[Array 15 (2022) 100201](https://doi.org/10.1016/j.array.2022.100201)

|  |  |  |
| --- | --- | --- |
|  | Contents lists available at [ScienceDirect](https://www.elsevier.com/locate/array) |  |
| Array |
| journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array) |
|  | | |

Node co-activations as a means of error detection—Towards fault-tolerant neural networks   
Lalli Myllyahoa,∗, Jukka K. Nurminena, Tommi Mikkonenb   
a *University of Helsinki, Finland*   
b *University of Jyväskylä, Finland*

|  |  |  |
| --- | --- | --- |
| A R T I C L E | I N F O | A B S T R A C T |
| *Keywords:*  Machine learning Fault tolerance  Neural networks Error detection  Concept drift  Dependability |  | **Context:** Machine learning has proved an efficient tool, but the systems need tools to mitigate risks during runtime. One approach is fault tolerance: detecting and handling errors before they cause harm.  **Objective:** This paper investigates whether rare co-activations – pairs of usually segregated nodes activating together – are indicative of problems in neural networks (NN). These could be used to detect concept drift and flagging untrustworthy predictions.  **Method:** We trained four NNs. For each, we studied how often each pair of nodes activates together. In a separate test set, we counted how many rare co-activations occurred with each input, and grouped the inputs based on whether its classification was correct, incorrect, or whether its class was absent during training. **Results:** Rare co-activations are much more common in inputs from a class that was absent during training. Incorrectly classified inputs averaged a larger number of rare co-activations than correctly classified inputs, but the difference was smaller.  **Conclusions:** As rare co-activations are more common in unprecedented inputs, they show potential for detecting concept drift. There is also some potential in detecting single inputs from untrained classes. The small difference between correctly and incorrectly predicted inputs is less promising and needs further research. |

**1. Introduction**

Machine learning (ML) models are statistical approximations, whim-sical and capricious in nature, and often made for environments that evolve over time. In such approximations, a 99% accurate model –something that is practically always correct – is wrong 1% of the time. What should be done if that 1% happens and causes errors in your system? Are there ways to mitigate the risk and prepare a software system for the inevitable ‘‘bad days’’ of your model? The trustworthi-ness of ML systems has been improved, for example, by establishing patterns for fault tolerance, but tools for measuring whether a model’s results are and remain trustworthy can still be improved [1]. Further-more, such detection should ideally not only be an afterthought, but detection should occur in real time while the model is running. During computation runs, one approach to mitigating the risk and making the system more fault-tolerant could be monitoring the model’s own inner structure.

The inner structure of a neural network – a currently common ML technique – is sometimes compared to the structure of biological neural circuitry (e.g. Abiodun et al. [2]). Like biological neural circuitry and

neurons, neural networks in computing consist of layers of intercon-nected nodes. These nodes are tiny computational units that receive an input, and either activate and pass on an output to the following nodes or remain dormant with an output of 0, having no effect on the computations made by the following nodes. We know from previous research on neural networks that after network training, specific groups of nodes tend to be responsible for specific outcomes and thus often activate concurrently [3]. For example, in an image recognition model, certain groups of nodes can be expected to activate when the image of a dog is shown, while at least a partially different group should activate for the image of a cat. Activations have been studied in the context of testing neural networks (e.g. [3–6]), but use in mitigating risks during runtime has been scarce [1].

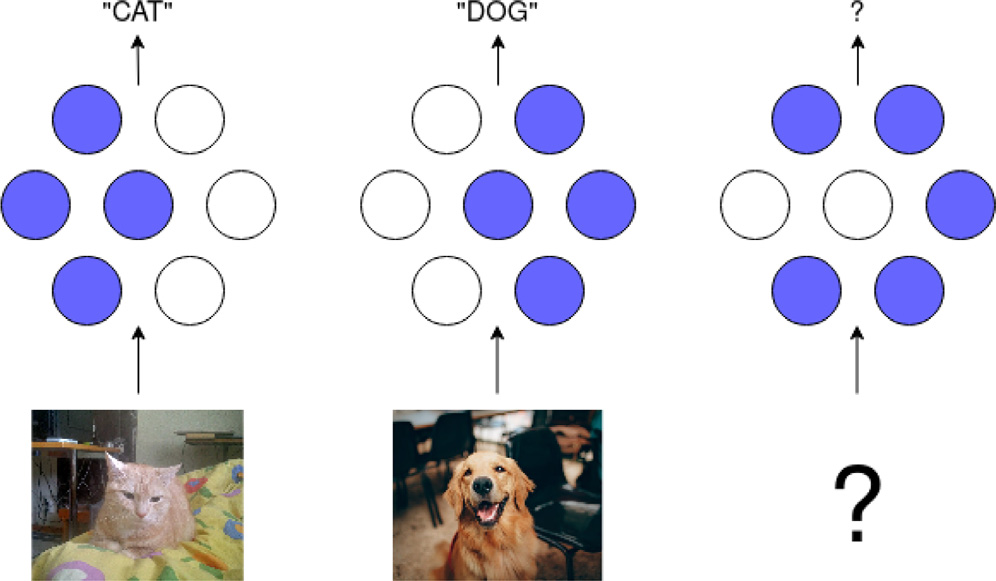
However, what if the activating nodes are suddenly ones that usu-ally do not activate together and thus do not belong to a same group (cf. Fig. 1)? Can something be inferred from this? Do these rare co-activations within a neural network indicate that the computation result is incorrect or that the input has never been seen before? If so, could rare co-activations be used to detect errors in neural networks,

∗ Corresponding au[thor.](mailto:lalli.myllyaho@helsinki.fi)

*E-mail addresses:* [lalli.myllyaho@helsinki.fi](mailto:lalli.myllyaho@helsinki.fi) (L. Myllyaho), [jukka.k.nurminen@helsinki.fi](mailto:jukka.k.nurminen@helsinki.fi) (J.K. Nurminen), [tommi.j.mikkonen@jyu.fi](mailto:tommi.j.mikkonen@jyu.fi) (T. Mikkonen).

