Accurately Predicting Probabilities of Safety-Critical Rare Events for Intelligent Systems

Ruoxuan Bai[1], Jingxuan Yang[2], Weiduo Gong[3], Yi Zhang[4], *Senior Member, IEEE,* Qiujing Lu[5], Shuo Feng[6*], *Member, IEEE*

Abstract— Intelligent systems are increasingly integral to our daily lives, yet rare safety-critical events present significant latent threats to their practical deployment. Addressing this challenge hinges on accurately predicting the probability of safety-critical events occurring within a given time step from the current state, a metric we define as "criticality". The complexity of predicting criticality arises from the extreme data imbalance caused by rare events in high dimensional variables associated with the rare events, a challenge we refer to as the curse of rarity. Existing methods tend to be either overly conservative or prone to overlooking safety-critical events, thus struggling to achieve both high precision and recall rates, which severely limits their applicability. This study endeavors to develop a criticality prediction model that excels in both precision and recall rates for evaluating the criticality of safety-critical autonomous systems. We propose a multi-stage learning framework designed to progressively densify the dataset, mitigating the curse of rarity across stages. To validate our approach, we evaluate it in two cases: lunar lander and bipedal walker scenarios. The results demonstrate that our method surpasses traditional approaches, providing a more accurate and dependable assessment of criticality in intelligent systems.

I. INTRODUCTION

Intelligent systems are playing an increasingly significant role in our daily lives. Safety-critical autonomous systems are the intelligent systems whose failure will cause considerable losses of human life and property, such as autonomous vehicles, robotics, and intelligence medical diagnosis system [1]–[3]. Given the potential catastrophic consequences, it is imperative to guarantee these systems operate safely and effectively for practical deployment.

To address the aforementioned challenges, the key lies in precisely predicting the probabilities of safety-critical events associated with their failures within a given time step from current state, a metric we refer to as "criticality". These events may span from a traffic accident involving an autonomous vehicle to a robot tipping over. Previous criticality prediction methods can be broadly categorized into metrics-based methods and model-based methods.

The metrics-based criticality prediction methods rely on specific metrics designed to directly quantify risk, for instance the well-known Time To Collision (TTC) metric used in automated driving to assess the probability of collision [4]–[6]. However, these metrics may not always yield precise results, and formulating appropriate metrics in complex systems can be a challenging task. In such scenarios, predictive models trained on data have emerged as a viable alternative. Although criticality prediction models have garnered significant attention in the field of disease diagnosis [7]–[9], achieving precise predictions remains exceptionally challenging.

In this study, we aim to develop a criticality prediction model for intelligent systems that achieves both high precision and recall rates. A high recall rate is essential to reduce missed alarms, ensuring that all safety-critical events can be identified and timely measures can be taken to avert potential safety hazards. A high precision rate helps mitigate false alarms, preventing the systems from being overly conservative. The primary challenge arises from the rarity of safety-critical events in high-dimensional scenarios , often referred to as the "curse of rarity" [10]. This rarity results in an extremely imbalanced dataset, presenting significant obstacles in learning a well-performed model.

A long line of work have been proposed to overcome the imbalanced dataset issues. One popular approach is data re-sampling strategies, which often have difficulties in learning robust and generalizable features. The over-sampling method [11]–[13] can lead to overfitting on the minority class, while the under-sampling method [14], [15] may cause significant information loss in the majority class, leading to underfitting. Another approach is data re-weighting method [16]–[21], where different weights are adaptively assigned to different classes or samples. Nonetheless, a potential problem is that the classifier may converge to a local optimal solution if the batches are dominated by majority class samples. These methods often yield a considerable number of false alarms, thereby resulting in a low precision rate. Recently, researchers have introduce decoupling method, meta-learning, transfer learning and contrastive learning [22]–[27] to enhance imbalanced learning. However, despite these efforts, existing approaches have struggled to obtain models with high precision and recall rate when confronted with extremely imbalanced testing dataset.

To effectively resolve such issues, we propose a multi-stage learning framework that gradually densifies the dataset and alleviates the degree of imbalance. As illustrated in

[1,2,3,5]Ruoxuan Bai, Jingxuan Yang, Weiduo Gong and Qiujing Lu are with the Department of Automation, Tsinghua University, Beijing 100084, China (brx22, yangjx20,gwd21, qiujinglu@mails.tsinghua.edu.cn).

