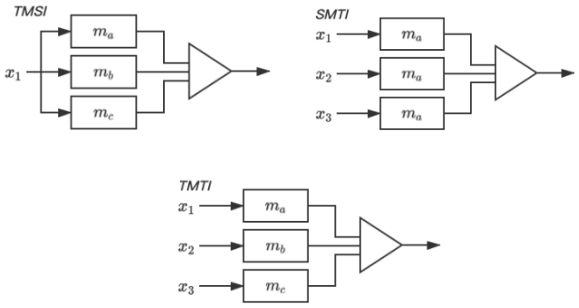

Figure 3: Three-version architecture

IV. Reliability Modeling

In this section, we present the reliability models to analyze the effectiveness of N-version ML architectures quantitatively. We extend the existing studies to propose the reliability model for three-version architectures using diversity metrics.

A. Existing reliability models for triple module systems

There are some existing triple redundancy models with a majority voting scheme. A triple module redundancy (TMR) system consists of three independent modules, each of which has a single output [14]. The failures of the modules are statistically independent, and the reliability of TMR is determined by the reliability of one module R_M by (1). Note that application of this type of redundancy cannot increase the reliability if R_M is less than 0.5.

$$R = {R_M}^3 + 3{R_M}^2(1 - R_M) = 3{R_M}^2 - 2{R_M}^3. \quad (1)$$

A reliability model for an NVP system with a majority voting decision is considered with a dependent failure parameter α that represents the similarity percentage of the input sets on which each pair of versions fail [15]. A joint probability can be formulated by using the definition of conditional probability. When the failure probability of each version is assumed to be p, the reliability of a three-version NVP model is given as follows.

$$R = 1 - [3\alpha p(1 - \alpha) + \alpha^2 p] = 1 - \alpha p(3 - 2\alpha). \quad (2)$$

In contrast to the existing studies, our focus in this paper is the system that assembles three ML modules whose outputs are dependent on each other. Thus, the reliability models (1) and (2) are not directly applicable to three-version ML architectures. To incorporate the dependency factors, we use two diversity metrics measuring the diversity of input data and ML models. The diversity metrics were first introduced in modeling two-version ML architectures [13], but the approach has never been examined for N-version architecture with N>2. Therefore, in this paper, we attempt to extend the approach to three-version ML architectures.

B. Notations and diversity measures

Table 1 summarizes the notations used in our reliability models throughout the paper. Let f_k be the probability that ML model m_k outputs errors. Let S be the total sample space of inputs and $E_k \subseteq S$ be the set of input data that leads to output error by m_k. The error probability f_k is given by

$$f_k = \frac{|E_k|}{|S|}. \quad (3)$$

Let E_i and E_j be the input sets that make ML models m_i and m_j output error; Define the intersection of errors $\alpha_{i,j} \in [0,1]$ as the ratio of the intersection over the smaller size of E_i and E_j, respectively. Here, $i, j \in N^+$. The intersection of errors represents the degree of coincidence of the inputs that caused the two models m_i, m_j to be errors. The intersection of errors is also referred to as model diversity [13]. A smaller intersection value is better as it indicates that two ML models are unlikely to reach a mutual error.

$$\alpha_{i,j} = \frac{|E_i \cap E_j|}{min\{E_i, E_j\}}. \quad (4)$$

Let x_i and x_j be the inputs to ML models from different data sources (e.g., sensors). Then, define the conjunction of errors $\beta_{k,i|j} \in [0,1]$ as the probability that m_k outputs error by x_i provided that m_k outputs error by x_j. Here, $i, j, k \in N^+$. The conjunction of errors is related to the conditional probability that m_k outputs error by x_i given the condition that m_k outputs error by x_j. The conjunction of errors is also referred to as input diversity [13]. A smaller conjunction value is better because the probability of a mutual error becomes small.

$$\beta_{k,i|j} = Pr[x_i \in E_k | x_j \in E_k]. \quad (5)$$

TABLE I. NOTATIONS USED IN THE RELIABILITY MODEL

Symbols	Descriptions
m_k	An ML model
x_i	An input data source for ML models
f_k	The probability that ML model m_k outputs errors
S	The total sample space of inputs
E_k	The set of input data that leads to output error by m_k
$\alpha_{i,j}$	The intersection of errors between m_i and m_j (model diversity)
$\beta_{k,i\|j}$	The conjunction of errors on m_k by x_j followed by x_i (input diversity)
$f_{p,q}$	The error probability of the ML architecture with p inputs and q models
$R_{p,q}$	The reliability of the ML architecture with p inputs and q models

C. Reliability of two-version architectures

The reliability of two-version architectures has been studied in [13]. It is assumed that any two-version architecture systems output errors when both ML models output errors. We briefly review the reliability models for two-version architectures below.