<https://doi.org/10.1016/j.array.2022.100201>  
[Received 29 March 2022; Received in revise](https://doi.org/10.1016/j.array.2022.100201)d form 30 May 2022; Accepted 3 June 2022   
Available online 10 June 2022   
2590-0056/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

*L. Myllyaho et al.*  *Array 15 (2022) 100201*



**Fig. 1.** Illustration of activation patterns in a simple neural network. Colouring indicates an activated node. Mutated activation pattern on the far right. Picture of the dog courtesy of Helen Lopez <https://www.pexels.com/photo/short-coated-tan-dog-2253275/>. (For interpretation of [the references to colour in this figure legend, the reader is](https://www.pexels.com/photo/short-coated-tan-dog-2253275/) referred to the web version of this article.)

prevent them from propagating and causing failures, and, thus, increase the fault tolerance of ML systems to mitigate inherent risks?

To test our hypotheses, we train four neural networks. For every node in a neural network, we calculate how often each node activates concurrently with every other node using the training data: i.e. how probable it is that two nodes activate concurrently. This way, we obtain a metric of which nodes often contribute together to the network output and thus belong to one or more of the same groups. Using a separate test data set, we count the number of rare co-activations happening within the neural network for each input and mark down whether the network’s output was correct, incorrect, or if the input belonged to a class that was not present in the training set. Once the tests have been run, we determine whether the number of rare co-activations differ statistically between the following scenarios: (1) test cases for which the output was correct, (2) cases where the output was incorrect, and (3) cases where the network was not trained for the input. Based on our data, we then estimate how well rare co-activations would fit in mitigating three specific major risks that are often present in ML systems: drift in incoming data, single inputs the model cannot handle, and inaccurate predictions [1].

Large numbers of rare co-activations indicate problems in predic-tions. Rare co-activations are, on average, much more common in inputs from untrained classes than in inputs the model has been trained for. Thus, rare co-activations show good potential in detecting drift in incoming data: should the average number of rare co-activations increase, drift is most likely imminent. However, inputs from trained classes contained outliers with a high occurrence number of rare co-activations as well, and some untrained inputs have a low number of occurrences. Thus, detecting inputs that the model cannot handle and preventing them from being used further down in the system is more problematic. Considering the difference in the average number of occurrences, it may be possible to find systems and contexts where using it is feasible, but the system should be able to deal with some false positives and negatives. Additionally, rare co-activations tended to be more common in incorrectly predicted inputs than in correctly predicted ones, but the difference was both smaller and statistically less significant. Thus, detecting single inaccurate predictions may not be feasible based on the number of occurrences alone, but the approach should at least be fine-tuned to find the most indicative co-activations. This paper is organized as follows: Section 2 describes key concepts of system dependability, fault tolerance, and neural networks, along with previous work on activation patterns in neural networks and their utilization in testing and monitoring the networks. Section 3 introduces the novel concepts in detail and describes our goals and research questions. Section 4 describes our experimental set-up, how data was

2

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

the next layer of nodes or remains dormant and, in practice, outputs 0, thus having no effect on the following computations.

The technique responsible for the activation is an activation func-tion [12]. A rectified linear unit (ReLU) is a commonly used activation function. ReLU very closely follows the philosophy of either activating or remaining dormant. Mathematically ReLU is usually formulated as *𝑓*(*𝑥*) = max(0*, 𝑥*). In practice, this means that if the input a node receives is negative or 0, it actually does not have an effect on the following computations, but if the inputs are very strong, the effect the node has on the following layer is also strong. According to Sharma et al. [12], ReLU has proved to be very effective and is one of the most used activation functions today.

*2.3. Related work*

In neural networks, various groups of nodes tend to take responsi-bility of different outcomes [4]. In their work, Tian et al. showed that different groups of nodes in a neural network for autonomous driving tended to activate based on whether the neural network proposed turning to the left or to the right. Xie et al. [5] also suggest that transforming a test input too much will lead to a deformed activation pattern and a wrong result, suggesting that the mutated pattern is related to the incorrect result.

The activations have been used in research concerning the testing of neural networks (e.g. [3–6]). Usually this means finding nodes that have not activated during testing or exploring improved methodologies for creating test cases to find such nodes. This is rooted in the idea that so-called ‘‘neuron coverage’’ is related to code and statement coverage in traditional software: if the neuron has not activated during testing, the effect of that neuron is not known [3].

However, activation-related error detection measures have not been used widely and consistently in practice to achieve fault tolerance [1]. That is, activations have been monitored to initially test the model prior to deployment but not to continuously validate the system during runtime. The idea has raised some interest in practitioners, but how the activations should actually be monitored to detect errors and what conclusions should be made based on them has remained unclear [1]. Numerous attempts have also not been made on the research side, as we are aware of only one paper attempting to build fault toler-ance by specifically utilizing activation monitors. In their work, Cheng et al. [13] form a pattern from the activations of the penultimate layer for each class. If the model prediction differentiates too much from the previously established pattern, the output is flagged as potentially erroneous. This shows promise in detecting some misclassifications. However, they only focus on the penultimate layer and a certain subset of the nodes they consider to be the core nodes affecting the outputs. Thus, they do not entertain the idea of how activations in the earlier layers or outside the core set behave. Also, the focus is on immediate error detection, and how the activations behave across various scenarios (i.e. whether the output was correct, incorrect, or something the model was not trained for) is not addressed. Thus, which types of failures the activation monitors are effective against remains uncertain, as does whether they could also be utilized when monitoring concept drift.

**3. Concepts and goals**

In this section, we introduce the concept of *rare co-activations*, a novel approach to estimate the typicality of activation patterns in a neural network. Our goal is to show that correctly predicted inputs dif-fer from problematic inputs with regards to rare co-activations within the network. Thus, atypical activation patterns would indicate untrust-worthy predictions. If this is the case, monitoring rare co-activations would show potential in error detection in neural networks.

First, we describe the concept of rare co-activations in detail in Section 3.1. Then, we discuss the motivation of our research goal and present our research questions in Section 3.2.

3

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 1**  Models used to test hypotheses. | |  |  |
| Model | Filtered class | Number of outputs | Accuracy in test set  without the filtered class |
| CNN-ankle boot CNN-ankle boot9 CNN-shirt  MLP-ankle boot | 9 (ankle boot) 9  6 (shirt)  9 | 10  9  10  10 | 91.8%  91.7%  95.6%  88.4% |

activation patterns that have occurred within the network before, and more atypical it is. In this study, we are looking into the connection be-tween the problematic predictions and the number of rare co-activation occurring when the prediction is made.