[4,6]Yi Zhang, Shuo Feng are with the Department of Automation, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China (zhyi, fshuo@tsinghua.edu.cn).
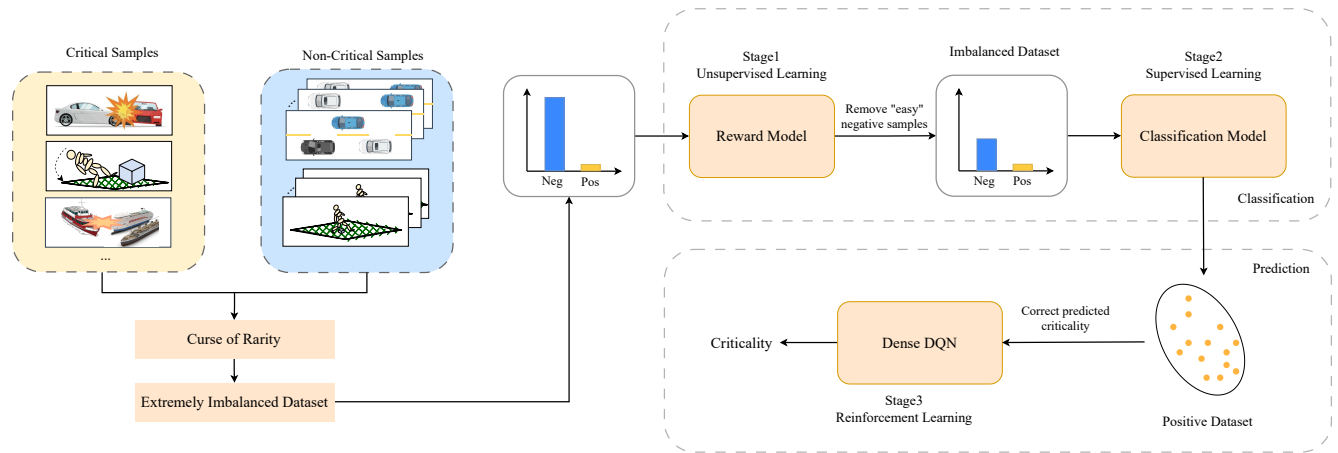
Fig. 1. Overview of the multi-stage learning framework. Our approach consists three stages. In the first stage, we remove those obvious non-critical samples and reduce imbalance ratio by unsupervised learning. Then in the second stage, we use labeled data to train a supervised classification model to further categorize those samples unable to be distinguished by unsupervised learning model. Lastly, in the third stage, we turn to improve the accuracy of predicted criticality other than continuously focusing on unclassified samples. Dense DQN method is developed to fine-tune last layers of classification model.

Fig. 1, in the first stage, we employ unsupervised learning methods to filter out "easy" negative samples, the criticality of which is considered to be zero. Subsequently, in the next stage, we turn to supervised learning and the enhanced bilateral-branch network (BBN) [28] is designed to perform more fine-grained classification task. Intuitively, samples difficult to classify in one feature space may become easier to separate in another space. By combining supervised and unsupervised methods, samples are projected into different state spaces for classification, enabling a coarse-to-fine identification of safety-critical events. Eventually, dense reinforcement learning method [29] is introduced to further improve the precision of predicted criticality. The effectiveness of dense reinforcement learning stems from the utilization of a more balanced gradient to fine-tune classification model in the second stage.

To evaluate the effectiveness of our proposed method, we conducted experiments in two different scenarios and compared our approach with classical methods. The results demonstrate that our predictive model achieves high precision and recall rate, outperforming traditional approaches. Our contributions can be summarized as follows:

- To address the challenges posed by the curse of rarity, we develop a multi-stage learning framework to learn criticality prediction models with high precision and recall rate by progressively mitigating the imbalance ratio and enhancing accuracy.

- Our method exhibits applicability to real-world scenarios featuring extremely imbalanced test datasets, with an imbalance ratio exceeding 10^4.

The rest of this paper is organized as follows. The problem of criticality prediction is formulated in Section II. Our approach and algorithms are described in Section III. In Section IV, we analyze experiment results. Finally, Section V summarizes our proposed method and prospects for future work.

II. PROMBLEM FORMULATION

In this section, we formulate the problem of criticality prediction. If we denote a critical event as A and the state of systems and environment as $\boldsymbol{X}$, the task of criticality prediction essentially involves estimating the conditional probability $\mathbb{P}(A|\boldsymbol{X})$. The criticality prediction we study in this work is in essence a rare-event probability estimation problem in a high-dimensional space due to the curse of dimensionality and the curse of rarity. The curse of dimensionality refers that state $\boldsymbol{X}$ is spatially and temporally complex, requiring numerous high-dimensional variables for accurate representation. The "curse of rarity", which is the primary source of our challenges, refers to the infrequent occurrence of the critical event A. This rarity implies that the majority of points within the variable space do not represent critical situations, leading to unreliable and noisy data.

