Array 15 (2022) 100201

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Node co-activations as a means of error detection—Towards fault-tolerant neural networks

Lalli Myllyaho [a,*], Jukka K. Nurminen [a], Tommi Mikkonen [b]

[a] University of Helsinki, Finland
[b] University of Jyväskylä, Finland

ARTICLE INFO

Keywords:
Machine learning
Fault tolerance
Neural networks
Error detection
Concept drift
Dependability

ABSTRACT

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, whimsical 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 trustworthiness 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]. Furthermore, 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 interconnected 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 usually 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 author.
E-mail addresses: lalli.myllyaho@helsinki.fi (L. Myllyaho), jukka.k.nurminen@helsinki.fi (J.K. Nurminen), tommi.j.mikkonen@jyu.fi (T. Mikkonen).

https://doi.org/10.1016/j.array.2022.100201
Received 29 March 2022; Received in revised form 30 May 2022; Accepted 3 June 2022
Available online 10 June 2022

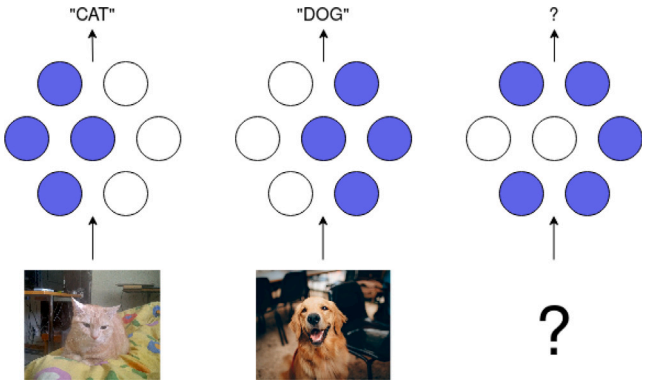

Fig. 1. Illustration of activation patterns in a simple neural network. Colouring indicates an activated node. Mutated activation pattern on the far right. Picture of the dog courtesy of Helen Lopez https://www.pexels.com/photo/short-coated-tan-dog-2253275/. (For interpretation of the references to colour in this figure legend, the reader is 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 predictions. 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 collected and the methods of analysis. Results can be found in Section 5. Section 6 discusses the results, while Section 7 discusses the study validity. Section 8 concludes the paper.

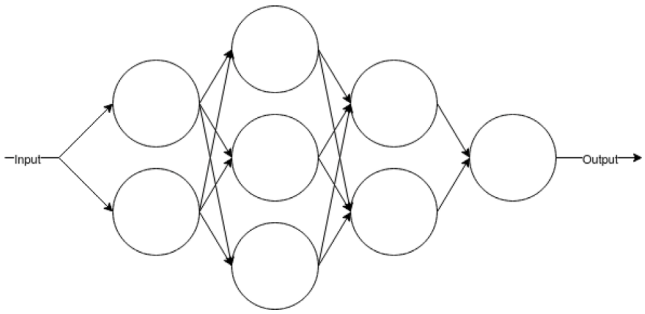

Fig. 2. A simple neural network, more precisely, a multi-layer perceptron.

2. Background

2.1. Dependability and fault tolerance

The *dependability* of a system means its trustworthiness [7]. Dependability is usually assessed by evaluating a system's reliability, availability, and maintainability. Essentially, a dependable system – at the very least – delivers correct service consistently, does not suffer from long periods of downtime, and is easily corrected and altered.

System dependability is threatened by *failures*, *errors*, and *faults* [8]. Failures are deviations from a desired service. They are caused by propagating errors made by the system, i.e. incorrect functioning of the system. Errors are caused by faults that are defects in system components (software or hardware), activated by given inputs in a given state.

Fault tolerance is one tool for diminishing these threats [8]. Fault tolerance aims for a system design that can prevent occurring errors from propagating in the system and causing failures by detecting the error and handling it before further damage is done. The need for fault tolerance in ML systems has lately been recognized more [1]. This is in no small part due to the nature of the ML models themselves. According to Myllyaho et al. [1], the problems in ML systems often originate from inaccuracies that the models hold, along with their proneness to so-called concept drift. That is, an ML model is trained and, ideally, it can generalize what it has learned to all data that are similar to the training data. However, the generalization rarely, if ever, is successful enough to reach a 100% prediction accuracy in new data in the first place, and the model rarely is able to handle data that are vastly different from the training data. Thus, ML systems can be seen as inherently faulty because of their approximate nature [9], and an initially adequate system can erode and become faulty over time [10]. To add insult to injury, erroneous behaviour is often difficult to detect [1], which is why we focus on the detection phase of fault tolerance in this paper.

Research on fault tolerance in ML systems has mainly focused on various input and output observers and model redundancy [1]. This means that, for example, changes in inputs and outputs can be monitored, unacceptable values are handled differently, and the system may contain multiple models that handle inputs with some orchestration. The inner workings and structures of the models, however, are rarely utilized in error detection to achieve fault tolerance.

2.2. Neural networks

The structure of neural networks consists of node layers [11] (see Fig. 2). The previous layer is connected to the next one. That is, when a node receives an input, it either activates and passes on an output to

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 function [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 responsibility 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 tolerance 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 differ from problematic inputs with regards to rare co-activations within the network. Thus, atypical activation patterns would indicate untrustworthy 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.

3.1. Co-activation rate & Rare co-activations

As described in Section 2, nodes in neural networks either activate or remain dormant, and these activations form patterns responsible for certain outputs. If two nodes belong to one or more of the same patterns, they could be expected to activate together a fair share of the time.

To estimate whether two nodes do not belong to any of these patterns, we present the idea of *co-activation rate*: how likely is it that node m activates when node n activates. More specifically, for every node n in a neural network, we calculate its co-activation rate with node m in a set of inputs I as

$$rate(n, m) = \frac{\sum_{i=1}^{I} n_i \cap m_i}{\sum_{i=1}^{I} n_i},$$

where $rate(n, m)$ is the co-activation rate of node n with node m, and $n_i, m_i = 1$ if nodes n and m activate with input i and otherwise $n_i, m_i = 0$.