As for RQ2, if the distributions actually differ in the various sce-narios, we aim to detect what types of misbehaviour [1] could be addressed by abusing the rare co-activations. Differences in the number of rare co-activations in and of itself does not mean that the result is useful in error detection and fault tolerance as is. Also, as not every form of fault tolerance is suitable for every type of misbehaviour [1], we must consider how the results could link the approach to known misbehaviour types. In this case, the potential to tackle some forms of misbehaviour must be deduced from how the rare co-activations man-ifest in various scenarios. Specifically, we consider three misbehaviour types that pose a major risk to some systems and that we believe could potentially reveal themselves in the rare co-activations: untrustworthy predictions, inputs that could be problematic for the network, and drift in the incoming data [1].

*3.2. Research goal and questions*  **4. Experimental set-up**

The goal of our research is to show that atypical activation patterns indicate untrustworthy predictions. To build dependable systems, a general need currently exists for fault tolerance in ML systems. How-ever, approaches utilizing node activations to detect errors in neural networks have not been extensively studied regardless of their role in the computation process. The reasoning we have here is that by showing that activation patterns – rare co-activations in our case –behave differently in correct and problematic predictions, we can argue that observing the activation pattern has potential in error detection. In this paper, we study activations in the context of a classification problem, where certain classes are excluded from the training set but remain present in the separate test set. More specifically, we explore how activation patterns behave in the following scenarios:

1. Test cases for which the output is correct;   
2. Test cases for which the output is incorrect despite its class being present in the training set;   
3. Test cases where the input does not belong to any class in the training set.

Based on this, we aim to assess whether the activation patterns we study can be used to improve fault tolerance, especially by detecting erroneous outputs, problematic inputs, and potential concept drift. Henceforth, we will address cases in the scenarios as *correctly predicted inputs*, *incorrectly predicted inputs*, and *untrained inputs*, respectively. The pattern we study is rare co-activations introduced above in Section 3.1. As the activations tend to form patterns [4], it makes sense that nodes in shared groups often activate together. If often activating together implies being in one or more of the same patterns, it may not be unreasonable to think that the disjointed and atypical patter manifested in rare co-activations implies a broken pattern and an untrustworthy prediction. Thus, we try and show whether there is a utilizable connection between rare co-activations and untrustworthy predictions. More specifically, we aim to answer the following research questions:

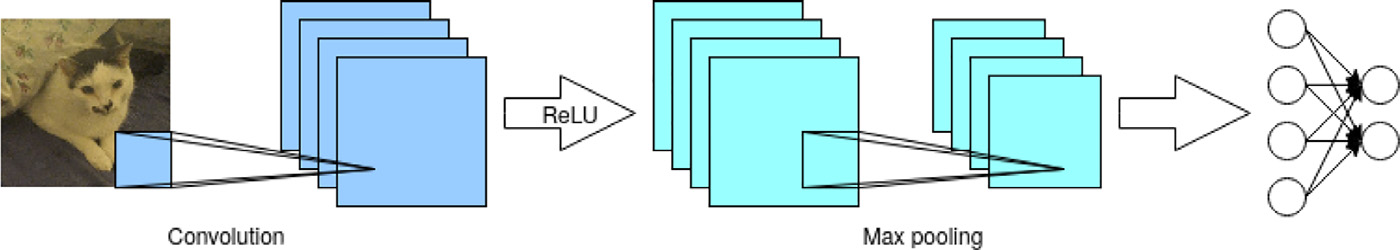
• RQ1: Does the number of rare co-activations statistically differ in the above scenarios?

• RQ2: Can rare co-activations be used to detect erroneous be- haviour when building fault-tolerant ML systems, and how?

The aim of RQ1 is to explore whether the idea is valid in the first place. Only a statistically significant difference in the number of rare co-activations allows us to argue that our approach has any potential in building fault-tolerant ML systems. If the distributions between cases where the neural network made a correct prediction and cases where the prediction was wrong or the input never appeared in the training set are not statistically different, we cannot claim that any meaningful conclusions can be drawn from the number of rare co-activations.

4

*L. Myllyaho et al.*  *Array 15 (2022) 100201*



**Fig. 3.** A simple convolutional neural network. In the convolutional layers, every part of the input goes through a convolution, on which the activation is applied, after which

the strongest activations of every small area are gathered by a pooling layer. Convolutional layers are followed by fully connected layers.

maximum nor have they been pushed to their limits through training time. This decision is twofold. First, even though it were possible to push a neural network to basically label every input in Fashion-MNIST correctly (e.g. Kayed et al. in [18]), this would leave us with very little data to address mislabelled classes present in the training data (Scenario 2 in 3.2). This, in turn, would risk statistical significance and our ability to meet the research goals we have set. Second, real-life ML models may not reach such high levels of accuracy in their respective data sets (e.g. in [19]). Thus, not pushing the models to their limit makes them more on par with their industrial counterparts, rep-resenting them better. With these two reasons combined, our networks are, in a sense, intentionally broken, but also ‘‘good enough’’. In other words, they handle most cases correctly, showing some strength in their behaviour, and yet, make some errors in order to leave us with enough data to answer our research questions. This clearly poses some threats to the validity of the study, which are addressed in Section 7.

Next, we go through all models in more detail.

**CNN-ankle boot:** As the name suggests, CNN-ankle boot is a *convo-lutional neural network* [20] (cf. Fig. 3). The ‘‘ankle boot’’ in the name refers to the class (9, Ankle boot) that is filtered out from the training set for this neural network. The output node for the specific class is still present in the network, even if it is filtered out in the training phase. The structure of CNN-ankle boot begins with three convolutional layers. The convolutional layers consist of 3 × 3 -sized filters with a stride length of 1 and the *same* padding. The three convolutional layers have 32, 64, and 128 filters, respectively. Each convolutional layer is accompanied with a batch normalization layer [21], ReLU activation, and a 2 × 2 -sized max pooling layer [22].

The convolutional layers are followed by two fully connected layers. The fully connected layers also utilize ReLU activation, and consist of 64 and 128 neurons, respectively. Finally, the output layer consists of 10 neurons, utilizing the Softmax activation function [12].

