SAFE: Safety Analysis and Retraining of DNNs

Mohammed Oualid Attaoui
SnT Centre, University of Luxembourg
Luxembourg, Luxembourg
mohammed.attaoui@uni.lu

Fabrizio Pastore
SnT Centre, University of Luxembourg
Luxembourg, Luxembourg
fabrizio.pastore@uni.lu

Lionel Briand
SnT Centre, University of Luxembourg
School of EECS, University of Ottawa
Ottawa, Canada
lbriand@uottawa.ca

ABSTRACT

We present SAFE, a tool based on a black-box approach to automatically characterize the root causes of Deep Neural Network (DNN) failures. SAFE relies on VGGNet-16, a transfer learning model pre-trained on ImageNet, to extract the features from error-inducing images. After feature extraction, SAFE applies a density-based clustering algorithm to discover arbitrarily shaped clusters of images modeling plausible causes of failures. By relying on the identified clusters, SAFE can select a set of additional images to be used to retrain and improve the DNN efficiently. Empirical results show the potential of SAFE in identifying different root causes of DNN failures based on case studies in the automotive domain. It also yields significant improvements in DNN accuracy after retraining while saving considerable execution time and memory compared to alternatives. A demo video of *SAFE* is available at https://youtu.be/8QD-PPFTZxs.

CCS CONCEPTS

• **Software and its engineering → Software verification and validation**.

KEYWORDS

DNN explanation, Functional Safety Analysis, DNN Debugging

ACM Reference Format:
Mohammed Oualid Attaoui, Fabrizio Pastore, and Lionel Briand. 2024. SAFE: Safety Analysis and Retraining of DNNs. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3639478.3640028

1 INTRODUCTION

Deep Neural Networks (DNNs) have achieved high performance in many contexts where perception and decision-making tasks play critical roles: networked surveillance [24], medical imaging [4], and autonomous vehicles [23]. A good example of the latter is IEE [10], our industry partner in this research, which is extending its portfolio of in-vehicle monitoring systems with DNN-based products. A DNN can only be used in a safety-critical system if it is deemed trustworthy. Explanation methods aim at making DNN decisions analysable thus fostering trust [2, 8]. Indeed, explanation methods usually provide a visual aid accompanying a prediction to provide insights into the underlying reasons for the model output. One of their applications is root cause analysis, that is, identifying the source (root cause) of a DNN failure (e.g., an incorrect prediction). Root cause analysis is part of mandatory safety analysis procedures for safety-critical systems such as in the automotive domain [11].

Well known DNN explanation methods such as SHAP [13], LIME [17], Anchors [18] and Mean-Centroid PredDiff [12], when applied to image-processing DNNs, identify the image pixels that influence the DNN output. They determine relevant pixels by analyzing how DNN predictions vary when pixel values are modified; their main benefit is that they are black-box: they can be applied to any DNN without changes in its implementation. Approaches such as Grad-CAM [20], and Layer-Wise Relevance Propagation (LRP) [14], instead, are white-box since they identify relevant pixels by integrating heuristics to propagate the DNN prediction scores backward through the internal DNN layers; unfortunately, they require the modification of the DNN under analysis, which is expensive and error prone. Further, both these white-box and black-box approaches present a common limitation: although useful to explain a single classification, they do not help engineers in the presence of several images leading to DNN failures. Indeed, these approaches require engineers to inspect each failure individually and are thus not supporting the main purpose of root cause analysis, which is to determine if there are scenarios in which the DNN tends to fail, in order to assess risks and possibly identify countermeasures.