Algorithmically, the co-activation rates for a set of inputs I can be calculated with Algorithm 1. Using the algorithm, we will end up with a two-dimensional array *rates*, from which the co-activation rates for nodes n and m can be found as, in fact, $rates[n][m] = rate(n, m)$. The time complexity of Algorithm 1 is $O(in^2)$, where i is the number of inputs in I, and n is the number of nodes in a neural network NN.

Algorithm 1 Co-activationRates(*inputs*, NN)

inputs: Set of inputs in which the co-activation rates are calculated
NN: Neural network for which the co-activation rates are calculated

```
rates = [][]: an array in which to store the co-activation rates
for all i in inputs do
  for all node n in NN do
    if n activates with i then
      for all node m in NN do
        if m activates with i then
          rates[n][m] += 1
        end if
      end for
    end if
  end for
end for
for all node n in NN do
  for all node m in NN do
    rates[n][m] = rates[n][m] / rates[n][n]
  end for
end for
return rates
```

To produce meaningful results, the set of inputs used to calculate the co-activation rates should be chosen appropriately. The approach we chose was to calculate the co-activation rates after the networks were trained and use the same training set that was used to train them. In this way, the co-activation rates should represent the activation patterns of the input classes that the network should be able to generalize to. Thus, co-activation rates describe the inner workings of the neural networks in cases where the network can reasonably be expected to handle correctly, whereas cases that have no representation in the training set may not produce good results.

The activation pattern we study in this paper is derived from co-activation rate: a *rare co-activation* occurs when two nodes with a low co-activation rate activate within a neural network during a prediction. Thus, a rare co-activation is an indication of such computations that normally do not occur during a prediction. The more these rare co-activations occur, the more disjointed the activation pattern is from

Table 1
Models used to test hypotheses.

Model	Filtered class	Number of outputs	Accuracy in test set without the filtered class
CNN-ankle boot	9 (ankle boot)	10	91.8%
CNN-ankle boot9	9	9	91.7%
CNN-shirt	6 (shirt)	10	95.6%
MLP-ankle boot	9	10	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 between the problematic predictions and the number of rare co-activation occurring when the prediction is made.

3.2. Research goal and questions

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. However, 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 behaviour 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.

As for RQ2, if the distributions actually differ in the various scenarios, 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 manifest 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].

4. Experimental set-up

In this section, we describe how the experiments were conducted. First, Section 4.1 describes the neural networks from which we gathered the data about the rare co-activations, along with how the networks differ from each other and why. Then, in Section 4.2, we describe what kind of data about rare co-activations in those networks was gathered and how. Finally, in Section 4.3 we describe how the data was analysed to draw conclusions and to ensure statistical significance of our findings. This combination of data triangulation [14] across networks and statistical rigour [15] raises confidence in our findings.

4.1. Neural networks

To test the approach, four neural networks were built. The networks are intended to represent mundane neural networks in which we observe how the rare co-activations behave in the three different scenarios considered (correctly predicted, incorrectly predicted and untrained inputs, see Section 3.2). The differences between networks were designed to catch differences often present in neural networks (see descriptions below) and add data triangulation [14]. A general description of the models is presented in Table 1.

All networks were trained using the Fashion-MNIST [16] data set. Fashion-MNIST consists of 28 × 28 size greyscale images depicting pieces of clothing.[1] The training set and test set for Fashion-MNIST contain 60 000 and 10 000 images, respectively, both divided into 10 evenly sized classes. The classes are – in order of labels from 0 to 9 – T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot. The networks were trained for 15 epochs using Keras[2] machine learning library. To make the results easier to reproduce, a fixed random seed (3) was chosen with a throw of a d20 die.

To mimic Scenario 3 from Section 3.2 (a class of inputs was missing from the training set, but appears after training), one class was excluded from the training set, similarly to Ackerman et al. [17]. This way, in the testing phase with a separate test data set, we have both inputs that belong to classes the network was trained to recognize, along with inputs that the neural network should not have extensive knowledge of. Thus, the excluded class represents situations where all inputs do not resemble the data that the neural network was trained with.

The networks achieved an accuracy ranging from 88.4% to 95.6% (see Table 1) in the test set, excluding the filtered out class. Omitting the filtered out class when measuring the accuracy gives a better estimate of how well the neural networks perform in tasks they should know. This, in our opinion, better estimates the model's ability to learn the data set than by including the class it was not trained with in the first place. We note that the used networks are not fine-tuned to the

[1] Examples of Fashion-MNIST can be found at: https://github.com/zalandoresearch/fashion-mnist.
[2] https://keras.io/.

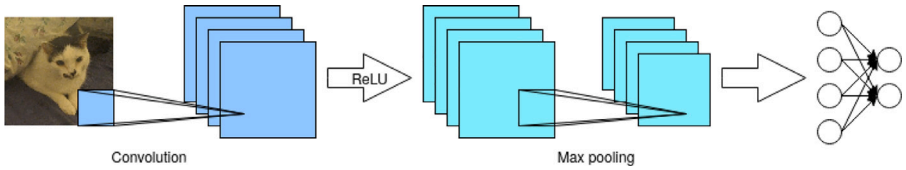

Fig. 3. A simple convolutional neural network. In the convolutional layers, every part of the input goes through a convolution, on which the activation is applied, after which the strongest activations of every small area are gathered by a pooling layer. Convolutional layers are followed by fully connected 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, representing 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 *convolutional 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 situations 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 *multilayer perceptron* [23] (cf. Fig. 2 in Section 2). That is, all the hidden layers in the neural network are fully connected layers, utilizing ReLU activation and batch normalization. There are two hidden layers with 64 and 128 nodes, respectively, 10 output nodes, and the filtered class is Ankle boot.