**CNN-ankle boot9:** CNN-ankle boot9 shares most features with CNN-ankle boot except for the number of nodes on the output layer. The filtered class is the same, along with the hidden layers in the neural network. The difference is that the output layer has no reserved output node for the filtered class. This naturally results in only having nine nodes on the output layer.

The reasoning behind this is that they represent two different sit-uations in training a neural network. In the case of CNN-ankle boot9, the imaginary developers are unaware that ankle boots exist and do not reserve an output node for it. With CNN-ankle boot, however, the developers know ankle boots exist, they just do not have enough data for them, and they are left underrepresented in the training set. This adds variety to the results, as CNN-ankle boot9 works ‘‘as intended’’by the imaginary developers, and begins receiving unexpected data, whereas CNN-ankle boot is left broken by the training data and begins receiving appropriate data only after the training is complete.

**CNN-shirt:** CNN-shirt is structurally identical to CNN-ankle boot. The difference is that the class filtered out from the training set is class 6 (Shirt) instead of class 9 (Ankle boot). The purpose of this is to assess whether the phenomena we find are independent from the filtered class or not. Shirt was chosen, as it is evidently different from ankle boots, whereas, for example, sneakers may not be.

**MLP-ankle boot:** MLP-ankle boot is–as the name suggests–a *mul-tilayer perceptron* [23] (cf. Fig. 2 in Section 2). That is, all the hidden

5

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |
| --- |
| **Algorithm 2** CountCo-activations(*𝑖𝑛𝑝𝑢𝑡𝑠*, *𝑁𝑁*, *𝑟𝑎𝑡𝑒𝑠*)  *inputs*: A set of inputs and their corresponding outputs for which the number of rare co-activations in NN are counted  *NN*: Neural network in which the co-activations are monitored  *rates*: Co-activation rates for NN  1: *rareCoActivations* = [][]: an array to store the number of rare co- activations for each input, along with information on whether the input was predicted correctly, incorrectly, or if it belongs to the untrained class  2: **for all** *𝑖* in *𝑖𝑛𝑝𝑢𝑡𝑠* **do**  3: **if** *𝑖* belongs to the untrained class **then**  4: *rareCoActivations*[i][0] = *’untrained’* 5: **else if** *𝑖* predicted correctly by *𝑁𝑁* **then**  6: *rareCoActivations*[i][0] = *’correct’* 7: **else**  8: *rareCoActivations*[i][0] = *’incorrect’* 9: **end if**  10: **for all** node *𝑛* in *𝑁𝑁* **do**  11: **if** *𝑛* activates with *𝑖* **then**  12: **for all** node *𝑚* in *𝑁𝑁* **do**  13: **if** *𝑚* activates with *𝑖* **then**  14: **if** *𝑟𝑎𝑡𝑒𝑠*[*𝑛*][*𝑚*] *<* 0*.*05 **then**  15: *rareCoActivations*[*𝑖*][1] += 1  16: **if** *𝑟𝑎𝑡𝑒𝑠*[*𝑛*][*𝑚*] *<* 0*.*01 **then**  17: *rareCoActivations*[*𝑖*][2] += 1  18: **if** *𝑟𝑎𝑡𝑒𝑠*[*𝑛*][*𝑚*] *<* 0*.*001 **then**  19: *rareCoActivations*[*𝑖*][3] += 1  20: **end if**  21: **end if**  22: **end if**  23: **end if**  24: **end for**  25: **end if**  26: **end for**  27: **end for**  28: **return** rareCoActivations |

in larger networks are absolutely rarer than in a smaller one. There is, figuratively speaking, more room for the activation patterns to be mostly or completely segregated, whereas the patterns may have to share a larger portion of their nodes in the smaller networks. Thus, we do not settle for one arbitrary threshold for rarity, but instead introduce a few to gain more information on the rarity in various networks. Henceforth, we address these thresholds as *rarity thresholds*.

*4.3. Data analysis*

Data analysis is based on statistical tests. To answer RQ1 (do the three scenarios differ in terms of rare co-activations), we assessed whether or not the data points in various groups actually originated from different distributions. That is, we are not only interested in whether our samples are different from each other, but we also want to generalize the results to the entire populations from which the samples originate. Using a statistical test, we can determine how certain we can be that not only the samples are different, but the populations behind them as well. Only after this do descriptive statistics, such as the mean, minimum, and maximum, hold strong relevance when comparing the groups. Once the difference is set by tests designed to do just that, these descriptive statistics reveal the nature of the difference.

We use the *Kruskal–Wallis test* [15] to determine that populations are, in fact, different. The Kruskal–Wallis test is an extension of the Mann–Whitney U test for samples that have more than two groups. As such, it is a non-parametric test that does not presume that samples are normally distributed. The outcome *𝑝* of the Kruskal–Wallis test should

6

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 3**  Results of pairwise comparisons between groups for CNN-ankle boot with activation threshold 0. | | | | |
| Activation  threshold | Rarity  threshold | Dunn-Bonferroni | | (*p*) |
| 0 | *<*5% | correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained | | **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.001**  **0.0**  **0.0** |
| *<*1% |
| *<*0.1% |
| **Table 4**  Descriptive statistics for different groups in CNN-ankle boot with activation threshold 0. | | | | |
| Correct | | Incorrect | | Untrained |
| N | 8265 | 735 | | 1000 |
| Rarity *<* 5% Mean  Median  Max  Min | 39026.32  9380  739477  3 | | 40490.21  13811  688498  25 | 115802.51 85318  494998  524 |
| Rarity *<* 1% Mean  Median  Max  Min | 2088.31 16  202251 0 | 2813.61  40  212147  0 | | 9445.02  985.5  96923  0 |
| Rarity *<* 0*.*1% Mean  Median  Max  Min | 61.34  0  30022  0 | 128.01  0  39403  0 | | 366.59  0  19564  0 |

Considering the mean and median (Table 4), the number of rare co-activations are – on average – slightly more common when the model prediction is incorrect and much more common when the input is from the class that was filtered out of the training set. The relative difference in mean even rises when lowering the rarity threshold, despite the number decreasing and the median in every scenario falling down to 0. The result is similar when comparing the minimum number.