1) Reliability of DMSI

Let $f_{2,1}(m_1, m_2; x_1)$ be the error probability of a DMSI system using two ML models m_1 and m_2. Without loss of generality, we can assume $|E_1| \leq |E_2|$ and hence we have

$$f_{2,1}(m_1, m_2; x_1) = \frac{|E_1 \cap E_2|}{|S|} = \alpha_{1,2} \frac{\min\{|E_1|, |E_2|\}}{|S|} = \alpha_{1,2} \cdot f_1. \quad (6)$$

The reliability of a DMSI system is calculated by

$$R_{2,1}(m_1, m_2; x_1) = 1 - f_{2,1}(m_1, m_2; x_1). \quad (7)$$

The model indicates that the reliability of DMSI relies only on the intersection of errors between two ML models.

2) Reliability of SMDI

Let $f_{1,2}(m_1; x_1, x_2)$ be the error probability of an SMDI system using two inputs x_1 and x_2. Using the input diversity, we have

$$\begin{aligned} f_{1,2}(m_1; x_1, x_2) &= Pr[x_1 \in E_1, x_2 \in E_1] \\ &= Pr[x_2 \in E_1 | x_1 \in E_1] \cdot Pr[x_1 \in E_1] \\ &= \beta_{1,2|1} \cdot f_1. \end{aligned} \quad (8)$$

The reliability of SMDI system is calculated by

$$R_{1,2}(m_1; x_1, x_2) = 1 - f_{1,2}(m_1; x_1, x_2). \quad (9)$$

The model indicates that the reliability of SMDI relies only on the conjunction of errors on two inputs.

3) Reliability of DMDI

Let $f_{2,2}(m_1, m_2; x_1, x_2)$ be the failure probability of a DMDI system using two ML models m_1 and m_2 and two inputs x_1 and x_2. By assuming conditional independence, we have

$$\begin{aligned} f_{2,2}(m_1, m_2; x_1, x_2) &= Pr[x_1 \in E_1, x_2 \in E_2] \\ &= Pr[x_2 \in E_2 | x_1 \in E_1] \cdot Pr[x_1 \in E_1] \\ &= \left[\beta_{1,2|1} \cdot \alpha_{1,2} + (1 - \beta_{1,2|1}) \cdot \frac{f_2 - \alpha_{1,2} \cdot f_1}{1 - f_1}\right] \cdot f_1 \end{aligned} \quad (10)$$

The reliability of DMDI system is calculated by

$$R_{2,2}(m_1, m_2; x_1, x_2) = 1 - f_{2,2}(m_1, m_2; x_1, x_2). \quad (11)$$

As the model represents, the reliability of DMDI depends on both the intersection and the conjunction of errors.

D. Reliability of three-version architectures

The modeling methodology for the two-version ML architecture can be generalized for more versions of architectures, but existing studies have never examined the reliability models of N-version architectures with N>2. To extend reliability models for three-version architectures, we need to consider i) a majority voting rule that determines the final system output from three dependent module outputs, and ii) multiple diversity parameters among three ML models and three input sources that characterize the dependencies of the outputs. First, we show the reliability models for TMSI and SMTI systems using the intersection of errors and the conjunction of errors, respectively. Then, we exploit both diversity parameters to formulate the reliability of a TMTI system.

1) Reliability of TMSI

Let $f_{3,1}(m_1, m_2, m_3; x_1)$ be the error probability of a TMSI system output using three ML models m_1, m_2 and m_3. Since the system outputs errors when at least two modules output errors,

$$\begin{aligned} f_{3,1}(m_1, m_2, m_3; x_1) = {} & Pr[x_1 \in E_1, x_1 \in E_2, x_1 \in \overline{E_3}] + \\ & Pr[x_1 \in E_1, x_1 \in \overline{E_2}, x_1 \in E_3] + \\ & Pr[x_1 \in \overline{E_1}, x_1 \in E_2, x_1 \in E_3] + \\ & Pr[x_1 \in E_1, x_1 \in E_2, x_1 \in E_3] \\ = {} & Pr[x_1 \in E_1, x_1 \in E_2] + Pr[x_1 \in E_1, x_1 \in E_3] + \\ & Pr[x_1 \in E_2, x_1 \in E_3] - 2Pr[x_1 \in E_1, x_1 \in E_2, x_1 \in E_3]. \end{aligned}$$

For the last term, when we assume conditional independence,

$$\begin{aligned} & Pr[x_1 \in E_1, x_1 \in E_2, x_1 \in E_3] \\ & = Pr[x_1 \in E_3, x_1 \in E_2 | x_1 \in E_1] \cdot Pr[x_1 \in E_1] \\ & = Pr[x_1 \in E_3 | x_1 \in E_1] \cdot Pr[x_1 \in E_2 | x_1 \in E_1] \\ & \quad \cdot Pr[x_1 \in E_1]. \end{aligned}$$