We have addressed the above limitations in previous work (HUDD) to better support safety analysis [6, 7]. Given a set of images, all leading to DNN failures, it groups images into clusters; the commonalities observed across images in a particular cluster capture the root cause of their mispredictions. A root cause characterizes the scenario in which the DNN is likely to fail; for example, drowsiness may not be detected by the DNN when the car driver wears sunglasses. HUDD relies on clustering algorithms relying on distance values computed on heatmaps; a heatmap is a matrix that is generated using the LPR method and captures the relevance of an input pixel (or neuron) to the DNN decision. Unfortunately, heatmap-based distance is expensive to compute and can be affected by noise. Further, it requires the DNN model to be extended with LRP features [14].

Other related work includes the use of attention maps to identify low-confidence DNN outputs, which is a problem orthogonal to ours [22]. DeepHyperion [25, 26] relies on illumination search to derive probability maps characterizing failing inputs; similar to work combining search with decision tree algorithms [1], and different from SAFE, it can be applied only when inputs are generated through a simulator. Further, explanation techniques based on either feature maps or heatmaps have shown poor performance in

ICSE-Companion '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0502-1/24/04.
https://doi.org/10.1145/3639478.3640028

empirical studies with image classifiers [27].

In this paper, we introduce *SAFE*, the tool that implements our black-box automated approach for root cause analysis [3]. *SAFE* relies on feature extraction based on transfer-learning and density-based clustering. Transfer learning (i.e., relying on a pre-trained VGGNet-16 [21]) is used as a black-box approach to extract features from images and generate clusters, and enables *SAFE* to overcome HUDD's limitations (i.e., it does not require the modification of the DNN under analysis and is robust to noise). Density-based clustering, instead, forms non-convex clusters of images that capture different DNN failures' root causes (hereafter, root cause clusters — RCCs). For the retraining of the DNN, *SAFE* relies on the identified RCCs to select an unsafe image set used to improve the accuracy of the DNN. Further, *SAFE* comes with a GUI that facilitates its usage, which was not available for HUDD.

We conducted an empirical evaluation of SAFE on six DNNs [3]. Our empirical results show the cost-effectiveness of *SAFE* in identifying plausible root causes with a reasonable human effort and its efficiency in memory usage and computation time. *SAFE* also achieved significant improvements in the retraining of DNNs (up to 35% improvement over the original models) and overall better results than alternatives.

The paper proceeds as follows. Section 2 describes our approach *SAFE*. Section 3 details the tool's architecture. Section 4 summarizes our empirical results. Section 5 concludes the paper.

2 THE SAFE METHODOLOGY

SAFE performs safety analysis and retraining in six steps depicted in Figure 1. In *Step 1*, SAFE relies on the features extracted by the pre-trained model to generate the root cause clusters. This step consists of four activities listed in the following:

- *Data acquisition and preprocessing:* SAFE downsamples the size of the image to match VGGNet-16's input size requirement (224×224 pixels).
- *Features extraction:* This activity aims to transform unstructured data into structured data for their exploitation by clustering algorithms [9]. SAFE relies on a transfer learning-based feature extraction method that leverages the VGGNet-16 model pre-trained on the ImageNet database.
- *Dimensionality reduction:* Dimensionality reduction aims at approximating data in high-dimensional vector spaces. SAFE relies on Principal Component Analysis (PCA) [15] to reduce the dimensions of the features from 4096 to 256.
- *Clustering:* SAFE relies on the DBSCAN clustering algorithm [5] to generate RCCs. SAFE relies on the elbow method [16] and the Silhouette index [19] to automatically configure DBSCAN's hyper-parameters.

Step 2 involves a visual inspection of the images belonging to each RCC. For visual inspection, the engineer may either visualize a random subset of the images in the RCC (five are sufficient, based on our empirical study [3]) or visualize an animated gif with all the images (see Section 3). This activity is key for safety analysis[11]; indeed, by identifying the commonalities across the images in the same RCC, the engineer can determine in which scenarios the DNN may fail. For example, Fig. 2 shows a subset of the images belonging to two RCCs for a DNN that classifies headpose (top-left, top-center, top-right, etc.). RCC 1 shows that failures may occur when the person is looking left (e.g., because one eye is not visible), and RCC 2 shows that failures may occur if the person is looking between middle-left and top-left. Based on domain knowledge, the engineer can determine the likelihood of those scenarios and their impact on the system (e.g., if the failure may lead to a hazard). Consequently, he can decide if the system is adequate for production or if improvement is needed. Improvement might be achieved through further retraining (e.g., to make the DNN robust to RCC 1) or countermeasures (e.g., using two cameras to always capture two eyes).