Firstly, we formulate the problem of criticality prediction as a classification task. The objective is to train a classification model that can accurately determine whether a critical event A will occur given the current state $\boldsymbol{X}$ of the system. To this end, suppose we have a dataset $\mathcal{D}$ containing N samples $\{\boldsymbol{X}_k, \boldsymbol{Y}_k\}_{k=1}^N$. Each sample $\boldsymbol{X}_k$ comprises the state of the intelligent system and its surrounding environment. The corresponding label $\boldsymbol{Y}_k$ are binary, taking values from $\{0, 1\}$. These data is labeled by

$$\boldsymbol{Y}_k = \mathbb{I}_A(\boldsymbol{X}_k), \tag{1}$$

where $Y_k = 1$ if the critical event A occurs within a certain time step given the state X_k and $Y_k = 0$ otherwise.
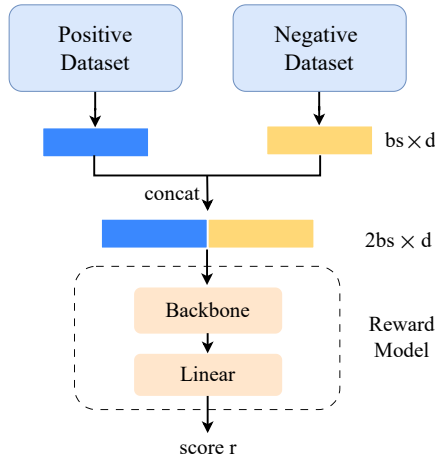
Fig. 2. Structure of reward model in stage one. The reward model is composed of the backbone and linear layer mapping features to a scalar r. Positive and negative sample pairs are taken as input. The model is trained to obtain higher r for positive samples than negative samples.

We define positive dataset $\mathcal{P}$ and negative dataset $\mathcal{N}$ as follows

$$\mathcal{P} = \{(\boldsymbol{X}, y) \in \mathcal{D} | y = 1\}, \quad (2)$$

$$\mathcal{N} = \{(\boldsymbol{X}, y) \in \mathcal{D} | y = 0\}, \quad (3)$$

In this study, the positive class refers to the minority class. For extremely imbalanced datasets, the number of samples belonging to the positive class ($|\mathcal{P}|$) is significantly smaller than the number of samples belonging to the negative class ($|\mathcal{N}|$). To quantify the degree of imbalance, the imbalance ratio (IR) is introduced, defined as

$$\text{IR} = \frac{|\mathcal{N}|}{|\mathcal{P}|}. \quad (4)$$

The imbalance ratio studied in this work exceeds 10^4, a magnitude considerably higher than most existing studies. Such extreme class imbalance can cause the classifier to exhibit significant bias toward the negative class, resulting in a high false positive rate. The probability predicted by the classification model that a sample $\boldsymbol{X}$ belongs to positive class serves as an estimation of criticality $\mathbb{P}(A \mid \boldsymbol{X})$.

III. METHODS

In this section, we propose a multi-stage training framework to progressively densify the dataset and alleviate the degree of imbalance, which consists of three stages. In the first stage, we utilize unsupervised learning model to pre-identify those non-critical negative samples. In this way, we can effectively reduce the imbalance ratio. In the second stage, labeled data is leveraged to train a classification model capable of further distinguishing between positive samples and hard negative samples. Finally, in the third stage, we employ dense reinforcement learning to improve the accuracy of criticality predicted by the classification model.

A. Stage One: Ranking Loss Based Unsupervised Learning

When confronted with a scarcity of positive samples, effectively capturing their distribution becomes a challenging task. In such scenarios, some approaches exploit anomaly detection to learn the distribution of negative samples and identify samples that deviate from this distribution as potential positive instances [30]. In this work, we leverage self-supervised learning to train a reward model. This model aims to increase the distance between positive and negative samples in the feature space. By doing so, we can identify samples that exhibit clear non-critical characteristics. Free from the influence of "label bias" [31], the reward model can learn more robust features.

Fig. 2 illustrates the architecture of reward model r_θ. During the training phase, the model takes pairs of positive and negative sample $(\boldsymbol{x}_{pi}, \boldsymbol{x}_{ni})$ as input. The objective of this reward model is to assign a scalar r to each input sample, with the purpose that positive samples get higher scores than negative ones. The loss function is defined as follows:

$$\mathcal{L}(\theta) = -\mathbb{E}_{\boldsymbol{X}_p \sim \mathcal{P}, \boldsymbol{X}_n \sim \mathcal{N}} \left[\log \sigma(r_\theta(\boldsymbol{X}_p) - r_\theta(\boldsymbol{X}_n)) \right], \quad (5)$$

where $\boldsymbol{X}_p$ represents the positive samples and $\boldsymbol{X}_n$ denotes the negative samples. σ denotes the sigmoid function. This loss function is widely utilized in the training of large language model (LLM) to obtain a reward model that scores the response of the LLM.

Our model essentially learns a mapping relationship between the feature space and the scalar space. By selecting appropriate threshold ϵ_1 and ϵ_2, we can effectively identify the "easy" negative samples or certainly non-critical samples, while retaining the positive samples. This means that a sample x is considered positive if $\epsilon_1 < r_\theta(x) < \epsilon_2$ and negative otherwise. Since our aim is to improve precision rate without compromising recall rate, the threshold ϵ_1 and ϵ_2 must guarantee that most positive samples are retained.

In addition, although performance of the reward model maybe affected by the extent of class overlap, employing unsupervised models allows us to substantially decrease the number of negative samples, thereby reducing the imbalance ratio of dataset.