Without loss of generality, we can assume $|E_1| \leq |E_2| \leq |E_3|$, and hence we have

$$f_{3,1}(m_1, m_2, m_3; x_1) = \alpha_{1,2} \cdot f_1 + \alpha_{1,3} \cdot f_1 + \alpha_{2,3} \cdot f_2 - 2\alpha_{1,2} \cdot \alpha_{1,3} \cdot f_1 . \quad (12)$$

Since the error probability of DMSI is given by (6), $f_{3,1}(m_1, m_2, m_3; x_1)$ can be expressed as follows:

$$f_{3,1}(m_1, m_2, m_3; x_1) = f_{2,1}(m_1, m_2; x_1) + f_{2,1}(m_1, m_3; x_1) + f_{2,1}(m_2, m_3; x_1) - 2\alpha_{1,2} \cdot \alpha_{1,3} \cdot f_1. \quad (13)$$

The reliability of a TMSI system is computed by

$$R_{3,1}(m_1, m_2, m_3; x_1) = 1 - f_{3,1}(m_1, m_2, m_3; x_1). \quad (14)$$

Similar to DMSI, the reliability of TMSI only depends on the intersections of errors among three ML models.

2) Reliability of SMTI

Let $f_{1,3}(m_1; x_1, x_2, x_3)$ be the error probability of an SMTI system using three inputs x_1, x_2 and x_3. Since the system outputs errors when at least two modules output errors,

$$\begin{aligned} f_{1,3}(m_1; x_1, x_2, x_3) &= Pr\left[x_1 \in E_1, x_2 \in E_1, x_3 \in \overline{E_1}\right] + Pr\left[x_1 \in E_1, x_2 \in \overline{E_1}, x_3 \in E_1\right] + Pr\left[x_1 \in \overline{E_1}, x_2 \in E_1, x_3 \in E_1\right] + Pr[x_1 \in E_1, x_2 \in E_1, x_3 \in E_1] \\ &= Pr[x_1 \in E_1, x_2 \in E_1] + Pr[x_1 \in E_1, x_3 \in E_1] + Pr[x_2 \in E_1, x_3 \in E_1] - 2Pr[x_1 \in E_1, x_2 \in E_1, x_3 \in E_1]. \end{aligned}$$

For the last term, when we assume conditional independence,

$$\begin{aligned} &Pr[x_1 \in E_1, x_2 \in E_1, x_3 \in E_1] \\ &= Pr[\, x_2 \in E_1, x_3 \in E_1 | x_1 \in E_1] \cdot Pr[x_1 \in E_1] \\ &= Pr[x_2 \in E_1 |\, x_1 \in E_1] \cdot Pr[x_3 \in E_1 |\, x_1 \in E_1] \cdot Pr[x_1 \in E_1]. \end{aligned}$$

Then, we have

$$f_{1,3}(m_1; x_1, x_2, x_3) = \beta_{1,2|1} \cdot f_1 + \beta_{1,3|1} \cdot f_1 + \beta_{1,3|2} \cdot f_2 - 2\beta_{1,2|1} \cdot \beta_{1,3|1} \cdot f_1. \quad (15)$$

Since the error probability of an SMDI system is given by (8) $f_{1,3}(m_1; x_1, x_2, x_3)$ can be expressed as follows:

$$f_{1,3}(m_1; x_1, x_2, x_3) = f_{1,2}(m_1; x_1, x_2) + f_{1,2}(m_1; x_1, x_3) + f_{1,2}(m_1; x_2, x_3) - 2\beta_{1,2|1} \cdot \beta_{1,3|1} \cdot f_1. \quad (16)$$

Then, the reliability of SMTI system is computed by

$$R_{1,3}(m_1; x_1, x_2, x_3) = 1 - f_{1,3}(m_1; x_1, x_2, x_3). \quad (17)$$

Similar to SMDI, the reliability of SMTI only depends on the conjunctions of errors among three input data.

3) Reliability of TMTI

Let $f_{3,3}(m_1, m_2, m_3; x_1, x_2, x_3)$ be the error probability of a TMTI system output using three ML models m_1, m_2 and m_3 and three inputs x_1, x_2 and x_3. Since the system outputs errors when at least two modules output errors,

$$\begin{aligned} f_{3,3}(m_1, m_2, m_3; x_1, x_2, x_3) &= Pr[x_1 \in E_1, x_2 \in E_2] + Pr[x_1 \in E_1, x_3 \in E_3] + Pr[x_2 \in E_2, x_3 \in E_3] - 2Pr[x_1 \in E_1, x_2 \in E_2, x_3 \in E_3]. \end{aligned}$$

For the last term, when we assume conditional independence,

$$\begin{aligned} &Pr[x_1 \in E_1, x_2 \in E_2, x_3 \in E_3] \\ &= Pr[\, x_2 \in E_2, x_3 \in E_2 | x_1 \in E_1] \cdot Pr[x_1 \in E_1] \\ &= Pr[x_2 \in E_2 |\, x_1 \in E_1] \cdot Pr[x_3 \in E_3 |\, x_1 \in E_1] \cdot Pr[x_1 \in E_1]. \end{aligned}$$

