Optimization of novelty detection through the use of information entropy

Rasmus Vedel[1]

[1]**Supervisor: Christian Schilling**

ABSTRACT

Modern neural networks can achieve high confidence levels of categorization of well-known classes. However, they fall short when trying to categorize classes not part of their training data. This paper proposes a method of optimizing the runtime of a previously proposed outside-the-box novelty detection method, which detects inputs of an unknown nature by monitoring the output of layers in the model on training data and comparing it to the output on any incoming data. The method achieves a reduction in runtime by reducing the amount of dimensions checked in the data through the use of Shannon entropy.

1 SUMMARY

Neural networks have emerged as a cornerstone of state-of-the-art machine learning and have been an important part of many advancements in a diverse set of fields such as computer vision, natural language processing, robotics, and healthcare. With the emergence of ai-assistants they have also become an important part of the everyday functioning for a lot of people across the globe.

It is therefore becoming increasingly important to identify and find ways around their limitations. One of these limitations is the fact that by definition any neural network is required to make a prediction based on what it has learned.

This becomes a problem when a neural network receives an input that is not within the scope of the training data it received for training, as it will still produce an output that will be based on the training data and therefore likely wrong.

In the context of classification neural networks this means that the network will always make a guess based on its learned classes.

This is where the research field of novelty detection becomes important, as the goal of novelty detection is detection of data points that deviate from the normality. We can therefore use this to make a neural network capable of identifying inputs of an unknown nature (novelties).

The method proposed in this paper improves on the outside-the-box novelty detection method proposed by Henzinger et al. (2020). This will be done by reducing the overall output dimensions checked by skipping dimensions that have a high entropy value when running the method on training data

The contributions of this paper can be summarized as follows: We propose a entropy-based approach to optimizing the outside-the-box method by reducing the number of dimensions checked using an entropy threshold. The result is a novelty detection method with faster runtime and close to the same accuracy and novelty detection errors.

2 RELATED WORK

Novelty detection

As stated novelty detection is an important research field within the topic of machine learning. Therefore a lot of different approaches have come forth to tackle the problem. Generally however approaches build some model of a training set that is selected to contain no examples (or very few) of the important (i.e., novel) class. Novelty scores are then assigned to the data and deviations from normality are detected according to a decision boundary that is usually referred to as the novelty threshold [Pimentel et al. (2014)].

As for the use of entropy in the field of novelty detection, entropy is mostly used to measure the distance from a probability distribution constructed from the training data to one constructed from the input. One example is Lai et al. (2023), where a probability distribution is created of the test data values within the latent space. Then Kullback–Leibler divergence (relative entropy) is used to measure the difference between probability distributions generated by any incoming data. Then if a significant difference is found the input is marked as a novelty.

Outside-the-box monitoring

The work made to produce the outside-the-box monitoring novelty detection method [Henzinger et al. (2020)] has been essential to this project and their follow-up work [Lukina et al. (2021); Kueffner et al. (2023)] which proposes methods of returning confidence levels of inputs being novelties.

Anomaly Detection

Anomaly detection differs from novelty detection in the way that it focuses on identifying data points that do not conform to the expected pattern, whereas novelty detection techniques attempt to identify unknown data.

One approach for anomaly detection that employs the same techniques as the method proposed in this paper is [Bereziński et al. (2015)]. Here Shannon entropy is used to detect modern botnet-like malware based on anomalous patterns in the network.

Entropy in machine learning

The use of entropy has seen widespread use in the field of machine learning, some examples are as follows:

Loss functions, where cross-entropy and KL-divergence see heavy use in both old as well as cutting-edge machine learning models.

Regularization, where one entropy-based method is Maximum entropy regularization that sees some use such as [Cheng et al. (2020)] where it is used to create robustness in a chinese text recognition application.

All of these have served as an inspiration for the method proposed in this paper.

3 OUTSIDE-THE-BOX MONITORING

The method presented in this paper expands upon the idea proposed in Henzinger et al. (2020). This section will briefly explain the outside-the-box method. This method works by monitoring the outputs of at least one layer ℓ. To simplify the explanation and implementation of the outside-the-box method, a monitor of only one layer will be implemented.

After training the model normally, the training data is used as input to a truncated model of the initial model with ℓ as its output layer. This results in n-dimensional data points, where n is equal to the number of nodes in ℓ.

To cover scenarios where the data points are split into different regions, the data points for each class are then clustered using a clustering algorithm. After this, the clusters are used to construct bounding boxes (just referred to as boxes) that represent the maximum and minimum value of the data points of that cluster on each of the n dimensions.

These boxes are then used to detect unfamiliar outputs (novelties) by the assumption that any input in the model, resulting in an output from ℓ that is a data point outside of any of the defined boxes for any of the known classes is something unfamiliar and therefore a novelty and can be discarded as such.

An example of 2 dimensions of some boxes and their points can be seen in Figure 1.

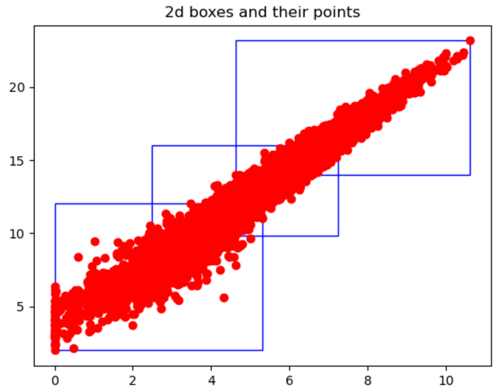

Figure 1. Representation of 2 dimensions of boxes of one class and their data points

4 DIMENSIONALITY-REDUCTION USING ENTROPY

Now that we have some boxes representing the different likely data point placements of known classes we want to discover ways to reduce the number of dimensions for each class.