B. Stage Two: Enhanced BBN based Supervised Learning

In the second stage of our framework, we employ supervised learning to train a classification model. It is worth noting that in this stage, our positive dataset $\mathcal{P}$ includes all positive samples utilized in the first stage. In contrast, our negative dataset $\mathcal{N}^-$ exclusively comprises false positive samples indentified in the first stage. We adopt a more complex model as the difficulty of samples increases compared to the first stage.

Inbalanced test dataset poses greater challenges for learned representation, and common rebalancing strategies such as re-sampling and re-weighting are often detrimental to feature learning. Intuitively, random sampling might lead to better representations. To address this issue, we develop an enhanced BBN method to improve the representation learning.
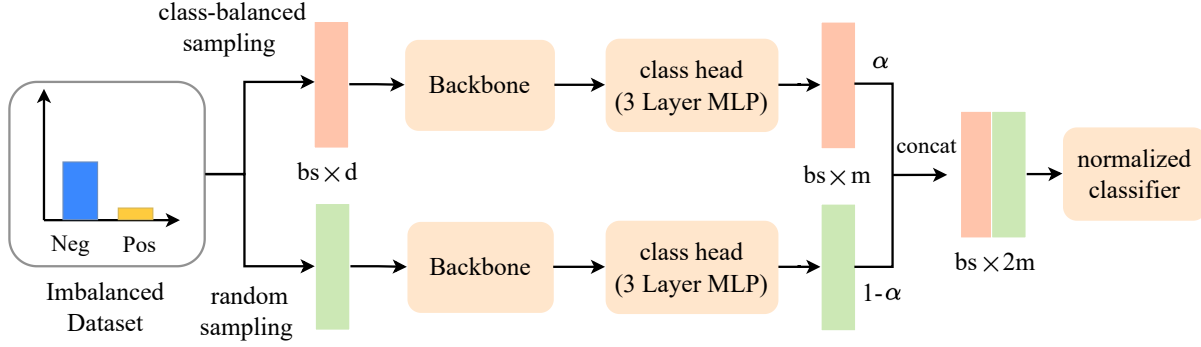
3245

Fig. 3. Structure of enhanced BBN in stage two. The upper branch focus on the representation learning of positive samples with class-balanced sampling and focal loss, while the lower branch focus on the representation learning of negative samples by uniform sampling and cross-entropy loss. Then features is mixed with adaptive parameter a. Finally a normalized classifier is adopted to mitigate the model's preference for negative samples.

As depicted in Fig. 3, in the upper branch, we utilize class-balanced sampling to focus on learning representations of positive samples, since inverse sampling strategy adopted in original BBN leads to high false positive rate on imbalanced test dataset. Simultaneously, in the lower branch, we employ random sampling to extract features from negative samples. The features from both branches are then combined as follows:

$$z = \alpha W_a^{\mathrm{T}} f_a + (1 - \alpha) W_b^{\mathrm{T}} f_b, \qquad (6)$$

where z represents the mixed feature representation, α is an adaptive hyper parameter controlling the mixing ratio and we also use the parabolic decay strategy [28]. W_a and W_b are the weight matrices for the respective branches, and f_a and f_b are the extracted features. Empirical observations suggest that the positive class tends to have larger classifier norms [32]. To mitigate the classifier's bias toward the majority class, we adopt a normalized linear classifier. This normalization ensures that the classifier's weights are scaled to have unit norm, effectively balancing the influence of different classes. The normalized weights are computed as

$$\hat{w}_i = \frac{w_i}{||w_i||^{\mathrm{T}}}, \qquad (7)$$

where $\hat{w}_i$ represents the normalized weight and w_i is the original weight. The loss function is given by:

$$\mathcal{L} = \alpha \mathcal{L}_a(p, y_a) + (1 - \alpha) \mathcal{L}_b(p, y_b), \qquad (8)$$

where p represents logits of classification model, and y_a and y_b are the corresponding labels for the two branches. $\mathcal{L}_a$ is the focal loss to further enhance feature learning of positive samples, and $\mathcal{L}_b$ is the cross-entropy loss. Using the enhanced BBN, we can further categorize samples that failed to be distinguished in the first stage. As the imbalance ratio decreases, the difficulty of training in this stage can be mitigated.

C. Stage Three: Dense Reinforcement Learning

Since our primary objective is to achieve precise criticality predictions, our focus in this stage shifts from continuously categorizing those hard or even non-separable samples to fine-tuning the classification model to further improve the accuracy of criticality assessments. Specifically, we employ dense reinforcement learning [29] to fine-tune the last layers of classification model in stage two, including the class head and classifier.