Without loss of generality, we can assume $|E_1| \leq |E_2| \leq |E_3|$. Since the error probability of DMDI is given by (10), $f_{3,3}(m_1, m_2, m_3; x_1, x_2, x_3)$ can be expressed as follows:

$$\begin{aligned} f_{3,3}(m_1, m_2, m_3) &= f_{2,2}(m_1, m_2) + f_{2,2}(m_1, m_3) + f_{2,2}(m_2, m_3) - 2f_{2,2}(m_1, m_2) \cdot f_{2,2}(m_1, m_3)/f_1 \\ &= \beta_{1,2|1} \cdot \alpha_{1,2} \cdot f_1 + (1 - \beta_{1,2|1}) \cdot (f_2 - \alpha_{1,2} \cdot f_1)/(1 - f_1) \cdot f_1 + \beta_{1,3|1} \cdot \alpha_{1,3} \cdot f_1 + (1 - \beta_{1,3|1}) \cdot (f_3 - \alpha_{1,3} \cdot f_1)/(1 - f_1) \cdot f_1 + \beta_{2,3|2} \cdot \alpha_{2,3} \cdot f_2 + (1 - \beta_{2,3|2}) \cdot (f_3 - \alpha_{2,3} \cdot f_2)/(1 - f_2) \cdot f_2 - 2[\beta_{1,2|1} \cdot \alpha_{1,2} \cdot f_1 + (1 - \beta_{1,2|1}) \cdot (f_2 - \alpha_{1,2} \cdot f_1)/(1 - f_1) \cdot f_1] \cdot [\beta_{1,3|1} \cdot \alpha_{1,3} + (1 - \beta_{1,3|1}) \cdot (f_3 - \alpha_{1,3} \cdot f_1)/(1 - f_1)]. \end{aligned} \quad (18)$$

As a result, the reliability of the TMTI system is

$$R_{3,3}(m_1, m_2, m_3; x_1, x_2, x_3) = 1 - f_{3,3}(m_1, m_2, m_3; x_1, x_2, x_3). \quad (19)$$

As the model shows, the reliability of TMTI depends both on the intersections of errors among three ML models and the conjunctions of errors among three input data.

V. Numerical Analysis

Based on the developed reliability models, we conduct numerical experiments to analyze the advantage and disadvantage of three-version architectures. Our numerical study aims to answer the following research questions.

- How does the model diversity affect the reliabilities of three-version architecture systems? (Q1)
- How does the input diversity affect the reliabilities of three-version architecture systems? (Q2)
- Do three-version architectures achieve higher reliabilities than two-version architectures? (Q3)
- How does the variance of diversity metrics impact the reliability of TMTI? (Q4)

To answer Q1 and Q2, we conduct sensitivity analyses of diversity parameter values on the reliability of different three-version architectures. The results are summarized in Section V.A and V.B, respectively. To answer Q3, we compare the reliabilities of two-version and three-version architectures under different parameter settings (symmetric and asymmetric cases). The results are discussed in Section V.C. Finally, for Q4, we focus on the reliability of TMTI and analyze the impact of variance of diversity metrics in Section V.D.

As we are interested in the impact of diversity in N-version architectures, we assume that the error probability of a single ML model is equal and is set to 0.1 (i.e., $f_1 = f_2 = f_3 = 0.1$) in the following experiments.

A. Reliability impacts of model diversity

To analyze the impact of the model diversity on the system reliability, we fix $\alpha_{1,3} = \alpha_{2,3} = 0.3$, $\beta_{1,2|1} = \beta_{1,3|1} = \beta_{1,3|2} = \beta_{2,3|2} = 0.4$, and vary the value of $\alpha_{1,2}$ in [0,1]. The reliabilities computed by our reliability models are shown in Fig. 4. As can be seen, the reliabilities of TMTI and TMSI decrease with the value of $\alpha_{1,2}$, but the reliabilities of SMTI and SMSI are not affected by this change. SMSI is the architecture with a single ML model