In *Step 3*, the engineer provides a new set of images (*improvement set*) from which we select the images for model retraining.

In *Step 4*, *SAFE* relies on the generated RCCs to select a subset of images from the improvement set called the *unsafe set*. The unsafe set consists of images in the improvement set that belong to an RCC; they are the images that are closer to any core point of the RCCs (core points are cluster members identified by DBSCAN that approximate the cluster shape). *SAFE* selects a number of images that is proportional to the test set accuracy; the selected images are uniformly distributed across RCCs. In *Step 5*, the engineer labels the images in the unsafe set, if needed (e.g., it is unnecessary for images generated with simulators).

Finally, in *Step 6*, *SAFE* automatically retrains the model to enhance its prediction accuracy.

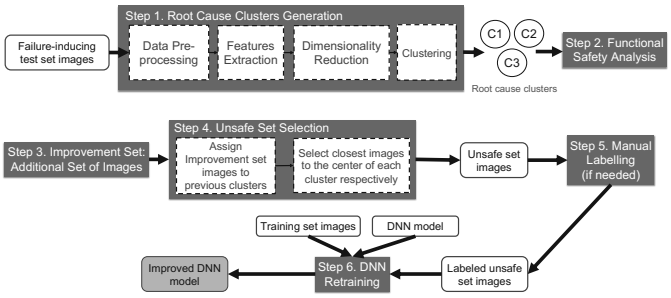

Figure 1: Overview of *SAFE*

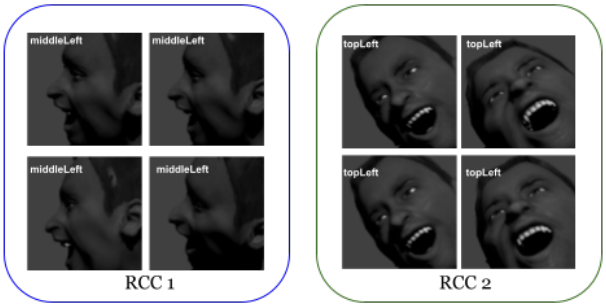

Figure 2: Example of RCCs generated by *SAFE*. The ground truth is shown in the top left of each image.

3 TOOL ARCHITECTURE

We implemented *SAFE* with Python. We use the Keras library[1] to load the VGG16 pre-trained model and to pre-process the images. We rely on the DBSCAN clustering algorithm and the PCA dimensionality reduction method from the Scikit-Learn library[2]. To manipulate images, we opt for the Pillow library[3]. The NumPy library[4] is used to pre-process the images. The *SAFE* tool is developed with a Graphical User Interface (GUI) using Gradio[5].

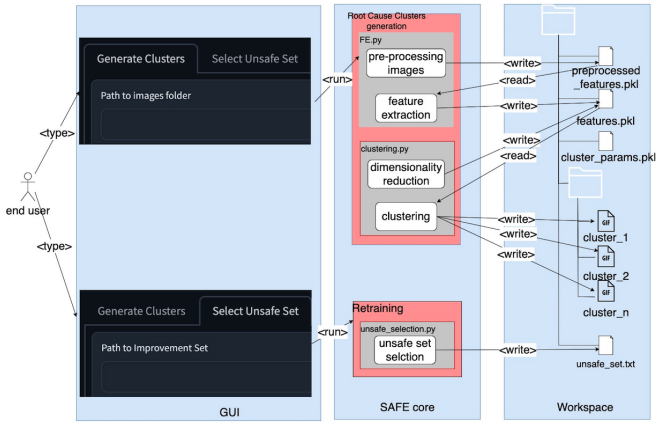

Figure 3: *SAFE* architecture and outputs

Figure 3 depicts the architecture of the *SAFE* tool. The *SAFE* tool consists of a GUI and a set of components (Python modules) working in the background.