However, the highest number of rare co-activations occurred when the model was correct. This applies for the highest rarity threshold, but the number remains relatively high with the lower thresholds as well, even if the highest maximum number is in the incorrectly predicted ones. This suggests that even correct outputs have outliers with large numbers of rare co-activations.

**Activation threshold 0.0156:** Next, we raise the activation thresh-old to 0.0156. Rare co-activations are more common in incorrectly predicted and untrained inputs than in correctly predicted ones. The average numbers of occurrences are higher and the differences are statistically significant. The chosen threshold is smaller than 99% of all non-zero activations that occurred in CNN-ankle boot in the training set.

As we can see from Table 5, every pairwise comparison suggests a difference in distribution (*𝑝 <* 0*.*05). Below, we present the descriptive statistics for every rarity threshold.

The descriptive statistics remain somewhat consistent despite the raise in activation threshold (Table 6). Rare co-activations in incorrect predictions are slightly more common on average and at minimum, and much more common in the untrained class. Despite this, the maximum number of occurrences in correctly predicted inputs is larger than in other scenarios with the highest rarity threshold and remains in line with the other scenarios with the lower thresholds as well. Median and minimum numbers fall down to 0 in all three scenarios when the rarity threshold is lowered.

7

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 8**  Descriptive statistics for different groups in CNN-ankle boot with activation threshold 0.112. | | | | |
| Correct | | Incorrect | | Untrained |
| Rarity *<* 5%  Mean  Median  Max  Min | 45173.06  11978  793519  25 | 46055.6 16338  770479 108 | 126369.11  91225.5  575951  1269 | |
| Rarity *<* 1%  Mean  Median  Max  Min | 2450.87  33  223204  0 | 3126.73 70  240410 0 | 10423.61  1695.5  92044  0 | |
| Rarity *<* 0*.*1% Mean  Median  Max  Min | 74.47  0  37735  0 | 179.18 0  36194 0 | 359.5  0  20938  0 | |
| **Table 9**  Results of the Kruskal–Wallis tests for CNN-ankle boot9. | | | | |
| Activation  threshold | Rarity threshold | Kruskal–Wallis (*p*) | | |
| 0 | *<*5%  *<*1%  *<*0.1%  *<*5%  *<*1%  *<*0.1%  *<*5%  *<*1%  *<*0.1% | **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0** | | |
| 0.015\* |
| 0.114\*\* |

|  |  |  |  |
| --- | --- | --- | --- |
| \*Activation threshold *<* 99% of activations. \*\*Activation threshold *<* 90% of activations.  **Table 10**  Results of pairwise comparisons between groups for CNN-ankle boot9 with activation threshold 0. | | | |
| Activation threshold | Rarity  threshold | Dunn-Bonferroni | (*p*) |
| 0 | *<*5% | correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained | **0.002 0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.012 0.0**  **0.0** |
| *<*1% |
| *<*0.1% |

*5.2. CNN-ankle boot9*

In this subsection, we go through the results for the model CNN-ankle boot9 in a similar manner. This model is otherwise similar to and similarly trained as CNN-ankle boot, but does not have an output node for the class that was filtered out of the training set. See Section 4.1 for more details.

As we can see from Table 9, for each activation threshold and rarity threshold, at least two groups representing the three scenarios are statistically different (*𝑝 <* 0*.*05) from each other. Thus, we can make meaningful interpretations about the rare co-activations between the groups with each threshold. Next, we present the pairwise comparisons and descriptive statistics of the groups for each activation and rarity threshold.

**Activation**  **threshold**  **0:**  For activation threshold 0, rare co-activations are more common in incorrectly predicted and un-trained inputs than in correctly predicted ones. The average numbers of occurrences are higher and the differences are statistically significant.

8

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 13**  Descriptive statistics for different groups in CNN-ankle boot9 with activation threshold 0.015. | | | |
| Correct | | Incorrect | Untrained |
| Rarity *<* 5%  Mean  Median  Max  Min | 27138.41  4935.5  575003  2 | 29470.76  7195.5  408198  22 | 79336.79 51817.5  398884  298 |
| Rarity *<* 1%  Mean  Median  Max  Min | 1743.35  7  183790  0 | 2244.95  18  144246  0 | 8142.28  710  83236  0 |
| Rarity *<* 0*.*1% Mean  Median  Max  Min | 63.72  0  47122  0 | 113.42  0  27704  0 | 149.35  0  13581  0 |
| **Table 14**  Results of pairwise comparisons between groups for CNN-ankle boot9 with activation threshold 0.114. | | | |
| Activation  threshold | Rarity  threshold | Dunn-Bonferroni | (*p*) |
| 0.114\*\* | *<*5% | correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained | 0.061  **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0**  **0.0** |
| *<*1% |
| *<*0.1% |

\*\*Activation threshold *<* 90% of activations.

not reach statistical significance (*𝑝 <* 0*.*05). Thus, we cannot make strong statements concerning the differences between correctly and incorrectly predicted inputs with that rarity threshold. However, we will present the descriptive statistics for that rarity threshold as well, because the difference between the untrained inputs and the other scenarios are statistically significant.

The descriptive statistics in CNN-ankle boot9 with activation thresh-old 0.015 (Table 13) follow the trend set by the earlier results. Again, on average, rare co-activations are much more common in untrained inputs and slightly more common in incorrectly predicted inputs than in correctly predicted ones. The same goes for the minimum number of occurrences. Again, the highest maximum number of occurrences can be found in correctly predicted inputs.

With the rarity threshold *<* 0*.*1%, we must remember that the difference between correctly and incorrectly predicted inputs is not statistically significant. Thus, strong claims relating to the differences should be avoided. We would, however, like to point out that the mean number of occurrences in incorrectly predicted inputs is still higher than in correctly predicted inputs, which does follow the trend set by the higher rarity thresholds.

**Activation threshold 0.114:** For activation threshold 0.114, rare co-activations are more common in incorrectly predicted and untrained inputs than in correctly predicted ones. The average numbers of oc-currences are higher and the differences are statistically significant for except one.