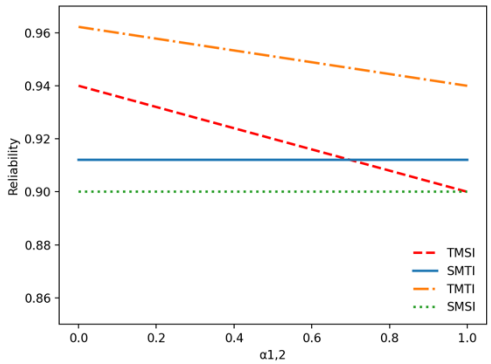

Figure 4: Reliability impacts of $\alpha_{1,2}$

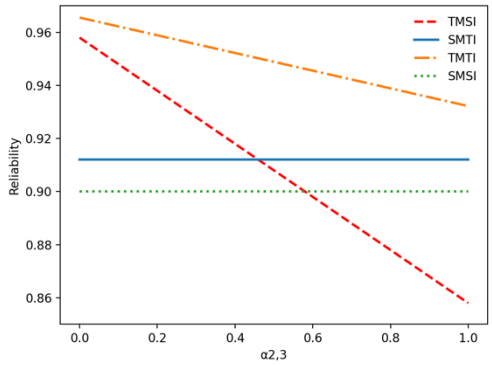

Figure 5: Reliability impacts of $\alpha_{2,3}$

with a single input, and the reliability of SMSI is a baseline to other systems. This figure clearly shows the reliability of TMTI is higher than the other three systems. The reliability of TMSI is higher than SMTI when $\alpha_{1,2}$ is smaller than 0.7. The results confirm that a smaller intersection of errors increases the model diversity leading to improved reliabilities of TMSI and TMTI systems.

Next, we fix $\alpha_{1,2} = \alpha_{1,3} = 0.3$ and $\beta_{1,2|1} = \beta_{1,3|1} = \beta_{1,3|2} = \beta_{2,3|2} = 0.4$, and vary the value of $\alpha_{2,3}$ in [0,1]. The reliabilities of different configurations are computed as shown in Fig. 5. We can observe a similar trend as observed in Fig. 4. Both reliabilities of TMTI and TMSI decrease as the value of $\alpha_{2,3}$ increases. However, Fig. 5 illustrates that the reliability of TMSI can be worse than a single model when $\alpha_{2,3}$ is larger than around 0.6. The results imply that TMSI is not a good option when the overlapping part of E_2 and E_3 increases to a certain amount.

Observation 1. The higher model diversity (i.e., the smaller value of intersection of errors) leads to the higher reliability of TMTI and TMSI.

Observation 2. The reliability of TMSI can be worse than even a single model when the intersection of errors is high.

B. Reliability impacts of input diversity

Next, we fix $\alpha_{1,2} = \alpha_{1,3} = \alpha_{2,3} = 0.3$, $\beta_{1,3|1} = \beta_{1,3|2} = \beta_{2,3|2} = 0.4$ and vary the value of $\beta_{1,2|1}$ the reliability of different configurations by varying $\beta_{1,2|1}$ in [0,1]. The reliabilities of different configurations are computed as shown in Fig. 6. We can see that the reliabilities of TMTI and SMTI decrease as the value of $\beta_{1,2|1}$ increases. The smaller conjunction of errors increases the input diversity, and hence the reliabilities of TMTI and SMTI become higher.

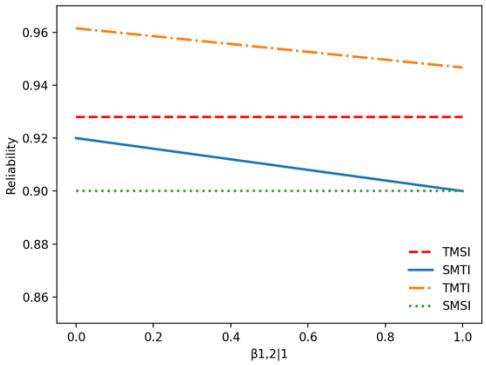

Figure 6: Reliability impacts of $\beta_{1,2|1}$

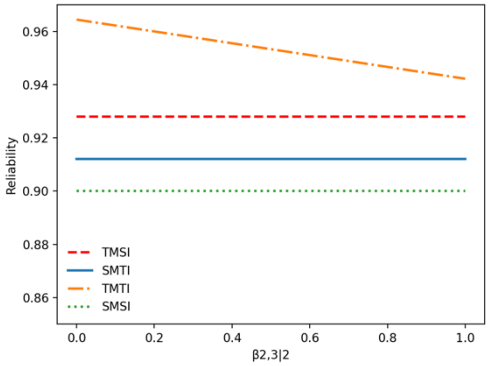

Figure 7: Reliability impacts of $\beta_{2,3|2}$

When we fix $\alpha_{1,2} = \alpha_{1,3} = \alpha_{2,3} = 0.3, \beta_{1,2|1} = \beta_{1,3|2} = \beta_{1,3|1} = 0.4$, and vary the value of $\beta_{2,3|2}$, the reliability of different configurations is changed as shown in Fig. 7. It also shows that the reliability of TMTI is better than other systems. As presented in Section IV.D.2, the reliability model for SMTI does not have the term $\beta_{2,3|2}$, only TMTI is affected by the change in the value of $\beta_{2,3|2}$. However, the reliability of TMTI is always better than other three-version architectures in this case as well.

Observation 3. The higher input diversity (i.e., the smaller value of conjunction of errors) results in the higher reliability of TMTI, when other parameters are fixed.

C. Three-version vs. two-version architectures

To answer Q3, we compare the reliabilities of DMDI and TMTI under different parameter settings.

