2021 22nd IEEE International Conference on Industrial Technology (ICIT) | 978-1-7281-5730-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICIT46573.2021.9453482

A Feature-Based Machine Learning Approach for Mixed-Criticality Systems

Nelson Vithayathil Varghese
Deptartment of Electrical, Computer and Software Engineering
Ontario Tech University
Oshawa, Canada
nelson.vithayathilvarghese@ontariotechu.ca

Akramul Azim
Department of Electrical, Computer and Software Engineering
Ontario Tech University
Oshawa, Canada
akramul.azim@ontariotecthu.ca

Qusay H. Mahmoud
Department of Electrical, Computer and Software Engineering
Onatrio Tech University
Oshawa, Canada
qusay.mahmoud@ontariotechu.ca

***Abstract*—Driven by the recent technological advancements in the field of artificial intelligence, machine learning has emerged as a promising representation learning and decision-making method in many technological domains. Inspired by impressive these results, now machine learning techniques are also being applied to address the decision-making and control problems in the area of cyber-physical systems. For instance, some of these systems fall under the category of safety-critical systems such as chemical plants, autonomous vehicles, surgical robots, and modern medical equipment. One of the major performance issues related to the applicability of machine learning with safety-critical systems is related to the probability-based prediction nature of machine learning components used within such systems. This particular characteristic of machine learning makes it extremely difficult to guarantee safety as directed by standards such as ISO 26262. More importantly, the non-transparent and complex nature of machine learning algorithms make both the reasoning as well as formally establishing the safety aspects of the underlying system extremely difficult. The objective of this research work is to investigate on this key issue, and further on propose an efficient machine learning methodology based on the mixed-criticality approach feasible to safety-critical systems.**

Keywords— Machine Learning, Deep Learning, Cyber-Physical Systems, Safety-Critical Systems, Partitioning.

I. Introduction

Recent years have been witnessing an ever increasing level of development of machine learning (ML) based components. These components are still unavailable in many of the safety-critical systems but there is an increasing interest to add machine learning-based components into such systems. Undoubtedly, ML components brings a lot of advantages by working with incomplete knowledge of the system and help to do predicative and prescriptive analysis. Some of the safety-critical systems such as autonomous vehicles or surgical robots can be benefited by adding machine learning-based features but at the same time, safety needs to be guaranteed to use them widely. An incorrect decision or analysis may cause a catastrophe in such systems and therefore ML algorithms cannot be used without understanding the safety impacts. Hazard and risk analysis are essentially required to be performed to better understand the impacts of various functionalities in autonomous vehicles as mandated by the safety standard ISO 26262 [1]. Moreover, these standards often lack details related to the evaluation of the use of ML algorithms. On the other hand, real-time systems mandate the availability of correct values at the right time and no deadline misses are allowed. These tight constraints make it hard for ML algorithms to make them acceptable for guaranteeing functional safety within safety-critical systems environment. However, non safety-critical systems can still use ML algorithms without such issues. Often safety-critical and non safety-critical components co-exist in a mixed-criticality system and therefore it is required to separate the functional and safety requirements for each of these components.

The traditional approach followed within the safety-critical system design was predominantly based on the deterministic software components. Recent advancements in technologies such as ML, and deep learning (DL) motivated to have probabilistic systems integrated as part of modern safety-critical systems design process. However, ML systems implemented by leveraging on the power of DL with complex neural network algorithms represent probabilistic decision making. This is primarily due to the fact that neural network-based systems work based on the function approximation methodology. In other words, aforementioned probabilistic nature of ML algorithms conflicts with various safety requirements and mandates that are set by the safety standards such as ISO 26262 towards the safe adaptation of ML components within the safety-critical systems [1]. As a result of this change, modern safety-critical systems designed with ML components are required to have mandatorily increase their safety, robustness, and reliability barriers to ensure that these systems function without any failure within their large input space. At the same time, some of the characteristics of ML algorithms posed major challenges to meet this new safety, robustness, and reliability requirements.