In reinforcement learning, the value of each state-action pair, denoted as $Q(s, a)$, represents the expected return obtained by executing action a in state s. To learn the Q-value, we utilize a dense deep Q-learning (DQN) approach, where the classifier serves as the Q-network and the predicted criticality represents the Q-value

$$Q(s, a) \triangleq \mathbb{P}(A|s, a) = \mathbb{P}(A|X). \qquad (9)$$

In practical applications, we introduce historical states to enrich the input information for improved classification accuracy. Consequently, the input X in previous stages contains not only s and a at current step but also previous states. For simplicity, we represent the input X as (s, a) in the rest of the paper. We adopt an offline training approach for efficiency.

The term "dense" refers to our use of critical episodes only for training Q-network. An episode, denoted as $(s_{i1}, s_{i2}, ..., s_{it})$, is considered critical if a critical event A occurs at s_{it}. According to our definition of train dataset $\mathcal{D}$, the state s_{it} is a positive sample and s_{ij} for $j = 1, ..., t - 1$, are negative samples. Additionally, states belonging to non-critical episodes are also considered as negative samples. This allows us to focus on the most relevant and informative data for fine-tuning the classifier. Our reward function is defined as follows

$$r(s) = \mathbb{I}_A(s), \qquad (10)$$

where $r = 1$ if at state s critical event happen and 0 otherwise. This reward function encourages the Q-network to assign higher Q-values to states that lead to critical outcomes. The Q-network is updated according to the following loss

function

$$\mathcal{L}(\boldsymbol{\theta}_k) = \sum_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\in\mathcal{D}} (y - Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}_k))^2 \mathbb{I}_{\boldsymbol{s}\in\mathbb{S}_c}, \qquad (11)$$

where $y = r + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{s}',\boldsymbol{a}';\boldsymbol{\theta}_{k-1})$ and $\mathbb{S}_c$ denotes the set of states belonging to critical episodes.

By employing dense DQN, we can optimize the criticality predicted by classifier. Due to the large of number of negative samples, the gradients are dominated by negative samples, leading to the classifier biased toward the negative class [33]. With dense DQN, we are able to fine-tune classification model with more balanced gradient. The derivative of the loss function is

$$\frac{\nabla\mathcal{L}(\boldsymbol{\theta}_k)}{\nabla\boldsymbol{\theta}_k} = -2 \sum_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\in\mathcal{D}} (y - Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}_k)) \frac{\nabla Q(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta}_k)}{\nabla\boldsymbol{\theta}_k} \mathbb{I}_{\boldsymbol{s}\in\mathbb{S}_c} \tag{12}$$

Then the loss function $\mathcal{L}(\boldsymbol{\theta}_k)$ can be reformulated as the sum of the positive class loss function $\mathcal{L}_p(\boldsymbol{\theta}_k)$ and the negative class loss function $\mathcal{L}_n(\boldsymbol{\theta}_k)$, we have

$$\begin{aligned}
\frac{\nabla\mathcal{L}(\boldsymbol{\theta}_k)}{\nabla\boldsymbol{\theta}_k} &= \frac{\nabla\mathcal{L}_p(\boldsymbol{\theta}_k)}{\nabla\boldsymbol{\theta}_k} + \frac{\nabla\mathcal{L}_n(\boldsymbol{\theta}_k)}{\nabla\boldsymbol{\theta}_k} \\
&= 2\sum_{i=1}^{N_p} (f_{\boldsymbol{\theta}_k}(\boldsymbol{s}_i,\boldsymbol{a}_i) - 1)\frac{\nabla f_{\boldsymbol{\theta}_k}}{\nabla\boldsymbol{\theta}_k} + \\
&\quad 2\sum_{j=1}^{N'_n} (f_{\boldsymbol{\theta}_k}(\boldsymbol{s}_j,\boldsymbol{a}_j) - \gamma\max_{\boldsymbol{a}'_j} f_{\boldsymbol{\theta}_k}(\boldsymbol{s}'_j,\boldsymbol{a}'_j))\frac{\nabla f_{\boldsymbol{\theta}_k}}{\nabla\boldsymbol{\theta}_k}
\end{aligned} \tag{13}$$

where $f_{\boldsymbol{\theta}_k}$ is the classification model. We can observed that N_p equals $|\mathcal{P}|$ and the number of negative samples $N'_n \ll |\mathcal{N}|$ due to the dense method. Therefore, in comparison to the second stage, we further alleviate the impact of negative samples on the loss function. This helps in reducing the classifier's preference for negative samples and improving the precision of criticality.

IV. RESULTS

In this section, we applied our approach to two well-known cases: Bipedal Walker and Lunar Lander, both of which are tasks under the Gym environment. Next, we provide a brief overview of each case and present the corresponding ROC curves and Precision-Recall curves. Eventually, ablation studies are performed to verify the effectiveness of each components of our method.
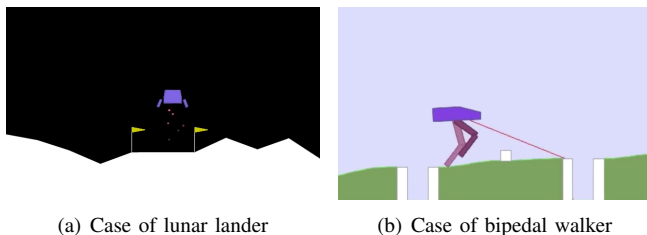