A pairwise comparison with activation threshold 0.114 is given in Table 14. The activation threshold is smaller than 90% of the non-zero activation in CNN-ankle boot9 with the training set. As can be seen, with the rarity threshold *<* 5%, correctly and incorrectly predicted inputs do not differ from each other to a degree that is statistically significant, although just barely. Every other comparison is statistically significant (*𝑝 <* 0*.*05).

9

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 17**  Results of pairwise comparisons between groups for CNN-shirt with activation threshold 0. | | | |
| Activation  threshold | Rarity  threshold | Dunn-Bonferroni | (*p*) |
| 0 | *<*5% | correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained correct–incorrect  correct–untrained incorrect–untrained | 1  **0.038**  **0.002**  0.507  **0.015**  **0.012**  1  **0.034**  **0.0** |
| *<*1% |
| *<*0.1% |
| **Table 18**  Descriptive statistics for different groups in CNN-shirt with activation threshold 0. | | | |
| Correct | | Incorrect | Untrained |
| N | 8609 | 391 | 1000 |
| Rarity *<* 5% Mean  Median  Max  Min | 38167.45  11702  695969  3 | 33424.39  11950  548909  150 | 42830.75 15505.5  701704  82 |
| Rarity *<* 1% Mean  Median  Max  Min | 2481.47  45  206947  0 | 2910.51  35  155831  0 | 3370.17  67.5  212366  0 |
| Rarity *<* 0*.*1% Mean  Median  Max  Min | 64.96  0  34808  0 | 221.68  0  30814  0 | 107.7  0  26338  0 |

predicted inputs. The descriptive statistics show higher averages and the differences are statistically significant. Rare co-activations are arguably more common in incorrectly predicted inputs than in correctly predicted ones as well according to the descriptive statistics but the differences are not statistically significant.

For activation threshold 0, untrained inputs differ statistically (*𝑝 <* 0*.*05) from the other two scenarios with every rarity threshold (Table 17). However, the pairwise comparisons between correctly and incorrectly predicted inputs do not reach statistical significance with any rarity threshold, nor are they close to reaching it. Therefore, we do not compare their descriptive statistics below, but focus on their differences compared with the untrained inputs.

With activation threshold 0, rare co-activations are, on average, more common in untrained inputs than in the other two scenarios, except for rarity threshold *<* 0*.*1%, where they are more common in incorrectly predicted inputs (Table 18). Apart for the exception, both mean and median are larger than in the counterparts. Contrasting with the previous networks, here, the maximum number of occurrences is also higher in the untrained inputs than in the other two with the rarity thresholds *<* 5% and *<* 1%. The minimum number of occurrences, however, is higher in the incorrectly predicted inputs than in the untrained inputs.

**Activation threshold 0.018:** For activation threshold 0.018, rare co-activations are more common in untrained inputs than in correctly predicted inputs. The descriptive statistics show higher averages and the differences are statistically significant. Rare co-activations are ar-guably more common in incorrectly predicted inputs than in correctly predicted ones as well according to the descriptive statistics but the differences are not statistically significant.

Results of the pairwise comparison with activation threshold 0.018 can be found in Table 19. As with the previous activation threshold, the difference between correctly and incorrectly predicted inputs is not statistically significant (*𝑝 <* 0*.*05) or even close to it. Also, with rarity

10

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 21**  Results of pairwise comparisons between groups for CNN-shirt with activation threshold 0.154. | | | |
| Activation threshold | Rarity  threshold | Dunn-Bonferroni | (*p*) |
| 0.154\*\* | *<*0.1% | correct–incorrect | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 24**  Results of pairwise comparisons between groups for MLP-ankle boot with activation threshold 0. | | | |
| Activation threshold | Rarity  threshold | Dunn-Bonferroni | (*p*) |
| 0 | *<*5% | correct–incorrect | 0.119 |

|  |  |  |  |
| --- | --- | --- | --- |
| correct–untrained  incorrect–untrained | **0.0**  **0.026** | correct–untrained  incorrect–untrained | **0.0**  **0.029** |

\*\*Activation threshold *<* 90% of activations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 22**  Descriptive statistics for different groups in CNN-shirt with activation threshold 0.154 and rarity threshold *<* 0.1%. | | | |
| Correct | | Incorrect | Untrained |
| Mean  Median  Max  Min | 78.13  0  39924  0 | 229.97  0  32371  0 | 146.77  0  32224  0 |
| **Table 23**  Results of Kruskal–Wallis tests for MLP-ankle boot. | | | |
| Activation threshold | Rarity threshold | | Kruskal–Wallis (*p*) |
| 0 | *<*5%  *<*1%  *<*0.1%  *<*5%  *<*1%  *<*0.1%  *<*5%  *<*1%  *<*0.1% | | *<***0.001**  1  1  *<***0.001**  1  1  *<***0.014**  1  1 |
| 0.0339\* |
| 0.34\*\* |

\*Activation threshold *<* 99% of activations.

\*\*Activation threshold *<* 90% of activations.

The difference between untrained inputs and the other two scenarios is statistically significant (*𝑝 <* 0*.*05). Conversely, correctly and incor-rectly predicted inputs do not differ from each other to a statistically significant degree. Thus, we only compare the untrained inputs with the other two.

The descriptive statistics with rarity threshold *𝑝 <* 0*.*1% can be found in Table 22. Rare co-activations are, on average, more common in untrained inputs than in correctly predicted ones. Conversely, un-trained inputs average a smaller number of occurrences than incorrectly predicted inputs. The maximum number of occurrences is smaller in untrained inputs than in the other two, with correctly predicted inputs having the largest number. The median and minimum number of occurrences is 0 in every scenario.

*5.4. MLP-ankle boot*

In this subsection, we present the results for the MLP-ankle boot model in a similar manner. MLP-ankle boot is unlike the other models, as it is a multi-layered perceptron instead of a CNN and contains much fewer nodes than the other models. See Section 4.1 for more details. Results of the Kruskal–Wallis tests for each activation threshold in MLP-ankle boot can be found in Table 23. The test reaches statistical significance (*𝑝 <* 0*.*05) with each activation threshold, but only with rarity threshold *<* 5%. Thus, we only perform the pairwise and descrip-tive statistics comparisons with this rarity threshold as only with those thresholds the groups are meaningfully different with regards to the number of rare co-activations.