The reasoning behind the inclusion of MLP-ankle boot is twofold. First, including models with different topology provides additional information whether the results are dependent on certain technologies or not. Second, MLP-ankle boot is a smaller network than the other neural networks. This should give us implications of the effect size that the size of the network has on the phenomena.

4.2. Data collection

Data were collected with an experimental set-up utilizing Keras and NumPy.[3] First, the networks were trained using a training set, from which one class was entirely excluded. Next, co-activation rates for each node in each neural network were calculated using Algorithm 1, introduced in Section 3.1, and the same data that were used for training the network, still excluding one class. Finally, the number of rare co-activations was computed and saved for each input in a separate test data set that included all the classes.

Additional details were considered before calculating the co-activation rates, e.g. when should a node be counted as activated. An apparent choice would be when the node outputs a non-zero number, as that is de facto how a ReLU activation function works. However, a node could output a very small number $\delta > 0$ that has no actual effect on the outcome of the computations. It is less obvious if such activations are actually meaningful regarding the outcome. To assess this, we calculate the co-activation rates using three thresholds that are counted as an activation: 0 and two model-specific thresholds, namely, a threshold that is smaller than 90% of that network's non-zero activations in the training set and one that is smaller than 99% of the non-zero activations. Henceforth, we will address these thresholds as *activation thresholds*.

Furthermore, it is not obvious which output should be counted in the convolutional parts of the CNNs. Activation functions are applied first in the CNNs, after which the strongest activations close to each other are gathered by the pooling layer, while the weakest are filtered out. Thus, there are two consecutive parts that have the outputs of the activation function as their values. We chose to use the outputs of the pooling layer, as they are the ones actually affecting the computations of the following layers.

After the co-activation rates for every neural network were calculated, the number of rare co-activations was counted and saved using Algorithm 2. First, for each input in the Fashion-MNIST test set, we mark down which of the three scenarios the input represents: is the input predicted correctly, incorrectly, or is it untrained (cf. Section 3.2). Next, the number of rare co-activations in the neural network NN with that input are counted and saved. This way, we obtain three types of data points that, as a whole, represent the three scenarios. Finally, in practice, the data are saved to a .CSV file.

On lines 14–23 of Algorithm 2, we attempt to capture the elusive keyword *rare*. As we do not know how the rarity behaves in various networks, it is entirely possible that, for example, the rare co-activations

[3] https://numpy.org/.

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

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
for all *i* in *inputs* **do**
  if *i* belongs to the untrained class **then**
    rareCoActivations[i][0] = *'untrained'*
  else if *i* predicted correctly by *NN* **then**
    rareCoActivations[i][0] = *'correct'*
  else
    rareCoActivations[i][0] = *'incorrect'*
  end if
  for all node *n* in *NN* **do**
    if *n* activates with *i* **then**
      for all node *m* in *NN* **do**
        if *m* activates with *i* **then**
          if $rates[n][m] < 0.05$ **then**
            rareCoActivations[*i*][1] += 1
            if $rates[n][m] < 0.01$ **then**
              rareCoActivations[*i*][2] += 1
              if $rates[n][m] < 0.001$ **then**
                rareCoActivations[*i*][3] += 1
              end if
            end if
          end if
        end if
      end for
    end if
  end for
end for
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 be interpreted so, that with $1 - p$% of certainty, at least two of the populations from which the samples originate from are different. We use the common $p < 0.05$ for the significance level. Thus, when the test suggests that, with more than 95% certainty, at least two groups come from different distributions, we accept that this is actually the case. To assess which groups are different when the Kruskal–Wallis test is significant, we use *Dunn's test* [24] with Bonferroni correction.

Once the Kruskal–Wallis test finds a significant difference between the groups and Dunn's test has identified which groups are different, we compare the descriptive statics of those groups. This way, we acquire knowledge on the nature of the difference: Are rare co-activations more common in certain scenarios? Is there a lot of overlap? Based on these statistics, along with information on which groups actually differ from each other, we assess the potential usefulness in the context of fault tolerance. As descriptive statistics we use mean, median, maximum, and minimum, and – when feasible – cross-tabulation.

All statistics were gathered using SPSS.[4]

Table 2
Results of Kruskal–Wallis tests for CNN-ankle boot.

Activation threshold	Rarity threshold	Kruskal–Wallis (p)
0	<5%	**0.0**
	<1%	**0.0**
	<0.1%	**0.0**
0.0156*	<5%	**0.0**
	<1%	**0.0**
	<0.1%	**0.0**
0.112**	<5%	**0.0**
	<1%	**0.0**
	<0.1%	**0.0**

*Activation threshold < 99% of activations.
**Activation threshold < 90% of activations.

5. Results

In this section, we examine our results obtained using the experimental set-up. The results are presented for every neural network in their own subsection. In turn, for every network, the results are presented for each activation threshold and each rarity threshold, starting from the lowest one. (see Section 4.2 for more details). Only statistically significant results are presented in detail.

5.1. CNN-ankle boot

As presented in Table 2, at least two groups in CNN-ankle boot are statistically different ($p < 0.05$) from each other for each threshold. Thus, we can make meaningful interpretations about the rare co-activations between the groups with each threshold. Now, we examine the pairwise comparisons of groups for each activation threshold and rarity threshold.

Activation threshold 0: For activation threshold 0, 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.

Every pairwise comparison between groups is statistically significant ($p < 0.05$) (Table 3). Thus, we can confidently say that in CNN-ankle boot, co-activations that occurred between nodes with a co-activation rate of less than 5%, 1%, or 0.1% in the training set, manifest differently when the prediction is correct, incorrect, or with unknown inputs. Next, we present the descriptive statistics of the groups to compare how the groups differentiate. The comparison is made for each rarity threshold, as they all hold significance.

[4] https://www.ibm.com/analytics/spss-statistics-software.