This paper proposes a partitioned mechanism for separating the safety-related and non-safety related components in a mixed-criticality system. The novelty of the work is to identify the safe ML features from the input space and match them with the components that require functional safety as mandated by standards. This requires us to apply filtering-based mechanisms to identify partitions based on the threshold probability to map to different safety integrity levels (SIL) of IEC 61508.

The remainder of the paper is organized as follows: Section II discusses the background of the paper and Section III provides details on the state-of-the-art with respect to the research efforts

978-1-7281-5730-6/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: Ontario Tech University. Downloaded on June 23,2021 at 17:42:00 UTC from IEEE Xplore. Restrictions apply.

done on ML in safety critical systems. Section IV presents the problem statement related to the uncertainty factors associated within the safety-critical systems with ML components. The Section V discusses the proposed system model designed based on the mixed-criticality based approach that could be used to address the problem of inherent uncertainty. The contents outlined under Section VI gives details about the methodology adopted to implement the proposed solution. The section VII shares the experiment plans used for the evaluation of the proposed solution, and various test results obtained during the evaluation of the proposed solution's prototype. Finally, the Section VIII provides a conclusion about the proposed solution.

II. Background

A. Safety Aspect of Machine Learning

Recent advancements within the ML arena, especially the evolution of DL, have motivated designers and developers for increasing adoption of intelligence in various safety-critical software systems such as self-driving cars and medical diagnosis [2]. The traditional way of engineering a safety-critical system had some level of uncertainty but often software architecture design was able to handle such uncertainties to a great extent. Additionally, the existence of various safety standards such as ISO26262 acted as an additional layer to filter out any potential hazards which were still existed within the designed solution. However, the addition of ML brings new challenges of handling increased uncertainty due to the probabilistic-based prediction. Therefore, the prediction of an ML algorithm requires to be deterministic to enable ML safely in safety-critical systems. The accuracies of ML algorithm can validate the deterministic behavior of ML algorithms for various inputs. This will also help us to handle different criticalities in a mixed-criticality system.

B. Safety Challenges of Machine Learning Algorithms

Some of the key characteristics of ML algorithms that raise challenges concerning the safety, reliability, and robustness aspects of the safety-critical systems are mentioned as follows.

Model Complexity: Implementations of ML systems that are based on DL use the neural networks, which are often quite complex. From a system perspective, machine learning models are often considered difficult to provide evidence of safety [1]. More importantly, attempting to provide evidence of safety of these models is infeasible.

Opacity: Machine learning models with strong expressive power, especially having deep neural network implementations, are often considered to be highly non-transparent. Having a high level of non-transparency is an obstacle to safety assurance as it makes extremely difficult to understand by an assessor. In such situations, the model appears almost like a black box, and hence it is almost impossible to build sufficient confidence that the model is operating as intended [3].

Fully Probabilistic Output: The softmax layer output of deep learning-based systems represents probabilistic decision making. This probability-based approach makes it difficult to analyze safety requirements as mandated by standards [3].

Perturbation: Deep learning algorithms often work under the assumption that both training and testing data are drawn from the same distribution, and due to this algorithm could be highly sensitive even to small distribution drifts in data (perturbation). A safety-critical system such as an autonomous vehicle has the input space that vary widely and therefore the perturbation effect can cause a catastrophic consequence [3].

Limited Testing Scenarios: Large input space puts a cap on the number of test scenarios that are feasible to test as evidence on deciding the deterministic behaviour requires to operate on bounded input space. Consequently, model deployed requires to operate on large input dataset which is difficult to obtain in many cases [3]. In other words, there is a high probability that an input feature space that lacks training data can have a much higher error rate, which could potentially lead to serious safety concern for the underlying safety-critical system.