The feature extraction component (it is part of the module *FE.py*) takes as input the set of images and pre-processes them. Then, it applies the VGG16 model to extract the features. These features are processed by the module *clustering.py*, which first applies the PCA method to reduce the dimensions to 256 features and then applies the DBSCAN clustering algorithm, after automatically choosing the optimal values for the DBSCAN parameters.

After running the clustering algorithm, the clustering component generates a text file (*labels.txt*) that associates a clustering ID to each processed image. For visual inspection, the images belonging to each RCC are saved in the same animated gif image. These images are automatically saved in the *gifs/* folder and shown in the GUI.

To generate the RCCs, the user starts the SAFE tool by executing *SAFE_Tool.py*. The GUI is started as a Web service whose Web interface can be opened in a browser. The Web interface enables the end-user to provide the path to the folder containing the images. RCCs are automatically generated with a single click on a dedicated button in the GUI, and no further configuration is needed. The generated clusters are shown in the result pane (see Fig. 4) and one can browse the generated clusters on the bottom bar where one can click on any of them for visualization in the pane. Thanks to its Web interface and its black-box nature requiring only failure-inducing images as input, the safety analysis feature of SAFE might be provided in the future as an online service for DNN specialists.

The unsafe set selection is performed using the same interface by switching to the dedicated tab. A new interface will appear where the end-user can provide the link to the improvement set images. The tool will then select a set of images called the unsafe set to be used for retraining. The list and number of selected images are displayed in the interface and saved in a file called *IS_filenames.txt*.

The *SAFE* tool is available online[6].

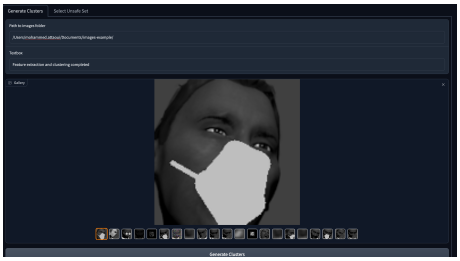

Figure 4: RCC gifs visualized in SAFE's GUI.

4 EMPIRICAL EVALUATION

We have applied SAFE to four DNNs that support gaze detection (GD), drowsiness detection (OC), headpose detection (HPD), and face landmarks detection systems (FLD) investigated with IEE Sensing, our industry partner. We also considered additional DNNs trained using real-world images that target traffic sign recognition (TS) and object detection (OD) and are typical features in automotive, DNN-based systems. Below, we summarize the results obtained when assessing our research questions.

RQ1: Is the number of generated clusters small enough to enable visual inspection? Our results show that *SAFE* yields an acceptable number of clusters for visual inspection. Without SAFE, an engineer needs to visualize all the failure-inducing images. Under the assumption that the end-users inspect only five images to determine commonalities in a RCC, the ratios of error-inducing images to be inspected are low (between 1.07 and 26.15, with a median of 5.12). Thus, using *SAFE* can save significant effort compared to the manual inspection of the entire set of images.

RQ2: Does *SAFE* generate root cause clusters with a significant reduction in variance for simulator parameters? Since we relied on DNNs trained and tested with simulators, we could determine if images assigned to the same cluster present similar values for a subset of the simulator parameters (setting, or configuration option within a simulation that can be adjusted or configured to control various aspects of the simulation); indeed, similar parameter values should indicate that the images in an RCC are visually similar and, therefore, enable the engineer to identify commonalities across them. Precisely, the parameters of the images in a cluster should present a reduction in the variance of some parameters with respect to those computed on the entire test set.