(a) Case of lunar lander   (b) Case of bipedal walker

Fig. 4. Overview of cases

A. Overview of Cases

In the lunar lander case, as shown in Fig. 4(a), a rocket is optimized to achieve a safe touchdown on the landing pad, avoiding any potential crashes. The criticality event A occurs when the lander gets in contact with the moon. A well-trained rocket using PPO algorithm was utilized to collect the training dataset. Considering information related to the rocket's state and atmospheric conditions, the input feature X_k is defined as

$$\boldsymbol{X}_k = (\boldsymbol{s}_{k-10}, .., \boldsymbol{s}_{k-1}, \boldsymbol{s}_k), \tag{14}$$

where $\boldsymbol{s}_k \in \mathbb{R}^{11}$ represents the state at step k, and $\boldsymbol{s}_{k-10}, ..., \boldsymbol{s}_{k-1}$ represent historical states. The state $\boldsymbol{s}_k$ consists of both the rocket's state $\boldsymbol{s}_{kr}$ and the environment's state $\boldsymbol{s}_{ke}$, where $\boldsymbol{s}_{kr}$ includes the rocket's coordinates, linear velocities, angles, angular velocity, along with two boolean values indicating whether each leg is in contact with the ground. Additionally, $\boldsymbol{s}_{kr}$ includes the force of linear wind and rotational wind. If critical event occurs at terrain $\boldsymbol{s}_k$, the sample $\boldsymbol{x}_k$ is labeled as a positive sample ($y_i = 1$). The original train and test dataset exhibit an imbalanced ratio of 1.26×10^4.

In the bipedal walker case, as shown in Fig. 4(b), a four joint walker must navigate through a series of obstacles, including stumps and pitfalls, to reach its destination. The criticality event A is defined as the hull of walker contact with the ground without achieving a predefined distance. Utilizing the POET [34] algorithm, a well-trained walker was developed. The input feature $\boldsymbol{X}_k$ is

$$\boldsymbol{X}_k = (\boldsymbol{s}_{k0}, \boldsymbol{s}_{k1}, \boldsymbol{s}_{k2}, ..., \boldsymbol{s}_{k9}), \tag{15}$$

where $\boldsymbol{s}_{k0} \in \mathbb{R}^{25}$ represents the walker's state, including hull angle speed, angular velocity, horizontal speed, vertical speed, position of joints and joints angular speed, legs contact with ground, and lidar measurements. $\boldsymbol{s}_{ki} \in \mathcal{R}^8$ represents terrain within the detection range of lidar. If the critical event occurs at terrain $\boldsymbol{s}_{k9}$, the sample $\boldsymbol{X}_k$ is considered as positive sample with $y_k = 1$. The imbalanced ratio of train and test dataset approaches 1.56×10^4.

B. Results

For the Lunar Lander case, in the first stage, a 6-layer transformer serves as the backbone of our reward model. As depicted in Fig. 5(a), orange points represent results of positive samples, while blue points represent results of negative ones. The dashed lines indicate the thresholds we select. We manage to delete 95.02% negative samples and identify 99.27% positive samples with $\epsilon_1 = 5$ and $\epsilon_2 = +\infty$. The backbone used in the second stage is identical to that of reward model. Using both the reward model and classification model, we can achieve improved identification rates, 98.41% for negative samples and 99.13% for positive samples, as well as the AUC of 0.9612. Finally, dense DQN leads to a further improvement: 99.06% for negative samples and 99.18% for positive samples, resulting in AUC increase to 0.9853.
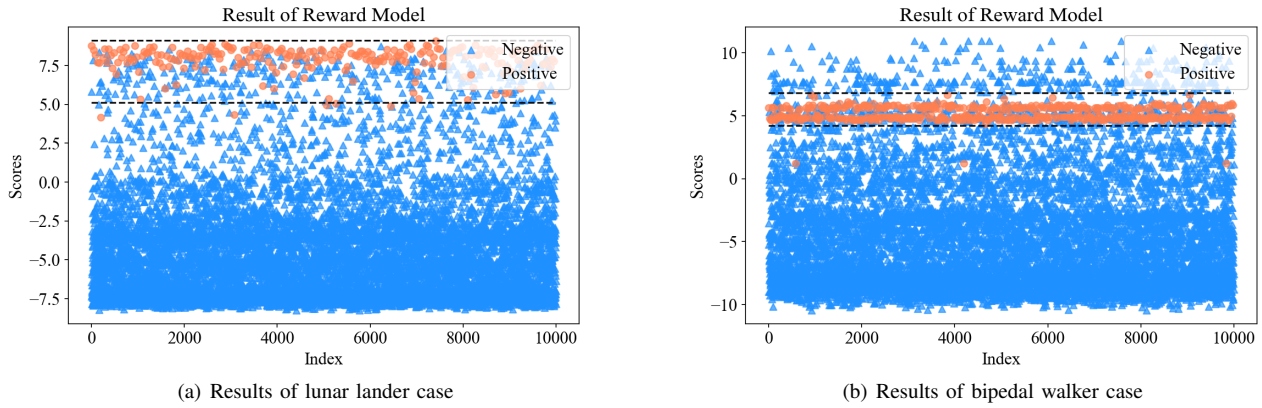
(a) Results of lunar lander case