III. Related Work

This section provides details on the state-of-the-art from the literature on the related work done within the ML based safety critical systems front. One of the most recent prominent works namely N-version ML architecture that aims to improve system reliability against the probabilistic outputs of individual ML modules [4]. The core idea of this architecture is based on the aspect of exploiting two kinds of diversities; input diversity and model diversity. This approach attempts to address existing ML systems' output uncertainty factor as it is being a big obstacle to ensure the quality of safety critical applications like autonomous vehicle [4]. Another research effort done in the same arena was related to mitigating the uncertainty factors associated with ML algorithms through the development of software architecture design patterns, and how it can potentially decrease the negative impact that ML components can have on safety critical systems [3]. Another active research work done by the DNV GL focuses, with much importance, on how phenomenological knowledge can be incorporated in probabilistic machine learning models, together with the use of active learning for the optimal decision making under uncertainty [5].

IV. Problem Statement

From the very basic probability-based prediction nature of ML algorithms, systems designed with ML-based components may incorrectly label a desired behaviour as an undesirable one or vice-versa. From a safety-critical system perspective, both of these cases may cause a catastrophic consequence. In general, the accuracy and effectiveness of the approximations made by ML-based systems heavily rely on the algorithms that are trained by chaining non-linear and linear transformations. Another key aspect of the ML algorithms is that they mostly appear as black-box components within the systems in which they are part of or integrated with. More importantly, ML algorithms that are based on the DL implementations often are sensitive to input distribution shifts. This is better understood when the predicted output and the related confidence of the underlying ML model are dropped significantly because of small perturbations applied to inputs. From a safety-critical system perspective, this often leads to a problem named inherent uncertainty. The objective of this research work will be on addressing this key issue -inherent uncertainty problem associated with the effective usage of machine learning within safety-critical systems.

Authorized licensed use limited to: Ontario Tech University. Downloaded on June 23,2021 at 17:42:00 UTC from IEEE Xplore. Restrictions apply.

V. System Model

The core objective of the proposed model is to address the inherent uncertainty problem mentioned above. As a solution, come up with a novel approach that would make every effort to reduce the negative impacts associated with deploying safety-critical systems having machine learning-based components with fully probabilistic output and highly sensitive to small distribution shifts (perturbation) with input values. Table 1 given below shows the list of various notations that will be used during the explanation of the proposed system and its related details in the forthcoming sections of this paper.

TABLE I. Notations

Name	Details
X	Input Feature Space
$\mathscr{R}$	Feature Partition
S($\mathscr{R}$)	Fitness score of a feature partition
M	Number of feature partitions
N	Size of a single partition
R($\mathscr{R}$)	The rank of a feature partition
ϕSML	Safe Machine Learning Partition
ϕRML	Regular Machine Learning Partition
TH($\mathscr{R}$)	Partition Filter Threshold value
m	Index of a randomly selected partition
Y	System Output Space
F($\mathscr{R}$)	Objective function

The core idea behind the proposed system model is based upon the two major aspects namely, mixed-criticality based partitioning of the input feature space, and delegation of responsibility. The key idea behind mixed-criticality oriented partitioning is to separate the input parameter set (denoted by input feature space as X) into different partitions based on an evaluation criterion. Likewise, ddelegation of the responsibility refers to the matter of deciding which particular module, safe machine learning module, or regular machine learning module, will handle each of the input feature space partitions.

Fig. 1 shows the high-level architecture of proposed system with various layers involved within its architecture. This model can be explained from a multi-layer perspective wherein the first layer named Input Parameter Space(X), refers to the complete set of inputs that the safety-critical system needs to handle. It is reasonable to assume that not all kinds of the inputs within the system will have the same level of priority and severity level. Based on this assumption, input parameter space will be divided into multiple partitions. Often, the baseline for the partitioning can be coupled with the safety integrity level of the safety-critical system's design according to the safety requirements set by standards such as ISO26262. Once the input feature space has been divided into the multiple partitions, there needs to be partition table layer maintained with the details of partitions that are created. This layer will be kind of a reference for routing the data to either a safe machine learning layer (SML) or a regular machine learning (RML) layer further down the line for the processing. The next layer handles the processing of data inputs, wherein data inputs from the SML feature partitions(ϕSML) will be handled by the respective SML components within the system. Similarly, all the data inputs from RML feature partitions(ϕRML) will be sent to the SML components. Once these data inputs have been processed by the respective units, the results will be sent to system output space for further analysis. One key aspect of the delegation of responsibility is to ensure that the handling of safe machine learning feature partitions by the SML component within the system would be inline and matching the safety requirements set by standards such as ISO26262 [1].