The clusters generated by *SAFE* show a significant variance

[1]https://github.com/keras-team/keras
[2]https://scikit-learn.org/stable/
[3]https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf
[4]https://numpy.org
[5]https://github.com/gradio-app/gradio/
[6]https://doi.org/10.5281/zenodo.10014621

reduction. *SAFE* yields a median of 100% of the clusters with a variance reduction above 90% for GD and OC (compared to 86% for GD and 57% for OC by using HUDD), and 90% for FLD and HPD. These results are due to the nature of the clusters generated by *SAFE*. Indeed, *SAFE* can find RCCs with arbitrary shapes.

RQ3: Does *SAFE* identify more distinct error root causes than HUDD? Under the assumption that DNN failures occur in specific areas of the simulator parameter space, we identify a set of unsafe parameters and corresponding unsafe values around which a DNN error is more likely to occur. For example, for the Gaze Angle parameter, unsafe values consist of the boundary values distinguishing labels about different gaze directions.

Since our simulators generate images with uniformly sampled parameter values within the input domain, every unsafe value has the same likelihood of being observed in the test set. Therefore, we aim for the root cause clusters to cover all such values.

The clusters generated by *SAFE* with GD, OC, and HPD cover (median) 92%, 64%, and 80% of the unsafe values, respectively (compared to 71%, 50%, and 60% by using HUDD). In addition, we report that 90% of the clusters cover a unique set of unsafe values (e.g., *Angle=337.5, H-Headpose=220, V-Headpose=340, Distance=25*). For the remaining 10%, we observe that they are still unique but differ concerning a parameter that is not unsafe. Therefore, we conclude that all the clusters generated by *SAFE* cover a unique set of parameter values that are useful to determine distinct failure causes.

RQ4: Does *SAFE* provide time and memory savings compared to the white-box approach? We investigate the execution time and the memory allocation savings provided by *SAFE* and HUDD when generating the root cause clusters. The time required to generate the RCCs for *SAFE* varies between 2.5 and 3.5 minutes; HUDD requires 16 to 65 minutes to perform the same task. Such execution time savings allow engineers to conduct DNN safety analysis and improvements in a much shorter time. Memory savings also have significant implications since they prevent the use of expensive hardware from performing such analysis. *SAFE* requires from 0.82 to 74.2Mb of memory allocation, while HUDD requires from 3551 to 78641Mb, mainly because of the memory required to store heatmaps.

5 CONCLUSION

We introduced *SAFE*, a tool based on a black-box approach to identify plausible causes of DNN mispredictions without requiring any modification to the DNN or access to its internal information.

SAFE combines transfer learning-based feature extraction and a density-based clustering algorithm to generate arbitrary-shaped clusters representing the root causes of DNN mispredictions.

Empirical results show that *SAFE* derives root cause clusters that can effectively help engineers determine the root causes for DNN mispredictions. The root cause clusters include images with similar characteristics that are likely related to mispredictions. Further, for our case study subjects, the generated clusters appear to capture all possible misprediction causes. Beyond these benefits, *SAFE* helps save large amounts of execution time and memory compared to HUDD, and alternative white-box approach.

ACKNOWLEDGMENTS

This project has received funding from the ESA discovery contract I-2022-02236 (TIA - Test Improve Assure, Activity No. 1000035943), IEE Luxembourg, Luxembourg's National Research Fund (FNR) under grant BRIDGES2020/IS/14711346/FUNTASY, and NSERC of Canada under the Discovery and CRC programs.