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	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<0.1%	correct–incorrect	**0.001**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**

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	39026.32	40490.21	115802.51
Median	9380	13811	85318
Max	739477	688498	494998
Min	3	25	524
Rarity < 1%			
Mean	2088.31	2813.61	9445.02
Median	16	40	985.5
Max	202251	212147	96923
Min	0	0	0
Rarity < 0.1%			
Mean	61.34	128.01	366.59
Median	0	0	0
Max	30022	39403	19564
Min	0	0	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 threshold 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.

Table 5
Results of pairwise comparisons between groups for CNN-ankle boot with activation threshold 0.0156.

Activation threshold	Rarity threshold	Dunn-Bonferroni	(p)
0.0156*	<5%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<0.1%	correct–incorrect	**0.001**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**

*Activation threshold < 99% of activations.

Table 6
Descriptive statistics for different groups in CNN-ankle boot with activation threshold 0.0156.

	Correct	Incorrect	Untrained
Rarity < 5%			
Mean	39735.17	40869.48	117150.13
Median	9629	13646	86450.5
Max	744755	692816	507542
Min	2	21	671
Rarity < 1%			
Mean	2141.24	2899.36	9744.11
Median	736	1646	15452.5
Max	389505	407002	214447
Min	0	0	12
Rarity < 0.1%			
Mean	62.64	132.6	360.68
Median	0	0	0
Max	31614	38045	19756
Min	0	0	0

Table 7
Results of pairwise comparisons between groups for CNN-ankle boot with activation threshold 0.112.

Activation threshold	Rarity threshold	Dunn-Bonferroni	(p)
0.112**	<5%	correct–incorrect	**0.001**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<0.1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**

**Activation threshold < 90% of activations.

Activation threshold 0.112: Next, we raise the activation threshold to 0.112, which is smaller than 90% of non-zero activations occurring in the neural network with the training set. 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 pairwise comparisons of the scenarios (Table 7) are statistically significant ($p < 0.05$) with every rarity threshold. Below, we present the descriptive statistics for every rarity threshold.

Based on the mean and median, rare co-activations are slightly more common in incorrectly predicted inputs for activation threshold 0.112 and much more common in the non-trained inputs (Table 8). Also, the minimum number of occurrences was highest in the untrained inputs and second highest in the incorrectly predicted inputs. However, the maximum number of occurrences was highest in the correctly predicted inputs in all but one rarity threshold. Overall, this follows the basic narrative of the lower activation thresholds.

Table 8
Descriptive statistics for different groups in CNN-ankle boot with activation threshold 0.112.

	Correct	Incorrect	Untrained
Rarity < 5%			
Mean	45173.06	46055.6	126369.11
Median	11978	16338	91225.5
Max	793519	770479	575951
Min	25	108	1269
Rarity < 1%			
Mean	2450.87	3126.73	10423.61
Median	33	70	1695.5
Max	223204	240410	92044
Min	0	0	0
Rarity < 0.1%			
Mean	74.47	179.18	359.5
Median	0	0	0
Max	37735	36194	20938
Min	0	0	0

Table 9
Results of the Kruskal–Wallis tests for CNN-ankle boot9.

Activation threshold	Rarity threshold	Kruskal–Wallis (p)
0	<5%	**0.0**
	<1%	**0.0**
	<0.1%	**0.0**
0.015*	<5%	**0.0**
	<1%	**0.0**
	<0.1%	**0.0**
0.114**	<5%	**0.0**
	<1%	**0.0**
	<0.1%	**0.0**

*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	**0.002**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<0.1%	correct–incorrect	**0.012**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**

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 untrained inputs than in correctly predicted ones. The average numbers of occurrences are higher and the differences are statistically significant.

Table 11
Descriptive statistics for different groups in CNN-ankle boot9 with activation threshold 0.

	Correct	Incorrect	Untrained
N	8256	744	1000
Rarity < 5%			
Mean	26220.5	28798.58	77398.41
Median	4546.5	7165	49562.5
Max	566458	391499	384479
Min	2	11	334
Rarity < 1%			
Mean	1672.11	2151.78	7923.46
Median	6	15	608.5
Max	179581	139666	82259
Min	0	0	0
Rarity < 0.1%			
Mean	63.46	111.02	152.59
Median	0	0	0
Max	46778	27016	13429
Min	0	0	0

Table 12
Results of pairwise comparisons between groups for CNN-ankle boot9 with activation threshold 0.015.

Activation threshold	Rarity threshold	Dunn-Bonferroni	(p)
0.015*	<5%	correct–incorrect	**0.009**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<0.1%	correct–incorrect	0.055
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**

*Activation threshold < 99% of activations.

From Table 10, we can see that each pairwise comparison is statistically significant ($p < 0.05$) with activation threshold 0. This suggests that the distribution of each group is different from one another, and comparisons between the groups can be made. Below, we present the descriptive statistics for each rarity threshold.

The descriptive statistics for rarity threshold seems to follow the trend in the previous model (Table 11). On average, the untrained inputs have a much larger number of rare co-activations than the other two scenarios, and rare co-activations in incorrectly predicted ones are slightly more numerous than in the correctly predicted ones. The same goes for the minimum number of occurrences, before it falls down to 0 in all scenarios.

However, the maximum number of occurrences is slightly different than with the previous network. Unlike in the previous neural network, where correctly and incorrectly predicted inputs were quite close to each other with almost every threshold, here, the correctly predicted inputs have a much larger maximum number of occurrences than either of the other two scenarios. Again, this provides more evidence that rare co-activations may occur in high numbers in some cases, even if the network's prediction is correct.

Activation threshold 0.015: For activation threshold 0.015, 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 for except one.

The pairwise comparison of scenarios in CNN-ankle boot9 with the raised activation threshold of 0.015 can be found in Table 12. The activation threshold is smaller than 99% of the non-zero activations in CNN-ankle boot9 with the training set. The most noticeable differences to the previous results is that with the rarity threshold < 0.1%, the differences between correctly and incorrectly predicted inputs do

Table 13
Descriptive statistics for different groups in CNN-ankle boot9 with activation threshold 0.015.