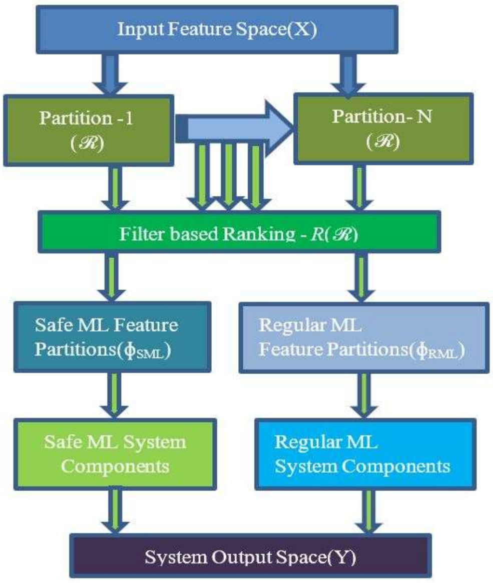
Input Feature Space(X)
Partition -1 ($\mathscr{R}$)
Partition- N ($\mathscr{R}$)
Filter based Ranking - $R(\mathscr{R})$
Safe ML Feature Partitions(ϕ_{SML})
Regular ML Feature Partitions(ϕ_{RML})
Safe ML System Components
Regular ML System Components
System Output Space(Y)

Fig. 1. A High-level Architecture of the Proposed System

From the perspective of a safety-critical system having machine learning components within it, safe machine learning feature partitions(ϕSML) can be considered as the highly critical inputs, whereas the regular machine learning feature partitions (ϕRML) can be treated as less critical inputs. The key idea is that by representing a specific input feature partition as SML feature partition(ϕSML), it is expected by the safety-critical system that the machine learning component handling it should be able to deliver the output by following the respective safety standards' set limits defined. Similarly, by designating the RML feature partition(ϕRML) as a less critical, safety-critical system expects that factors like fully probabilistic output and highly sensitive to small distribution shifts (perturbation) are not going to cause any kind of violations concerning the safety standard requirements.

VI. Methodology

The two major aspects of core methodology adopted behind proposed system design are mixed criticality- based partitioning and a ranking based partition filtering scheme. Input feature space partitioning is based on the concept of feature engineering methods adopted from the ML domain, which will be further followed by a ranking based filtering. In other words, an adaptation of input feature space partition selection followed by a score(rank) based partition filtering methodology will be the backbone of the proposed approach. This method will ensure that, from the wide spectrum of input feature space that a safety-critical system needs to deal with, only relevant feature space partitions will be filtered out towards the processing by machine

Authorized licensed use limited to: Ontario Tech University. Downloaded on June 23,2021 at 17:42:00 UTC from IEEE Xplore. Restrictions apply.

learning model. Thereby the safety system components based on the machine learning models will be in a better position to produce results that can meet the safety requirements set by standards such as ISO26262. Our objective is to distinguish the feature space partitions with highly accurate results possible with ML components used within the safety systems, from those with a low accuracy result.

A. Mixed Criticality Based Approach

The objective of this section is to give details about a mixed-criticality approach that can be applied to the ML used within safety-critical software systems. One of the key motives behind coming up with such an approach is that it can successfully or up to a great extent mitigate the challenges posed by ML algorithms such as fully probabilistic output and highly sensitive to small distribution shifts (perturbation) with input values. The key idea behind this mixed-criticality approach is based on an input feature space partitioning strategy [6].