(b) Results of bipedal walker case

Fig. 5.   Outputs of reward model

(a) ROC curve of lunar lander case

(b) ROC curve of bipedal walker case

(c) RR curve of lunar lander case

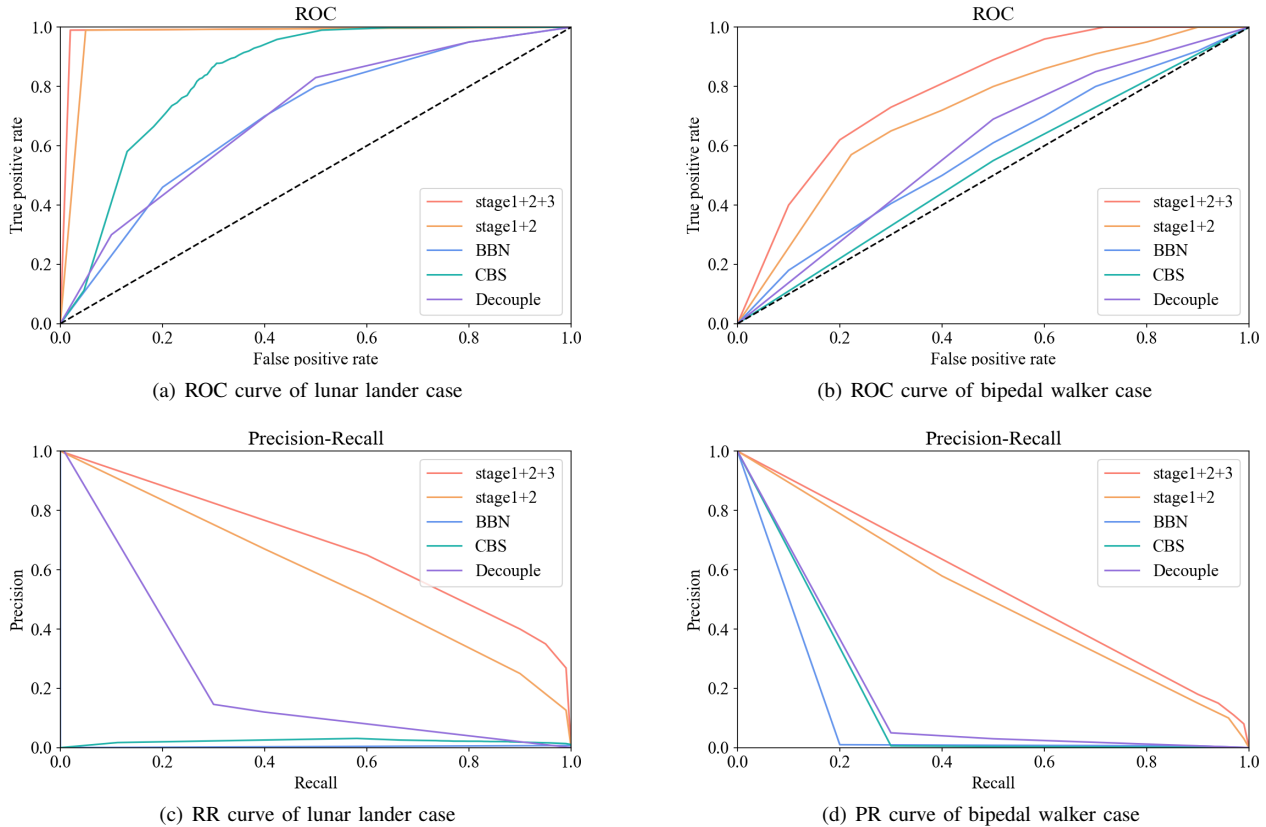(d) PR curve of bipedal walker case

Fig. 6.   ROC curves and Precision-Recall curves of supervised classification model and dense DQN fine-tuned model

For the bipedal walker case, in the first stage, a 6-layer transformer serves as backbone of our reward model. As shown in Fig. 5(b), by selecting the thresholds of 4.2 and 6.8, we can eliminate 79.35% negative samples and identify 87.60% positive samples. Then, benefiting from classification model, we can identify 88.72% negative samples and 90.12% positive samples and the AUC is 0.6752. By employing dense DQN, we can identify 91.66% negative samples and 93.09% positive samples and the AUC is 0.7163.

As illustrated in Fig. 6, we compared our method with traditional approaches including original BBN [28], class-balanced sampling (CBS) [18] and decoupling method [22]. It is evident that those classical methods mentioned above may not be effective when dealing with extremely imbal-

ance test dataset. Additionally, removing certain non-critical samples with unsupervised learning stage proved beneficial in improving identification rates. Furthermore, it can also be demonstrated that performance gains further improvement with dense DQN.