	Correct	Incorrect	Untrained
Rarity < 5%			
Mean	27138.41	29470.76	79336.79
Median	4935.5	7195.5	51817.5
Max	575003	408198	398884
Min	2	22	298
Rarity < 1%			
Mean	1743.35	2244.95	8142.28
Median	7	18	710
Max	183790	144246	83236
Min	0	0	0
Rarity < 0.1%			
Mean	63.72	113.42	149.35
Median	0	0	0
Max	47122	27704	13581
Min	0	0	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	0.061
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**
	<0.1%	correct–incorrect	**0.0**
		correct–untrained	**0.0**
		incorrect–untrained	**0.0**

**Activation threshold < 90% of activations.

Table 15
Descriptive statistics for different groups in CNN-ankle boot9 with activation threshold 0.114.

	Correct	Incorrect	Untrained
Rarity < 5%			
Mean	37741.43	40090.38	105518.56
Median	10466.5	14023	73586.5
Max	706157	529095	531651
Min	20	49	521
Rarity < 1%			
Mean	2289.14	3040.85	9884.75
Median	26	60.5	1373.5
Max	204396	160685	93372
Min	0	0	0
Rarity < 0.1%			
Mean	71.58	121.05	210.67
Median	0	0	0
Max	43674	27700	13628
Min	0	0	0

Table 16
Results of the Kruskal–Wallis tests for CNN-shirt.

Activation threshold	Rarity threshold	Kruskal–Wallis (*p*)
0	<5%	**0.002**
	<1%	**0.005**
	<0.1%	< **0.001**
0.018*	<5%	**0.009**
	<1%	**0.016**
	<0.1%	< **0.001**
0.154**	<5%	0.236
	<1%	0.14
	<0.1%	< **0.001**

*Activation threshold < 99% of activations.
**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 threshold 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 occurrences 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 ($p < 0.05$).

With rarity threshold < 5%, the difference between correctly and incorrectly predicted inputs is not statistically significant and the difference is relatively small. Thus, again, strong claims should be avoided, but it may be noteworthy that the numbers are somewhat similar than in the previous results with significant differences.

On average, rare co-activations are much more common in untrained inputs than in the other two scenarios (Table 15) and slightly more common in incorrectly predicted inputs than in correctly predicted inputs. Also, the minimum number of occurrences is larger in untrained inputs than in the other two, until it falls down to 0 with the lower rarity thresholds. The maximum number of occurrences is ever so slightly larger in untrained inputs than in incorrectly predicted inputs, but clearly lower than in correctly predicted inputs with the highest rarity threshold, but falls far behind with the lower ones.

5.3. CNN-shirt

In this subsection, we go through the results for the CNN-shirt model in a similar manner. CNN-shirt is otherwise similar to and similarly trained as CNN-ankle boot, but the class that was filtered out of the training set was class 6 (Shirt) instead of class 9 (Ankle boot). See Section 4.1 for more details.

As we can see from Table 16, every rarity threshold reaches statistical significance ($p < 0.05$) for activation thresholds 0 and 0.018. This means that at least two of the three groups are statistically from a different distribution and some meaningful interpretations about the differences in the occurrences of rare co-activations can be made. However, for activation threshold 0.154, only rarity threshold < 0.1% reaches statistical significance. Thus, we only present results of pairwise comparisons for groups and descriptive statistics for that rarity threshold, and disregard the higher rarity thresholds.

Activation threshold 0: For activation threshold 0, rare co-activations are more common in untrained inputs than in correctly

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	1
		correct–untrained	**0.038**
		incorrect–untrained	**0.002**
	<1%	correct–incorrect	0.507
		correct–untrained	**0.015**
		incorrect–untrained	**0.012**
	<0.1%	correct–incorrect	1
		correct–untrained	**0.034**
		incorrect–untrained	**0.0**

Table 18
Descriptive statistics for different groups in CNN-shirt with activation threshold 0.

	Correct	Incorrect	Untrained
N	8609	391	1000
Rarity < 5%			
Mean	38167.45	33424.39	42830.75
Median	11702	11950	15505.5
Max	695969	548909	701704
Min	3	150	82
Rarity < 1%			
Mean	2481.47	2910.51	3370.17
Median	45	35	67.5
Max	206947	155831	212366
Min	0	0	0
Rarity < 0.1%			
Mean	64.96	221.68	107.7
Median	0	0	0
Max	34808	30814	26338
Min	0	0	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 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.

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

Table 19
Results of pairwise comparisons between groups for CNN-shirt with activation threshold 0.018.

Activation threshold	Rarity threshold	Dunn-Bonferroni	(p)
0.018*	<5%	correct–incorrect	1
		correct–untrained	**0.008**
		incorrect–untrained	0.105
	<1%	correct–incorrect	0.639
		correct–untrained	**0.036**
		incorrect–untrained	**0.039**
	<0.1%	correct–incorrect	1
		correct–untrained	**0.001**
		incorrect–untrained	**0.026**

*Activation threshold < 99% of activations.

Table 20
Descriptive statistics for different groups in CNN-shirt with activation threshold 0.018.

	Correct	Incorrect*	Untrained
Rarity < 5%			
Mean	39685.32	35074.7	44127.99
Median	12684	12568	16815.5
Max	710863	549917	704391
Min	1	147	68
Rarity < 1%			
Mean	2530.51	2958.25	3529.03
Median	52	38	77.5
Max	213036	154045	219452
Min	0	0	0
Rarity < 0.1%			
Mean	70.07	221.27	121.64
Median	0	0	0
Max	37402	31610	27732
Min	0	0	0

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

threshold < 5%, the difference between incorrectly predicted inputs and untrained inputs is not statistically significant. As with previous cases, only scenarios with statistically significant differences should be compared with high confidence. The activation threshold is smaller than 99% of the non-zero activations in CNN-shirt with the training set.