**Activation**  **threshold**  **0:**  For activation threshold 0, rare co-activations are more common in untrained inputs than in cor-rectly predicted inputs. The descriptive statistics show higher averages but the differences are statistically significant only with the high-est rarity threshold. Rare co-activations are arguably more common

11

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 27**  Cross-tabulation for different groups and number of rare co-activations in MLP-ankle boot with activation threshold 0.0339 and rarity threshold *<* 5%. | | | | |
| N | | Correct | Incorrect\* | Untrained |
| 0  1 | 7842  118 | | 1014  26 | 962  38 |

\*Incorrectly predicted inputs do not statistically differ from either of the other two scenarios.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 28**  Results of pairwise comparisons between groups for MLP-ankle boot with activation threshold 0.34. | | | |
| Activation threshold | Rarity  threshold | Dunn-Bonferroni | (*p*) |
| 0.34\*\* | *<*5% | correct–incorrect | 0.177 |
| correct–untrained  incorrect–untrained | | | 0.131 **0.010** |

\*\* Activation threshold *<* 90% of activations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 29**  Descriptive statistics for different groups in MLP-ankle boot with activation threshold 0.34 and rarity threshold *<* 5%. | | | |
| Correct\* | | Incorrect | Untrained |
| N  Mean  Median Max  Min | 7960  0.36  0  9  0 | 1040  0.3  0  6  0 | 1000  0.37  0  6  0 |

\*Correctly predicted inputs do not differ statistically from either of the other two scenarios.

As each input in MLP-ankle boot resulted in either 0 or 1 rare co-activations with activation threshold 0 and rarity threshold *<* 5%, for clarity, we present the results as a cross-tabulation instead of descrip-tive statistics (Table 27). Again, rare co-activations are, on average, more common in untrained inputs than in correctly predicted inputs. Correctly predicted inputs are approximately eight times more common than untrained ones. Yet, in inputs where rare co-activations occurred, correctly predicted inputs are only ca. four times as common. Overall, rare co-activations are not very common. **Activation threshold 0.34:** For activation threshold 0.34, rare co-activations are only arguably more common in untrained inputs than in correctly predicted inputs. The descriptive statistics show higher averages but the differences are not statistically significant. As for the differences between incorrectly predicted inputs and correctly predicted ones, the differences are not statistically significant, nor are the descriptive statistics higher.

Results of the pairwise comparison for activation threshold 0.34 can be found in Table 28. The only difference that reaches statistical significance (*𝑝 <* 0*.*05) is between untrained and incorrectly predicted inputs. Correctly predicted inputs do not differ from either of the other two scenarios to a statistically significant degree. We therefore only compare the untrained and incorrectly predicted inputs. The activation threshold is smaller than 90% of the non-zero activations in MLP-ankle boot with the training set.

On average, rare co-activations are more common in untrained inputs than in incorrectly predicted ones (Table 29). However, rare co-activations are uncommon overall. The median and minimum number of occurrences for both scenarios is 0. The maximum number for both is 6, which is, arguably, not that large either.

**6. Discussion**

In this section, we discuss the results and what they mean for our research questions. In Section 6.1, we present the trends in differences between the scenarios in the neural networks. In Section 6.2, we

12

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

predicted inputs, the sample of incorrectly predicted inputs may just be too small to show that the difference is statistically significant. For example, there are only 391 inputs that CNN-shirt predicted incor-rectly. The other issue is that the co-activation rates were computed using the entire training set, excluding the class that we had decided to filter out for that model. This, of course, results in a situation where the co-activation rates are computed not only with the inputs that the model predicts correctly but also with those that are predicted incorrectly. This could, in a sense, mean that the co-activation rates are, perhaps idealistically, computed based on what the model *should* know, instead of what it *does* know. This, in turn, may result in raising the co-activation rate of some nodes that actually contribute in the model predicting the input incorrectly and thus making this harmful co-activation acceptable from the viewpoint of rare co-activation. One thing to note is that even though rare co-activations are, on average, least common in correctly predicted inputs, this does not mean that they are necessarily absent. In fact, correctly predicted inputs can have outliers with very large numbers of occurrences (cf. Fig. 4), and the maximum number of occurrences may be as high or even higher than in the other two scenarios. Conversely, the minimum number of occurrences tends to be the lowest in correctly predicted inputs, when it is not 0 for all three scenarios.

Regarding the goals we set for the different models in Section 4.1, changing the class that was filtered out in the training set makes little difference with regards to the main results. When changing the filtered class from ’ankle boot’ to ‘shirt’, rare co-activations remain at a higher level in the untrained inputs than in the correctly predicted ones, although by a smaller margin. An exception to this is the highest acti-vation threshold, which, combined with the rarity thresholds 5% and 1%, does not show statistically significant differences in the scenarios. However, lowering the rarity threshold to 0.1% yields statistically sig-nificant results, and the rare co-activations remain higher in untrained inputs.

The largest difference between CNN-ankle boot and CNN-shirt is that correctly and incorrectly predicted inputs do not differ in CNN-shirt to a degree that is statistically significant with our sample. Es-pecially with lower rarity thresholds, rare co-activations tend to be slightly more common in incorrectly predicted ones, but making any stronger claims on the matter is not possible with our sample. As discussed above, the statistical insignificance in this case may be due to our smallish sample size of incorrectly predicted inputs in CNN-shirt. Changing the overall structure of the network from a large CNN to a small MLP does not change the overall trend of the results but has a great effect on the number of occurrences. Rare co-activations are still most common in untrained inputs, with the comparison of correctly and incorrectly predicted inputs falling short of statistical significance. One large difference compared with the other networks is that rare co-activations occurred far less overall in MLP-ankle boot. With the two lowest activation thresholds, the maximum number of occurrences per input was 1, with 0 being far more common in every scenario. Occurrences were more common in untrained inputs than in correctly or incorrectly predicted inputs. The results were not much different with the highest activation threshold either.

Another thing to note in the MLP is that, while the rarity threshold in CNN-shirt had to be lowered to find statistical significance with a high activation threshold, in MLP-ankle boot, only the highest rarity threshold produced statistically significant results for each activation threshold. This occurred because there simply were not enough rare co-activations below the lower thresholds, meaning that most nodes in the small network tend to activate together at least sometimes. This suggests that the larger networks have more room for the activation patterns to grow partially or even completely separate, whereas, per-haps unsurprisingly, the nodes in the smaller MLP need to contribute more to each computation.