REFERENCES

[1] Raja Ben Abdessalem, Shiva Nejati, Lionel C Briand, and Thomas Stifter. 2018. Testing vision-based control systems using learnable evolutionary algorithms. In *2018 IEEE/ACM-ICSE 22*. IEEE, 1016–1026.
[2] Rob Ashmore, Radu Calinescu, and Colin Paterson. 2021. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. *ACM Comput. Surv.* 54, 5, Article 111 (may 2021), 39 pages. https://doi.org/10.1145/3453444
[3] Mohammed Oualid Attaoui, Hazem Fahmy, Fabrizio Pastore, and Lionel Briand. 2022. Black-box Safety Analysis and Retraining of DNNs based on Feature Extraction and Clustering. *ACM TOSEM* (2022).
[4] Nassima Dif, Mohammed Oualid Attaoui, Zakaria Elberrichi, Mustapha Lebbah, and Hanene Azzag. 2021. Transfer learning from synthetic labels for histopathological images classification. *Applied Intelligence* (2021), 1–20.
[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD'96* (Portland, Oregon). AAAI Press, 226–231.
[6] Hazem Fahmy, Fabrizio Pastore, Mojtaba Bagherzadeh, and Lionel Briand. 2021. Supporting Deep Neural Network Safety Analysis and Retraining Through Heatmap-Based Unsupervised Learning. *IEEE Transactions on Reliability* (2021). https://doi.org/10.1109/TR.2021.3074750
[7] Hazem Fahmy, Fabrizio Pastore, and Lionel Briand. 2022. HUDD: A Tool to Debug DNNs for Safety Analysis. In *Proceedings of 44th ICSE* (Pittsburgh, Pennsylvania). 100–104.
[8] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE-DSAA*. IEEE, 80–89.
[9] Gábor Gosztolya, Róbert Busa-Fekete, Tamás Grósz, and László Tóth. 2017. DNN-Based Feature Extraction and Classifier Combination for Child-Directed Speech, Cold and Snoring Identification. In *Interspeech*. ISCA.
[10] IEE. 2020. IEE Sensing solutions. www.iee.lu.
[11] International Organization for Standardization. 2020. ISO/PAS 21448:2019, Road vehicles: Safety of the intended functionality.
[12] Ding Li, Yan Liu, Jun Huang, and Zerui Wang. 2023. A Trustworthy View on Explainable Artificial Intelligence Method Evaluation. *Computer* 56, 4 (2023), 50–60. https://doi.org/10.1109/MC.2022.3233806
[13] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.
[14] Gregoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus Robert Müller. 2019. *Layer-Wise Relevance Propagation: An Overview*. Springer.
[15] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2, 11 (1901), 559–572.
[16] Nadia Rahmah and Imas Sukaesih Sitanggang. 2016. Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. In *IOP conference series: earth and environmental science*, Vol. 31. IOP Publishing, 012012.
[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 1135–1144.
[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.
[19] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.
[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE ICCV*. 618–626.
[21] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[22] Andrea Stocco, Paulo J Nunes, Marcelo D'Amorim, and Paolo Tonella. 2022. Thirdeye: Attention maps for safe autonomous driving systems. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*.

1–12.

[23] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th ICSE*. 303–314.

[24] G Vallathan, A John, Chandrasegar Thirumalai, SenthilKumar Mohan, Gautam Srivastava, and Jerry Chun-Wei Lin. 2021. Suspicious activity detection using deep learning in secure assisted living IoT environments. *The Journal of Supercomputing* 77, 4 (2021), 3242–3260.

[25] Tahereh Zohdinasab, Vincenzo Riccio, Alessio Gambi, and Paolo Tonella. 2021. Deephyperion: exploring the feature space of deep learning-based systems through illumination search. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 79–90.

[26] Tahereh Zohdinasab, Vincenzo Riccio, Alessio Gambi, and Paolo Tonella. 2023. Efficient and effective feature space exploration for testing deep learning systems. *ACM Transactions on Software Engineering and Methodology* 32, 2 (2023), 1–38.

[27] Tahereh Zohdinasab, Vincenzo Riccio, and Paolo Tonella. 2023. An Empirical Study on Low-and High-Level Explanations of Deep Learning Misbehaviours. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–11.