With all rarity thresholds, untrained inputs have, on average, more rare co-activations than the correctly predicted ones (Table 20). Both mean and median are higher for the untrained inputs. Also, the minimum number of occurrences is larger in untrained inputs, when it is not 0 for both. The maximum numbers of occurrences are very close to each other with the higher rarity thresholds but larger with the lowest threshold.

Incorrectly predicted inputs do not differ statistically from either correctly predicted or untrained inputs with rarity threshold < 5% but do differ from the untrained ones with the lower thresholds. Perhaps interestingly, rare co-activations are, on average and at maximum, more common in untrained inputs with rarity threshold < 5%, but, just like with the previous activation threshold, less common with rarity threshold < 0.1%.

Activation threshold 0.154: For activation threshold 0.154, rare co-activations are more common in untrained inputs than in correctly predicted inputs. The descriptive statistics show higher averages but the differences are statistically significant only with the lowest rarity threshold. 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.

With activation threshold 0.154, only rarity threshold < 0.1% reached statistical significance between any two groups. Thus, we only present the pairwise comparisons of that rarity threshold (Table 21).

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
		correct–untrained	**0.0**
		incorrect–untrained	**0.026**

**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	78.13	229.97	146.77
Median	0	0	0
Max	39924	32371	32224
Min	0	0	0

Table 23
Results of Kruskal–Wallis tests for MLP-ankle boot.

Activation threshold	Rarity threshold	Kruskal–Wallis (p)
0	<5%	<**0.001**
	<1%	1
	<0.1%	1
0.0339*	<5%	<**0.001**
	<1%	1
	<0.1%	1
0.34**	<5%	<**0.014**
	<1%	1
	<0.1%	1

*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 incorrectly 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, untrained 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 descriptive 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 correctly predicted inputs. The descriptive statistics show higher averages but the differences are statistically significant only with the highest rarity threshold. 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.

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	**0.0**
		incorrect–untrained	**0.029**

Table 25
Cross-tabulation for different groups and number of rare co-activations in MLP-ankle boot with activation threshold 0 and rarity threshold < 5%.

N	Correct	Incorrect	Untrained
0	7829	1013	958
1	131	27	42

Table 26
Results of pairwise comparisons between groups for MLP-ankle boot with activation threshold 0.0339.

Activation threshold	Rarity threshold	Dunn-Bonferroni	(p)
0.0339*	<5%	correct–incorrect	0.063
		correct–untrained	**0.0**
		incorrect–untrained	0.084

*Activation threshold < 99% of activations.

Results of the pairwise comparisons with activation threshold 0 can be found in Table 24. The difference between untrained inputs and the other two scenarios is statistically significant ($p < 0.05$). However, the difference between correctly and incorrectly predicted inputs does not reach statistical significance. Thus, we only compare untrained inputs with the other two.

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 descriptive statistics (Table 25). On average, rare co-activations are more common in untrained inputs than in the other two scenarios. Correctly predicted inputs are approximately eight times as common as untrained inputs but correctly predicted inputs where a rare co-activation occurred are only three times as common as untrained inputs where one occurred. There is almost an equal number of untrained inputs and incorrectly predicted inputs but nearly double the number of untrained inputs where a rare co-activation occurred. However, rare co-activations are overall not very common in any scenario.

Activation threshold 0.0339: For activation threshold 0.0339, rare co-activations are more common in untrained inputs than in correctly predicted inputs. The descriptive statistics show higher averages but the differences are statistically significant only with the highest rarity threshold. 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.

Results of the pairwise comparisons with activation threshold 0.0339 can be found in Table 26. The only statistically significant difference ($p < 0.05$) is measured between untrained inputs and correctly predicted inputs. Incorrectly predicted ones do not differ from either of the other two scenarios to a statistically significant degree. Thus, we will only compare the correctly predicted inputs with the untrained ones. The used activation threshold is < 99% of non-zero activations in MLP-ankle boot with the training set.

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	7842	1014	962
1	118	26	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	0.131
		incorrect–untrained	**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	7960	1040	1000
Mean	0.36	0.3	0.37
Median	0	0	0
Max	9	6	6
Min	0	0	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 descriptive 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 discuss what the differences mean for the usage of rare co-activations in error detection to achieve fault tolerance in systems utilizing neural networks.

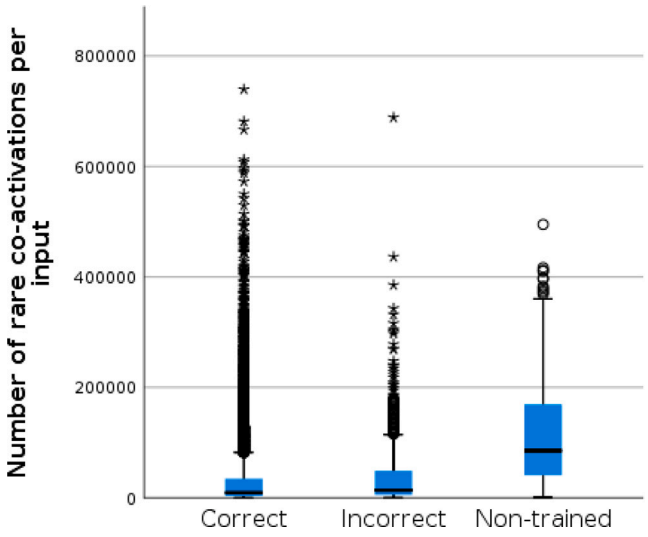

Fig. 4. Boxplot figure of the number of rare co-activations per input in CNN-ankle boot with activation threshold 0 and rarity threshold < 5%.