B. Feature Based Partitioning

The objective of partitioning is to divide input feature space (X) into multiple partitions. Each of these individual partitions will represent one section of the wide input feature space. Once the partitions are made, it is possible to analyze each of these partitions in depth for various aspects such as the size of the feature samples available for training the ML algorithms. Each of the individual partitions($\mathscr{R}$) can be evaluated using a custom-designed objective function to measure its fitness value, S($\mathscr{R}$). In our current context, the nature of the objective function and fitness evaluation can be tailored to the nature of the safety requirements mandated by the safety standards for the safety-critical system. Fitness evaluation by using an objective function can be interpreted as a scoring process for each of these partitions. The score values act as the basis for the ranking scheme which will be detailed in the coming section. More importantly, score value can be indirectly an indicator of whether this partition will be treated as a SML feature partition or a RML feature partition. For instance, after the evaluation of the partition with the objective function, having a fitness value ranging between 1-100 could mean that input feature space values for that partition correspond a particular rank value, R($\mathscr{R}$). This fitness value can be influenced by the number of training samples available in that particular input feature space. levelIn other words, boundaries of feature space partition can interpret whether it will be processed by a SML component or not, otherwise, it could also indicate whether this feature space input can have a less impact on the overall system safety and reliability (for instance, a 'Fail-Safe' point). Finally, the partition-based approach also gives the flexibility to reject input instances from data-lacking feature space partitions. With this approach, it is possible to view the entire input space as a feature space partition tree. From the viewpoint of a safety-critical system embedded with ML components in it, SML feature partitions(ϕSML) denotes the highly critical inputs, and RML feature partitions(ϕRML) are treated as less critical.

Fig. 2 depicts the process of partitioning based on the evaluation(objective) function, F($\mathscr{R}$), wherein each partition from the feature space will be evaluated for the fitness score value, S($\mathscr{R}$).

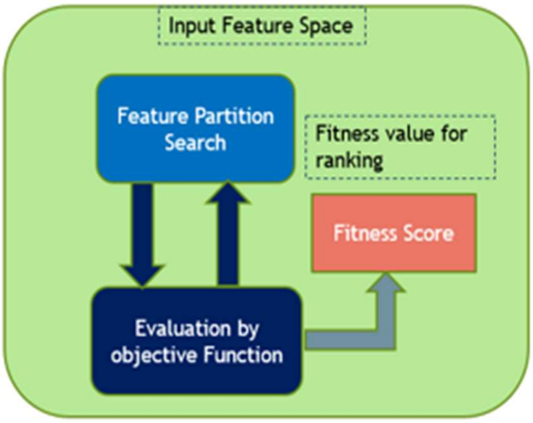

Fig. 2. Feature Space Partition Evaluation

C. Ranking Based Filtering

Along with the feature space partitioning, another important aspect of the methodology is the filtering scheme that is coupled with the ranking for each of the input feature space partitions ($\mathscr{R}$). The filter method with variable ranking criteria will be used for the ordering of input feature space partitions for selection by the machine learning components of the safety-critical systems. The ranking method is used to generate a sorted list of partition listing, by which we can channelize the input feature space partitions for the efficient and deterministic based processing either by the SML components or by the RML components within the safety-critical system. Depending on the mandates directed by the safety standards, a suitable ranking criterion is used to assign ranks to the input feature space partitions for deciding whether to route the input partitions to safe machine learning or non-machine learning components within the safety-critical system. More importantly, the filtering of ranked partitions will be centered around a threshold score value, TH($\mathscr{R}$). The threshold value that needs to be used is decided based upon the factors such as the safety-integrity level of the system under consideration, according to the safety standard requirements that the given safety system needs to meet. Having the aforementioned threshold value-based filtering system in place will ensure that all those partitions having ranking higher than the threshold filtering value will be handled by the safe machine learning components within the system, whereas remaining input feature space partitions will be handled by the regular machine learning components. This way, it will be ensured that always data from the right input feature space partitions will be channelized to the right system component – either to SML or RML, and prevent the occurrence of any kind of catastrophe that could result from the wrong assignment.