1) Symmetric scenario

When we fix $\alpha_{1,2} = \alpha_{1,3} = \alpha_{2,3} = 0.3$, $\beta_{1,3|1} = \beta_{1,3|2} = \beta_{2,3|2} = 0.4$, the reliability of different version systems by varying $\beta_{1,2|1}$ is computed as shown in Fig. 8. As the value of $\beta_{1,2|1}$ increases, both reliabilities of DMDI and TMTI decrease. It is counterintuitive that the reliability of DMDI is better than TMTI regardless of the value of $\beta_{1,2|1}$. Then, we fix $\alpha_{1,3} = \alpha_{2,3} = 0.3$, $\beta_{1,2|1} = \beta_{1,3|1} = \beta_{1,3|2} = \beta_{2,3|2} = 0.4$ and vary the value of $\alpha_{1,2}$ in [0,1]. The reliabilities of TMTI and DMDI are computed as shown in Fig. 9. As the value of $\alpha_{1,2}$ increases, both reliabilities of DMDI and TMTI decrease. In this scenario as well, the reliability of DMDI is always better than TMTI.

For the reason why the reliability of DMDI is better than TMTI, we can see from the error output probabilities of DMDI and TMTI, it is related to the diversity parameters (e.g., $\alpha_{1,2}$).

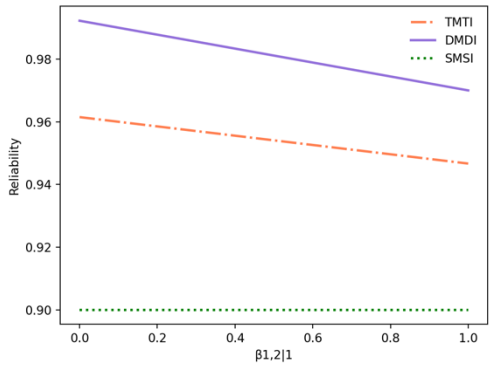

Figure 8: Reliability impacts of $\beta_{1,2|1}$

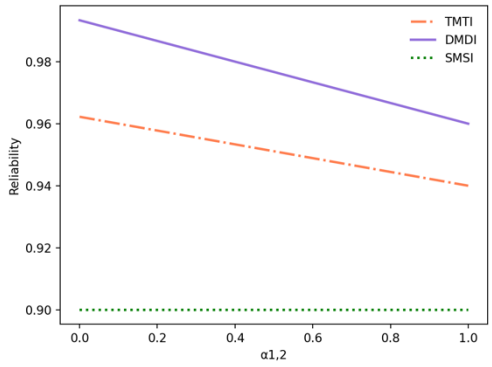

Figure 9: Reliability analysis by varying $\alpha_{1,2}$

2) Asymmetric scenario

Next, we consider an asymmetric case in terms of the intersection of errors by changing the values $\alpha_{1,2} = \alpha_{1,3} = 0.8$ and $\alpha_{2,3} = 0.1$, while keeping the values $\beta_{1,3|1} = \beta_{1,3|2} = \beta_{2,3|2} = 0.4$. The reliabilities of DMDI and TMTI by varying $\beta_{1,2|1}$ is computed as shown in Fig. 10. As the value of $\beta_{1,2|1}$ increases, the reliability of DMDI decreases compared to TMTI. When the value of $\beta_{1,2|1}$ is larger than 0.8, the reliability of TMTI is better than that of DMDI.

For an asymmetric case in terms of the conjunction of errors, we assign the values $\beta_{1,3|1} = \beta_{1,3|2} = \beta_{2,3|2} = 0.4$, $\beta_{1,2|1} = 0.8$, while setting $\alpha_{1,3} = 0.8$ and $\alpha_{2,3} = 0.1$. The reliabilities of DMDI and TMTI by varying $\alpha_{1,2}$ are computed as shown in Fig. 11. As the value of $\alpha_{1,2}$ increases, the reliability of DMDI decreases faster than the reliability decrease trend of TMTI. When the value of $\alpha_{1,2}$ is larger than 0.8, the reliability of TMTI is better than DMDI. The result of the asymmetric scenario indicates that there are cases where the reliability of TMTI can be better than DMDI, depending on the balance of the diversity metrics.

3) Results

Through the above analysis, we find that the reliability of DMDI tends to be better than the reliability of TMTI. When the values of $\beta_{1,2|1}$, $\alpha_{1,2}$, $\alpha_{1,3}$ become larger, but the value of $\alpha_{2,3}$ becomes smaller, which means more stringent restrictions, the advantage of TMTI emerges. Note that the result can also be attributed to the difference in decision rules for the final system output. In DMDI, we do not count the cases with single error output as a system error as we can detect the occurrence of the error at least. On the other hand, a TMTI system must reach a consensus on correct or error output by a majority voting rule.