6.1. Differences between scenarios (RQ1)

The three scenarios (correctly predicted, incorrectly predicted, and untrained inputs, presented in Section 3.2), differ in the occurrences of rare co-activations. For every model, at least some of the chosen combinations of activation and rarity thresholds produced statistically significant differences between the distributions of rare co-activations in the scenarios. This assures that rare co-activations as a concept is something to look into further.

Considering our viewpoint, we are most interested in how correctly predicted inputs differ from incorrectly predicted and untrained ones. Overall, rare co-activations are more common in untrained inputs than correctly predicted ones and they may be slightly more common in incorrectly predicted ones as well (for example, cf. Fig. 4). Both of these views are elaborated below.

Rare co-activations are, on average, more common in untrained inputs than in correctly predicted ones. This is the most consistent result we obtained across all models. In the vast majority of tests, the difference between the distributions of these two scenarios is statistically significant, and not once do correctly predicted inputs have a higher mean value of occurrences. The median can be 0 for both, but it is never higher for correctly predicted inputs. Thus, a larger number of rare co-activations is related to never-seen-before inputs.

Rare co-activations seem to be slightly more common in incorrectly predicted inputs than in correctly predicted ones, but this result is not as strong as the previous one. The biggest reason for downplaying the result is that the difference between the two scenarios is less frequently statistically significant. However, in support of this claim rare co-activations are more common in incorrectly predicted inputs in every statistically significant case. Also, rare co-activations tend to be slightly more common in cases where the difference is not statistically significant, even though this is not a given. As such, some evidence supports this idea, but making very strong claims for it would be ill-advised.

A few factors concerning the incorrectly predicted inputs worth considering in further studies are the amount of data and how the co-activation rates are computed. As the difference in the descriptive statistics tends to be relatively small between correctly and incorrectly

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 incorrectly. 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 activation 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 significant 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. Especially 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, perhaps 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 between CNN-ankle boot and CNN-ankle boot9. The numbers tend to be slightly smaller for CNN-ankle boot9, but the differences between the scenarios are similarly significant, along with the trends in the descriptive statistics.

6.2. Discussion of the implications for error detection and fault tolerance (RQ2)

When considering the usefulness of rare co-activations in error detection to achieve fault tolerance, the types of misbehaviour to be targeted with it must be considered. As discussed in Section 3.2, we consider whether drift in input data can be detected by utilizing the rare co-activations, whether a single input can be detected as something the network is not trained to handle, and whether an incorrect prediction can be detected. Below, we discuss how the rare co-activations would fit these tasks based on our results.

As to detecting drift in incoming data over a period of time, rare co-activations show great promise. For every used network, we found more than one combination of activation and rarity thresholds for which the average number of rare co-activations was largest for untrained inputs and the difference was statistically significant. Not only that, but the difference was often quite large, especially for larger networks, and it was not dependent on the input class that was excluded from the training phase. Based on this, drift in incoming data could be monitored by monitoring the number of rare co-activations: if the numbers per input grow, it could indicate drift.

Detecting untrained inputs on a level of a single input is not so straightforward. While rare co-activations are, on average, more common in untrained inputs, other scenarios do have outliers that can be as high or even higher than the maximum number of occurrences in untrained inputs (cf. Fig. 4). Thus, the approach is prone to false positives, where inputs that the network should be able to handle are flagged as inputs that the network is not trained for. Finding a higher threshold for how many rare co-activations must occur before the input is flagged would decrease the number of false positives, but would introduce more false negatives, where untrained inputs would not be flagged. The smaller MLP presents a special case of this, as rare co-activations are very uncommon overall, and the number of false negatives would be very large. This does not necessarily mean that rare co-activations cannot be used to flag single inputs, but it would probably necessitate work for finding appropriate thresholds for activation, rarity, and the number of occurrences, so that the result number of false positives and false negatives is tolerable — and an application which allows some of them.

Detecting incorrectly predicted inputs is less likely to be relevant based on our data. The differences between correctly and incorrectly predicted inputs are so small that they are not statistically significant in every network with our sample. Additionally, even if the differences were significant, they tend to be so small that it is arguable whether they are relevant when used to achieve fault tolerance. In other words, finding an appropriate number to be used as a threshold for flagging the result becomes very difficult and the usefulness of detecting single incorrect predictions suffers.

Usefulness could improved with regards to the approaches discussed here. In this paper, we have treated the rare co-activation as equals. However, we do not exactly know that this is the case, which could be suggested by the outliers where the networks predict correctly despite a large number of rare co-activations. As such, certain kinds of rare co-activations may possibly be more indicative of incorrect predictions or untrained inputs. Detecting these co-activations could enable improving the usefulness of rare co-activations for detecting untrained or incorrectly predicted inputs even on the single-input level. Potential approaches for future research could include comparing rare co-activations that happen across layers to those that happen within a layer, rare co-activations that occur early in the model to those that happen in later layers, rare co-activations that occur in convolutional

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% certainty 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 paper, 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 theoretical 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 believe we have produced believable and meaningful results, but fine-tuning our ideas could produce even better results, which, arguably, suggests that we do not have full understanding of the concepts — yet.

External validity refers to the generalizability of the results. To make the results more generalizable, we used multiple networks with various structures and different classes excluded from the training set. We believe we have answered the most imminent questions of what would happen to the results should the setting be altered. The networks not achieving extremely high accuracy could be seen as a threat to external validity. However, industrial ML models do not necessarily achieve high accuracy either. For example, De Clercq et al. tested multiple ML approaches to predict biogas production in an industrial plant, and the best model achieved an accuracy of 87% [26]. Thus, we consider the accuracies of our networks believable and do not believe this threat to be fatal to our results. Having said that, it would be interesting to see whether correctly and incorrectly predicted inputs differed from each other more in a network with extremely high accuracy, but gaining statistical relevance in that situation would probably require a larger data set.