D. Partitioning and Ranking Algorithm

Algorithm mentioned below illustrates procedure involved within the feature space partitioning followed by the filter based ranking scheme for the proposed system model. As illustrated under the feature-based partitioning and rank based filtering sections above, this algorithm facilitates the partitioning and ranking process.

Input: Input Feature Space: X^{MxN} ,FilterValue: PFV

Authorized licensed use limited to: Ontario Tech University. Downloaded on June 23,2021 at 17:42:00 UTC from IEEE Xplore. Restrictions apply.

```
Output: Output Probability Value: $OPV^M$

M is the number of feature space partitions in X;
N is the size of a single feature partition, $\mathscr{R}$;
$\phi_{SML}$ ← [];
$\phi_{RML}$ ← [];
for m € M do
S($\mathscr{R}$) - PartitionFitness$_m$ ← X [m]: Objective Function;
R($\mathscr{R}$)- PartitionRank$_m$ ← X [m]: Ranking(PartitionFitnes$_m$);
if R($\mathscr{R}$)- PartitionRank$_m$ > TH($\mathscr{R}$), $\phi$SML[m] ←
    X[m]:safeML;
if R($\mathscr{R}$)- PartitionRank$_m$ < TH($\mathscr{R}$), $\phi$MRL[m] ←
    X[m]:safeML;
end
return
```

VII. Experiments And Evaluation Results

Below section provides the details on the experiments and results obtained with the proposed model.

A. *Experiments*

The prototype of the proposed feature-based safe machine learning system for the safety-critical system was tested using a machine learning-based model. As part of the prototype experiment and evaluation plan, a standard machine learning dataset named scene was used, and taken from the MULAN library [7]. It is a Java library for learning from multi-label data. It offers different types of classification, ranking, thresholding, and dimensionality reduction algorithms, along with other algorithms for learning from hierarchically structured labels. For prototype validation, scene recognition dataset was used which is a multivariate dataset with different feature values and having multiple labels [8]. Used dataset contains characteristics about images and their classes. The original dataset is a multi-label classification problem with 6 different labels: Beach, Sunset, Fall Foliage, Field, Mountain, and Urban. The machine learning model used for this system prototype is based on the classification algorithm named KNN algorithm (K-Nearest Neighbors) [9]. As the dataset is multivariate having multiple labels, MLKNN which is the multilabel version of the standard KNN algorithm was used for building the system model. MLKNN is derived from the single label classifier [10] and therefore has a popularity to use for multiple labels classification. The classifier uses the majority of K neighboring samples to determine the labels of unseen data. If the posterior probability is higher among the neighboring samples then it does mean that the distribution of sample neighbors is higher. This will determine the class for a multi-label sample in MLKNN. Decision for test data is made based on neighbors in the training data. The value K is set to 10 in our experiments. With the help of the MLKNN algorithm, a prototype machine learning model was trained on the chosen dataset [10]. Additionally, as part of the prototype evaluation plan, the population of input feature data space partitions was generated by adopting an evolutionary algorithmic-based approach named differential evolution (DE) [11]. It is a method used in evolutionary computation to optimize a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones. In the current context, for prototype evaluation, this approach was used to generate a population of input feature space segments ($\mathscr{R}$), having different fitness values, S($\mathscr{R}$). Each of these individual feature space segments, ($\mathscr{R}$), will be associated with a fitness value, S($\mathscr{R}$), which will be further evaluated using an objective function, F($\mathscr{R}$), during the partitioning process. For instance, the objective function, F($\mathscr{R}$), could be trying to evaluate the hamming loss associated with chosen input feature space partition, and then assign a fitness value, S($\mathscr{R}$), accordingly. Further on, all of these input feature space segments that are partitioned according to the fitness score values, will be further assigned a respective rank value, R($\mathscr{R}$) with a filtering based ranking procedure. The above-mentioned method enables us to categorize the input feature as either safe machine learning feature partitions(ϕSML) or regular machine learning partitions(ϕRML). For the validation of the proposed prototype, MATLAB R2019b version was used as the development platform to conduct the experiments with the mentioned dataset.