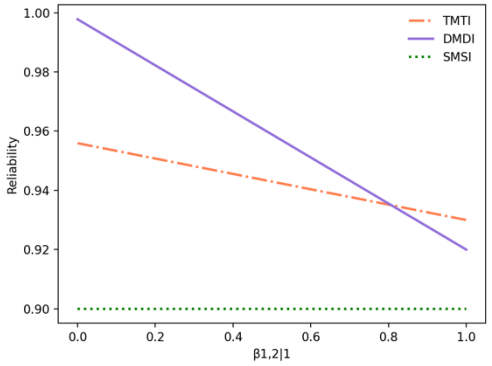

Figure 10: Reliability analysis by varying $\beta_{1,2|1}$

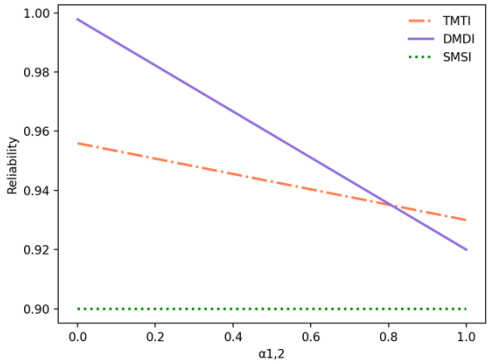

Figure 11: Reliability analysis by varying $\alpha_{1,2}$

Observation 4. The reliability of DMDI tends to be better than the reliability of TMTI because of the difference in decision logic. However, with more stringent restrictions, the advantage of TMTI can emerge.

D. Reliability impacts of TMTI

Among the different three-version architectures, we observe that TMTI tends to achieve the highest reliability. Thus, we further investigate how two kinds of diversity can affect the reliability of TMTI by changing the variance of diversity metrics. To investigate the impacts of variances of model diversity (i.e., the intersection of errors), we fix $f_1 = f_2 = f_3 = 0.1$, $\beta_{1,2|1} = \beta_{1,3|1} = \beta_{2,3|2} = 0.4$, and assign the values of $\alpha_{1,2}$, $\alpha_{1,3}$, $\alpha_{2,3}$ from $\{\mu_\alpha, \mu_\alpha \pm \sigma_\alpha\}$, where $\mu_\alpha = 0.3$ is the mean and σ_α is the absolute deviation of $\alpha_{1,2}$, $\alpha_{1,3}$, $\alpha_{2,3}$. We can see from Fig. 12, the trend of the reliability of TMTI is decreasing by varying σ_α. The greater the difference between the absolute deviation becomes, the lower the reliability is. On the same curve, when σ_α is equal to zero (i.e., the intersection between every two error sets is the same), the reliability of TMTI is maximized. Moreover, as the value of μ_α increases, the reliability of TMTI decreases, which is consistent with the results observed in Section V.A.

Next, to investigate the impact of the variance of input diversity (i.e., the conjunction of errors), we fix $f_1 = f_2 = f_3 = 0.1$, $\alpha_{1,2} = \alpha_{1,3} = \alpha_{2,3} = 0.3$, and assign the values of $\beta_{1,2|1}, \beta_{1,3|1}, \beta_{2,3|2}$ from $\{\mu_\beta, \mu_\beta \pm \sigma_\beta\}$, where $\mu_\beta = 0.3$ is the mean and σ_β is the absolute deviation of $\beta_{1,2|1}, \beta_{1,3|1}, \beta_{2,3|2}$. The reliability of TMTI by varying σ_β is computed as shown in Fig. 13. Different from the result in Fig. 12, the trend of the reliability of TMTI is increasing by varying σ_β. On the same curve, when σ_β is equal to zero (i.e., the conjunctions of errors are the same), the reliability of TMTI is minimized.

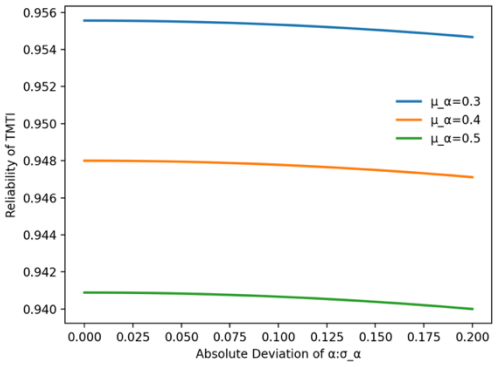

Figure 12: Reliability of TMTI by varying σ_α

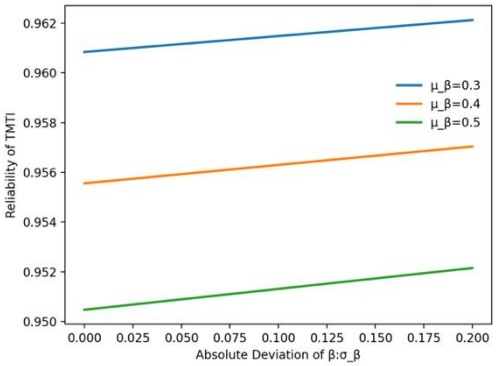

Figure 13: Reliability of TMTI by varying σ_β

Observation 5. A larger variance of model diversities negatively impacts the TMTI reliability, while a larger variance of input diversity has opposed impacts.