As to the data set used, the choice of using Fashion-MNIST is another threat to generalizability. Fashion-MNIST is widely used in research literature, and many consider it to be somewhat challenging to ML models. However, it is by no means an industrial data set. Because of the wide use of Fashion-MNIST, we do not believe this to be a fatal threat to our results, but we do believe that generalizability would benefit from the approach being applied in an actual industrial network, trained with industrial data.

8. Conclusions

We have presented a study that defines the concept of co-activation rate, investigates how rare co-activations manifest in correctly or incorrectly predicted inputs a neural network has been trained to handle and inputs it has not been trained for, and, based on the previous note, how the rare co-activations could be used in runtime risk mitigation as a tool to detect errors and promote fault tolerance. To produce the results, we first trained multiple different neural networks, after which co-activation rates were computed for each node. Using a separate test set and the co-activation rates, we counted how many times rare co-activations occurred for every input, and then labelled the input as a correctly or incorrectly predicted one if it belonged to a class that was present in the training set or an untrained input if it belonged to a class that was excluded from the training set.

Rare co-activations are more common in untrained inputs than in inputs that the network was trained to handle, and especially the ones that the network predicted correctly. Thus, monitoring rare co-activations over time could be used to monitor drift in the incoming data. If the number of rare co-activations per input rises, the share of inputs the network was not trained for also rises. However, detecting whether a single input is something the network is trained to handle is a bit trickier. This is mostly because the trained inputs, including the ones the network predicts correctly, also include few inputs with large numbers of rare co-activations.

The difference between correctly and incorrectly predicted inputs is not so clear. There is a tendency that rare co-activations occur slightly more often in incorrectly predicted inputs but the difference tends to be smaller than when comparing with untrained inputs. Thus, detecting incorrect predictions based solely on rare co-activations may not be feasible with this approach. However, as the number of occurrences tends to be slightly higher in incorrectly predicted inputs, trying to find which kind of rare co-activations are the most indicative of incorrect prediction could be a worthwhile research question for the future. This could mean, for example, studying whether rare co-activations on earlier or later layers or rare co-activations across layers is more indicative.

Additionally, the results would benefit from more empirical follow-up studies. Even if Fashion-MNIST is widely used, it is not an industrial data set. An actual industrial setting would provide an even stronger indication of the usefulness of the results we have found.

CRediT authorship contribution statement

Lalli Myllyaho: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Jukka K. Nurminen:** Conceptualization, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Tommi Mikkonen:** Conceptualization, Writing – 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. For full disclosure statements refer to https://doi.org/10.1016/j.array.2022.100201.

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.
[2] Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H. State-of-the-art in artificial neural network applications: A survey. Heliyon 2018;4(11):e00938.
[3] Pei K, Cao Y, Yang 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.
[4] Tian Y, Pei K, Jana S, 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.
[5] Xie X, Ma L, Juefei-Xu F, Xue M, Chen H, Liu Y, et al. 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.
[6] Ma L, Juefei-Xu 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.
[7] IEEE standard dictionary of measures of the software aspects of dependability. IEEE Std 982.1-2005 (Revision of IEEE Std 982.1-1988), 2006, p. 1–41.
[8] Avizienis A, Laprie J-C, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans Dependable Secure Comput 2004;1(1):11–33.
[9] Ramanathan A, Pullum LL, Hussain F, Chakrabarty D, Jha SK. Integrating symbolic and statistical methods for testing intelligent systems: Applications to machine learning and computer vision. In: 2016 Design, automation & test in Europe conference & exhibition. IEEE; 2016, p. 786–91.
[10] Tsymbal A. The problem of concept drift: Definitions and related work. Comput Sci Dep, Trinity College Dublin 2004;106(2):58.
[11] Wang S-C. Artificial neural network. In: Interdisciplinary computing in Java programming. Springer; 2003, p. 81–100.
[12] Sharma S, Sharma S, Athaiya A. Activation functions in neural networks. Int J Eng Appl Sci Technol 2020;4(12):310–6.
[13] Cheng C-H, Nührenberg G, Yasuoka H. Runtime monitoring neuron activation patterns. In: 2019 Design, automation & test in Europe conference & exhibition. IEEE; 2019, p. 300–3.
[14] Denzin NK. Sociological methods: A sourcebook. Routledge; 2017.
[15] Kruskal WH, Wallis WA. Use of ranks in one-criterion variance analysis. J Amer Statist Assoc 1952;47(260):583–621.
[16] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. 2017, arXiv preprint arXiv:1708.07747.
[17] Ackerman S, Farchi E, Raz O, Zalmanovici M, Dube P. Detection of data drift and outliers affecting machine learning model performance over time. In: JSM proceedings. American Statistical Association; 2020, p. 144–60.
[18] Kayed M, Anter A, Mohamed H. Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture. In: 2020 International conference on innovative trends in communication and computer engineering. IEEE; 2020, p. 238–43.
[19] Gobert C, Reutzel EW, Petrich J, Nassar AR, Phoha S. Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. Addit Manuf 2018;21:517–28.
[20] Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In: 2017 International conference on engineering and technology. Ieee; 2017, p. 1–6.
[21] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR; 2015, p. 448–56.
[22] Scherer D, Müller A, Behnke S. Evaluation of pooling operations in convolutional architectures for object recognition. In: International conference on artificial neural networks. Springer; 2010, p. 92–101.
[23] Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: A tutorial. Computer 1996;29(3):31–44.
[24] Dunn OJ. Multiple comparisons using rank sums. Technometrics 1964;6(3):241–52.
[25] Shadish WR, Cook TD, Campbell DT, et al. Experimental and quasi-experimental designs for generalized causal inference/William R. Shedish, Thomas D. Cook, Donald T. Campbell. Boston: Houghton Mifflin; 2002.
[26] De Clercq D, Jalota D, Shang R, Ni K, Zhang Z, Khan A, et al. Machine learning powered software for accurate prediction of biogas production: A case study on industrial-scale Chinese production data. J Cleaner Prod 2019;218:390–9.