B. *Evaluation Results*

As part of the experiment evaluation analysis for the proposed model, Fig. 3 indicates test results captured for the metrics related to hamming loss vs partition ranking.

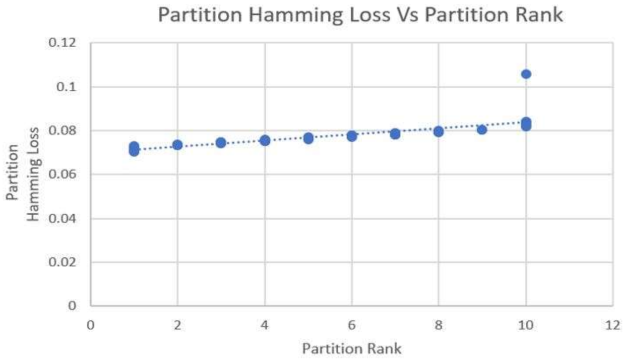

Fig. 3. Hamming Loss – Partition Rank

Key observation from Fig. 3 indicates that with the proposed partition-based, filtering rank method, from the wide spectrum of input feature space (X), it is possible to distinguish those input feature space partitions ($\mathscr{R}$), that are more suitable to be named as safe machine learning partitions(ϕSML) category. This is predominantly because input feature space partitions, ($\mathscr{R}$), with a less hamming loss, are associated with high-rank values, R($\mathscr{R}$). This indicates that those feature space partitions($\mathscr{R}$), having high-rank values R($\mathscr{R}$) greater than the threshold value, TH($\mathscr{R}$), could be processed by the safe machine learning components within the safety-critical system to satisfy the safety-related requirements mandated by respective standards such as ISO26262. In our case study experiments, notion of the threshold value is linked with hamming loss such as value less than or equal to 0.08. This value can be closely linked with the safety integrity levels of the respective application domain safety standards.

Authorized licensed use limited to: Ontario Tech University. Downloaded on June 23,2021 at 17:42:00 UTC from IEEE Xplore. Restrictions apply.

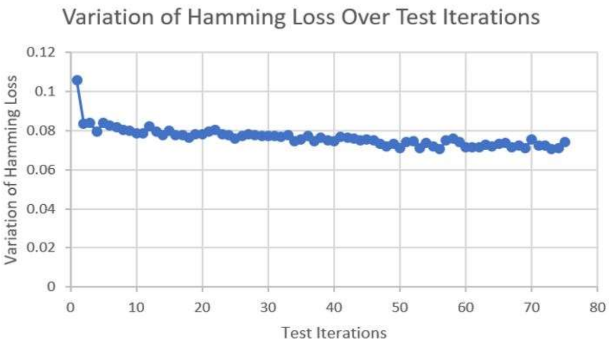

Fig. 4. Hamming Loss Improvement over Test Iterations

Fig. 4 shows the variation of hamming loss improvement over multiple test iterations. This prototype experiment was conducted for a relatively smaller number of test iterations with a less computationally intensive offline dataset. The results indicate that this proposed approach could be extended to safety-critical systems having more-critical datasets to achieve the same kind of results.