Whether the untrained input has an output node or not does not seem to make a difference in the results: the results are quite similar

13

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

layers of a CNN to those that happen in the fully connected layers, and– as coRate(*𝑛, 𝑚*) is not necessarily equal to coRate(*𝑚, 𝑛*) – whether co-activations that are rare both ways would be more indicative than those that are rare in only one way.

**7. Validity**

We base our validity discussion on the work of Shadish et al. [25]. Thus, validity is considered through statistical conclusion validity, internal validity, construct validity, and external validity.

Statistical conclusion validity means the validity of the statistical tests and claims made based on them. In our work, we have taken the following measures to ensure the validity of our statistical claims. We have chosen the tests so that we do not violate the assumptions the tests make of the data. The Kruskal–Wallis test was chosen specifically because it does not assume the data to be normally distributed, and we used a Dunn-Bonferroni post hoc test to see which scenarios actually differ from each other. The level for determining significance was chosen beforehand, all data were gathered in one go, and the tests were run only after all the data were gathered to avoid tinkering the results to our liking by, for example, adjusting the significance level or by gathering additional data that would reach that level.

However, the choice of significance level is the most apparent threat to statistical conclusion validity. The tests for statistical significance do not, strictly speaking, show that a phenomenon exists or does not exist. What they do show, is, considering the size of the samples and magnitude of difference between them, the probability that the two samples come from equally distributed populations. Considering this, the significance level of *𝑝 <* 0*.*05 is, although common and customary, quite conservative: it means that there must be more than 95% cer-tainty before we declare something to be ‘‘different enough’’. Absence of evidence is not necessarily evidence of absence, especially when *𝑝* is very close to the chosen level. For this reason, especially for incorrectly predicted inputs, we have noted trends that are quite consistent but not statistically significant in a toned-down manner, hinting at results that could be found in, for example, a larger sample. Coincidentally, the sample size of incorrectly predicted inputs is another thing we consider to be a threat to statistical conclusion validity, especially in the CNNs. Internal validity means the validity of causality: do the treatment and the outcome actually reflect the causality between them? In our pa-per, this would mean whether or not the number of rare co-activations is actually a valid indication of an incorrect prediction or an untrained input. The answer seems to be twofold. According to statistical tests and descriptive statistics, larger numbers of rare co-activations occurring especially in untrained inputs are both significant and, we would argue, relevant. As such, rare co-activations are in some relation to the phenomena we are studying. However, the large maximum number of rare co-activations in correctly predicted inputs suggests that the number of rare co-activations should not be treated as an absolute indication of an incorrect prediction or an untrained input. Thus, there could be more fine-tuned details that are even more indicative of these troubled inputs, as discussed at the end of Section 6.2. Also, we try to avoid making too strong claims of the usefulness of rare co-activations with regards to promoting fault tolerance when the statistical results do not imply strong enough leverage to do so (see Section 6.2). Construct validity means the validity of conceptualization and the-oretical generalization, i.e., whether the concepts are properly defined and understood. This type is difficult to assess, as the novelty of this study are the constructs we must deal with. Concepts of co-activation rate and rare co-activations are something we define here, and what implications they may have is something we aim to understand. In other words, gaining a better understanding is our goal. As such, on the one hand, we have full understanding of the concepts, as we are the ones who defined them here. On the other hand, we are only just finding out how they behave in certain situations, and the results we have is all the understanding we have gained of the phenomenon. We

14

*L. Myllyaho et al.*  *Array 15 (2022) 100201*

**CRediT authorship contribution statement**

**Lalli Myllyaho:** Conceptualization, Methodology, Software, Vali-dation, Formal analysis, Investigation, Data curation, Writing – origi-nal draft, Visualization. **Jukka K. Nurminen:** Conceptualization, Re-sources, Writing – review & editing, Supervision, Project administra-tion, Funding acquisition. **Tommi Mikkonen:** Conceptualization, Writ-ing – review & editing, Supervision, Project administration, Funding acquisition.

**Declaration of competing interest**

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

**Acknowledgements**

This work was funded by local authorities (‘‘Business Finland’’) under grant agreement ITEA-2019-18022-IVVES of ITEA3 programme and grant agreement ITEA-2020-20219-IML4E of ITEA4 programme. We acknowledge the help of Antti Klemetti, Dennis Muiruri, and Juha Mylläri in implementing the experimental set-up, the help of Mikko Raatikainen and Tomi Männistö in revising the manuscript, and thank CSC – IT Center for Science, Finland, for computational resources.

**References**

[1] [Myllyaho L, Raatikainen M, Männistö T, Nurminen JK, Mikkonen T. On misbehaviour and fault tolerance in machine learning systems. J Syst Softw 2022;183:111096.](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb1)

[2] [Abiodun OI, Jan](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb1)[tan A, Omolara AE, Dada KV, Mohamed NA, Arshad H.](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb2)

[State-of-the-art in artificial neural network applications: A survey. Heliyon 2018;4(11):e00938.](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb2)

[3] [Pei K, Cao Y, Yang](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb2) [J, Jana S. Deepxplore: Automated whitebox testing of deep learning systems. In: Proceedings of the 26th symposium on operating systems principles. 2017, p. 1–18.](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb3)

[4] [Tian Y, Pei K, Jana S,](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb3) [Ray B. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th international conference on software engineering. 2018, p. 303–14.](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb4)

[5] [Xie X, Ma L, Juefei-Xu F, Xue M, Chen H, Liu Y, et al](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb4)[. Deephunter: A coverage-guided fuzz testing framework for deep neural networks. In: Proceedings of the 28th ACM SIGSOFT international symposium on software testing and analysis. 2019, p. 146–57.](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb5)

[6] [Ma L, Juefei-Xu](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb5) [F, Zhang F, Sun J, Xue M, Li B, et al. Deepgauge: Multi-granularity testing criteria for deep learning systems. In: Proceedings of the 33rd ACM/IEEE international conference on automated software engineering. 2018, p. 120–31.](http://refhub.elsevier.com/S2590-0056(22)00050-9/sb6)

15