VI. Conclusion

In this paper, we apply a software fault-tolerant approach to improve the reliability of ML-based software systems. To investigate the effectiveness of N-version ML architectures, we propose reliability models for three-version architectures by introducing diversity metrics for measuring the diversity of ML models and the diversity of input data. We conduct numerical analysis on the proposed model and find that i) the reliability of TMTI systems is the highest among other three-version systems (i.e., TMSI and SMTI systems), ii) the reliability of a DMDI system is generally more reliable than the reliability of a TMTI system, and iii) the variance of model diversity negatively impacts on TMTI reliability, while the variance of input diversity has positive impacts. Our reliability models and the preliminary findings must value designs of safety-critical ML systems. In future work, we can extend our model to higher versions and compare the reliability of different architectures. Since TMTI architecture has the potential to achieve the highest reliability even compared with DMDI, we can analytically identify the conditions where TMTI achieves higher reliability than others. Estimating the parameter values of diversity parameters from empirical studies is also an important future work.

Acknowledgment

This work is partly supported by JSPS KAKENHI Grant Numbers 19K24337 and 22K17871.

References

[1] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, Machine learning testing: Survey, landscapes and horizons, IEEE Transactions on Software Engineering, 2020.

[2] K. Pei, Y. Cao, J. Yang, and S. Jana, DeepXplore: Automated whitebox testing of deep learning systems, In Proc. of the 26th Symposium on Operating Systems Principles, pp. 1–18. ACM, 2017.

[3] T. Chen, S. Cheung, and S. Yiu, Metamorphic testing: a new approach for generating next test cases. Technical report, Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, 1998.

[4] X. Xie, J. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Chen, Application of metamorphic testing to supervised classifiers. In Proc. of International Conference on Quality Software, pp. 135–144, 2009.

[5] S. Al-Azani and J. Hassine, Validation of machine learning classifiers using metamorphic testing and feature selection techniques. In International Workshop on Multidisciplinary Trends in Artificial Intelligence, pp. 77–91. Springer, 2017.

[6] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, and Y. Wang, Deepgauge: Multi-granularity testing criteria for deep learning systems. In Proc. of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018, pp. 120–131, 2018.

[7] L. Ma, F. Zhang, J. Sun, M. Xue, B. Li, F. Juefei-Xu, C. Xie, L. Li, Y. Liu, J. Zhao, and Y. Wang. Deep Mutation: Mutation Testing of Deep Learning Systems, 2018.

[8] W. Shen, J. Wan, and Z. Chen, MuNN: Mutation Analysis of Neural Networks. In Proc. of International Conference on Software Quality, Reliability and Security Companion (QRSC), pp. 108–115, 2018.

[9] I. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, https://arxiv.org/abs/1412.6572, 2014.

[10] Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929–1958, 2014.

[11] L. Chen and A. Avizienis, N-version programming: A fault-tolerance approach to reliability of software operation, In Proc. of 8th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-8), pp. 3-9, 1978.

[12] A. Gujarati, S. Gopalakrishnan and K. Pattabiraman, New wine in an old bottle: N-version programming for machine learning components, In Proc. of IEEE International Symposium on Software Reliability Engineering Workshops, pp. 283-286, 2020.

[13] F. Machida, N-version machine learning models for safety critical systems, In Proc. of the DSN Workshop on Dependable and Secure Machine Learning, pp. 48-51, 2019.

[14] L. Robert E and W. Vanderkulk, The use of triple-modular redundancy to improve computer reliability, IBM journal of research and development, Vol. 6, No. 2, pp. 200-209, 1962.

[15] M. Ege, A. Eyler M and MU. Karakas, Reliability analysis in N-version programming with dependent failures, In Proc. of IEEE EUROMICRO Conference, pp. 174-181, 2001.

[16] S. Latifi, B. Zamirai and S. Mahlke. PolygraphMR, Enhancing the reliability and dependability of CNNs, In Proc. of 50th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 99-112, 2020

[17] Y. Makino, T. Phung-Duc and F. Machida, A Queueing Analysis of Multi-model Multi-input Machine Learning Systems, In Proc. of the DSN Workshop on Dependable and Secure Machine Learning, pp. 141-149, 2021.

[18] F. Machida, On the diversity of machine learning models for system reliability, In Proc. of IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 276-285, 2019.

[19] H. Xu, Z. Chen, W. Wu, Z. Jin, S. Kuo, M. R. Lyu, NV-DNN: towards fault-tolerant DNN systems with N-version programming, In Proc. of the DSN Workshop on Dependable and Secure Machine Learning, pp. 44-47, 2019.

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

[21] Z H., Zhou, Ensemble methods: foundations and algorithms. Chapman and Hall/CRC, 2019.

[22] R. S. Ferreira, J. Arlat, J. Guiochet, and H. Waselynck, Benchmarking safety monitors for image classifiers with machine learning, In Proc. of IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 7-16, 2021.