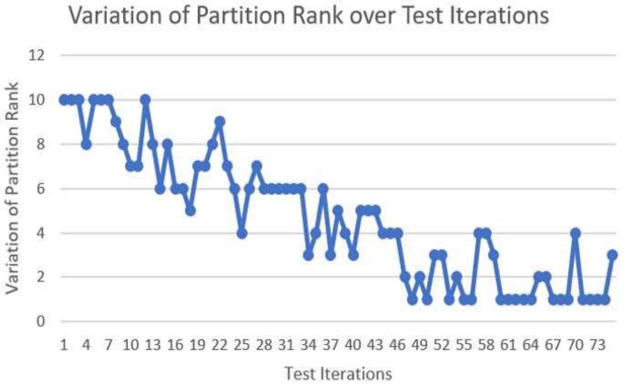

Fig. 5. The Ranking Improvement over Test Iterations

Fig. 5 shows the details on variation of ranking improvement R($\mathscr{R}$) over multiple test iterations. Having seen from this result, partitions having high-rank values are associated with low Hamming loss. Hence, it is reasonable to conclude that this ranking based approach can separate the input feature space partitions as safe machine learning partitions and regular machine learning partitions. In comparison to the state-of-the-art detailed in section II, mixed-criticality oriented approach attempts to embed the ML into the safety-critical systems by following the combination of input feature space partitioning followed by a ranking based filtering to mitigate the impacts ML's uncertainty factor.

VIII. Conclusion

In this research work, we investigated inherent uncertainty problem associated with safety-critical systems built with ML components, and proposed a new approach to mitigate this issue. The core design idea behind the proposed system model is based on two major aspects, mixed-criticality oriented partitioning of the input feature space, and delegation of responsibility. With this methodology, input feature space partitions are divided into two classes- SML partitions and RML partitions, which will be handled by SML components and RML respectively. Based on the test results obtained with ML model built with this approach, it is evident that feature space partitioning followed by filtering based ranking can separate the input feature space partitions into two classes as stated above. In this paper, we show that feature space partitions having high-rank values greater than the threshold value could be processed by the safe machine learning components within a mixed-critical system to satisfy safety requirements mandated by respective standards.

Acknowledgement

This work was supported in part by a Natural Sciences and Engineering Research Council of Canada (NSERC) Grant.

References

[1] M. Gharib, P. Lollini, M. Botta, E. Amparore, S. Donatelli and A. Bondavalli, "On the Safety of Automotive Systems Incorporating Machine Learning based Components: A Position Paper," in 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2018.

[2] N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez and Y. Tassa, "Learning continuous control policies by stochastic value gradients," in Advances in Neural Information Processing Systems, 2015, pp. 2944--2952.

[3] A. C. Serban, "Designing safety critical software systems to manage inherent uncertainty," in 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), IEEE, 2019, pp. 246--249.

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

[5] C. Agrell, "https://www.dnvgl.com/," DNV GL, 2019. [Online]. Available: https://www.dnvgl.com/research/review2019/featured-projects/probabilistic-machine-learning.html. [Accessed 14 November 2020].

[6] X. Gu and A. Easwaran, "Towards safe machine learning for cps: infer uncertainty from training data," in Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, 2019, pp. 249--258.

[7] G. Tsoumakas, I. Katakis and I. Vlahavas, "Mining multi-label data," in Data mining and knowledge discovery handbook, Springer, 2009, pp. 667--685.

[8] M. R. Boutell, J. Luo,, X. Shen and C. M. Brown, "Learning multi-label scene classification," Pattern recognition, vol. 37, no. 9, pp. 1757--1771, 2004.

[9] N. Suguna and K. Thanushkodi, "An improved k-nearest neighbor classification using genetic algorithm," International Journal of Computer Science Issues, vol. 7, no. 2, pp. 18--21, 2010.

[10] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," Pattern recognition, vol. 40, no. 7, pp. 2038--2048, 2007.

[11] K. V. Price, "Differential evolution," in Handbook of Optimization, Springer, 2013, pp. 187--214.

Authorized licensed use limited to: Ontario Tech University. Downloaded on June 23,2021 at 17:42:00 UTC from IEEE Xplore. Restrictions apply.