V. CONCLUSIONS

In conclusion, our study focuses on developing a criticality prediction model for intelligent systems that achieved both high precision and recall rates. To address the challenge of curse of rarity, we propose a multi-stage learning framework, which gradually densifies the dataset and alleviates the degree of imbalance through a combination of unsupervised learning and reinforcement learning methods. Our research

provides valuable insights into the criticality prediction of intelligent systems, thus facilitating their development and real-world deployment. In the future, we will delve deeper into the theoretical underpinnings of our proposed method and determine the imbalance ratio for which it is applicable.

REFERENCES

[1] J. P. Cerrolaza, R. Obermaisser, J. Abella, F. J. Cazorla, K. Grüttner, I. Agirre, H. Ahmadian, and I. Allende, "Multi-core devices for safety-critical systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–38, 2020.

[2] M. Cummings and D. Britton, "Regulating safety-critical autonomous systems: past, present, and future perspectives," in *Living with robots*. Elsevier, 2020, pp. 119–140.

[3] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robotics and Autonomous Systems*, vol. 94, pp. 43–52, 2017.

[4] L. Westhofen, C. Neurohr, T. Koopmann, M. Butz, B. Schütt, F. Utesch, B. Neurohr, C. Gutenkunst, and E. Böde, "Criticality metrics for automated driving: A review and suitability analysis of the state of the art," *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 1–35, 2023.

[5] S. Feng, Y. Feng, C. Yu, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part i: Methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1573–1582, 2021.

[6] S. Feng, Y. Feng, H. Sun, S. Bao, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part ii: Case studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5635–5647, 2021.

[7] K. Cao, Y. Xia, J. Yao, X. Han, L. Lambert, T. Zhang, W. Tang, G. Jin, H. Jiang, X. Fang *et al.*, "Large-scale pancreatic cancer detection via non-contrast ct and deep learning," *Nature medicine*, vol. 29, no. 12, pp. 3033–3043, 2023.

[8] I. J. Ratul, A. Al-Monsur, B. Tabassum, A. M. Ar-Rafi, M. M. Nishat, and F. Faisal, "Early risk prediction of cervical cancer: A machine learning approach," in *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2022, pp. 1–4.

[9] L. Jena, B. Patra, S. Nayak, S. Mishra, and S. Tripathy, "Risk prediction of kidney disease using machine learning strategies," in *Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 2*. Springer, 2021, pp. 485–494.

[10] H. X. Liu and S. Feng, "" curse of rarity" for autonomous vehicles," *arXiv preprint arXiv:2207.02749*, 2022.

[11] S. Ando and C. Y. Huang, "Deep over-sampling framework for classifying imbalanced data," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*. Springer, 2017, pp. 770–785.

[12] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen *et al.*, "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2018, pp. 112–117.

[13] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. Springer, 2016, pp. 467–482.

[14] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks*, vol. 106, pp. 249–259, 2018.

[15] H. Lee, M. Park, and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3713–3717.

[16] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?" in *International conference on machine learning*. PMLR, 2019, pp. 872–881.

[17] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *Advances in neural information processing systems*, vol. 32, 2019.

[18] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.

[19] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1367–1381, 2018.

[20] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5375–5384.

[21] S. Khan, M. Hayat, S. W. Zamir, J. Shen, and L. Shao, "Striking the right balance with uncertainty," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 103–112.

[22] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," *arXiv preprint arXiv:1910.09217*, 2019.

[23] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7610–7619.

[24] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," *Advances in neural information processing systems*, vol. 32, 2019.

[25] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2537–2546.

[26] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Feature transfer learning for face recognition with under-represented data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5704–5713.

[27] J. Shi, J. Zheng, S. Yin, Y. Lu, Y. Xie, and Y. Qu, "Clip-guided federated learning on heterogeneous and long-tailed data," *arXiv preprint arXiv:2312.08648*, 2023.

[28] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9719–9728.

[29] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.

[30] Y. Xu, Y.-L. Li, J. Li, and C. Lu, "Constructing balance from imbalance for long-tailed image recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 38–56.

[31] H. Liu, J. Z. HaoChen, A. Gaidon, and T. Ma, "Self-supervised learning is more robust to dataset imbalance," *arXiv preprint arXiv:2110.05025*, 2021.

[32] C. Thrampoulidis, G. R. Kini, V. Vakilian, and T. Behnia, "Imbalance trouble: Revisiting neural-collapse geometry," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27225–27238, 2022.

[33] M. Molahasani, A. Etemad, and M. Greenspan, "Continual learning for long-tailed recognition: Bridging the gap in theory and practice," in *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

[34] R. Wang, J. Lehman, J. Clune, and K. O. Stanley, "Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions," *arXiv preprint arXiv:1901.01753*, 2019.