The idea here is to look for dimensions where the spread of points is so big that most incoming data points will be within the boxes for that dimension regardless of whether they are the class in question.

The proposed method is to use entropy or more specifically information entropy, sometimes known as Shannon's entropy, to measure the spread of the data points of each dimension, then if the entropy is above a certain threshold that dimension is disabled when doing outside-the-box monitoring.

Discretization of Continuous Values

As the values of the data points are obtained from the outputs of a neural layer, they will be continuous and almost guaranteed to be unique. So to make the probability distribution required for Shannon's entropy work, the values must be converted into something discrete. Therefore for each dimension of the data points for every box of each class, we discretize the continuous values into M bins using the NumPy histogram routine [1]. The histogram provides the density values D_i within each bin, and the boundaries B_i of all of the bins edges for the i-th dimension. E.g. if you have a histogram with 2 bins spanning the range from 2 to 4 the bin boundary edges would be 2, 3, and 4 and the bin densities would be the number of data points in each bin.

Calculation of Probabilities

The probability P_i for each bin in dimension i is obtained by multiplying the density by the width of the bin, which we get from the difference of the boundaries of the two bins:

$$P_{i,j} = D_{i,j} \cdot W_{i,j}$$

where $D_{i,j}$ is the j-th bin count in the histogram, and $W_{i,j}$ is the width between the two bins, calculated with the following formula: $W_{i,j} = (B_{i,j+1} - B_{i,j})$ Where $B_{i,j}$ and $B_{i,j+1}$ are the bin boundary edge locations of the j-th and $j+1$ edges.

Entropy Calculation

The entropy H_i, for data point values in the i-th dimension ($\mathbf{X}_i$), is then calculated using the function for Shannon's entropy:

$$H(X_i) = -\sum_{j=1}^{M} P_{i,j} \log(P_{i,j} + \varepsilon)$$

[1] https://numpy.org/doc/stable/reference/generated/numpy.histogram.html

Here, ε is a small constant added to avoid taking the logarithm of zero and

$$P_{i,j}$$

is the previously found probabilities for each bin.

Finding the Threshold

Now that we have an entropy value $H(X_i)$ for each dimension representing the spread of the points in that dimension, we have to set a threshold value T for $H(X_i)$ such that if $H(X_i) > T$ dimension i is disabled in the corresponding box. This threshold is found by choosing the highest value that does not seem to impact the accuracy of the novelty detection too much and is found using a trial-and-error strategy.

5 EXPERIMENTAL EVALUATION

In this section, the method and setup of the experiments listed in this paper are elaborated upon. For all results to be somewhat consistent and not be affected by variables outside the scope of these experiments, all experiments were conducted using the same dataset, model, loss function, etc. These will be presented in this section. The results of the experiments can be found in section 5.1 and a simple flow-chart of the overall experimental process can be seen in figure 2. It should be noted that the implementation is general and parameters can be easily changed, though doing so might affect the outcome.

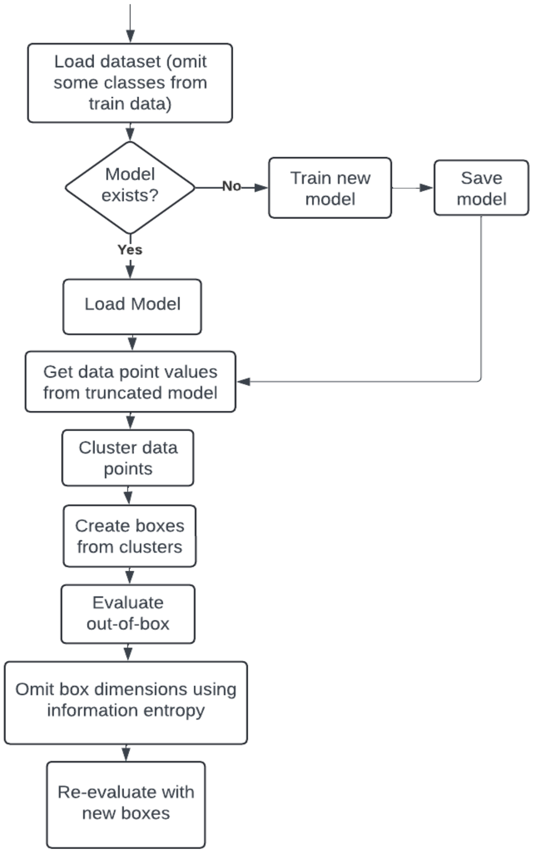
Load dataset (omit some classes from train data)
Model exists?
No
Train new model
Save model
Yes
Load Model
Get data point values from truncated model
Cluster data points
Create boxes from clusters
Evaluate out-of-box
Omit box dimensions using information entropy
Re-evaluate with new boxes

Figure 2. Overall flow-chart of the experiment

The MNIST dataset [Deng (2012)] was used for all experiments. This includes 70000 grayscale images of handwritten 0-9 digits, split into 60000 train images and 10000 test images. for the experiments, the images and class labels of the 9 digit are then omitted from the train data to create a novelty in the dataset, resulting in a final size of 54051 train images. The test data is kept as is.

A model is then loaded, which for all of the experiments was the ResNet50V2 model [He et al. (2016)] as a base model, which is a premade model consisting of several convolutional layers and is located with no pretrained weights as these would be based on a another dataset. The base model is followed with a flatten layer and then a dense layer with 128 neurons using relu activation function and with an output layer with softmax activation function on top. The model is compiled with the Adam optimizer and categorical-crossentropy as its loss function with the default keras learning rate of 0.001

The model is then trained using the train data over a total of 5 epochs and data point values for each class are gathered from the last hidden layer, meaning the last layer before the output layer. This layer consists of 128 nodes, meaning outputted data points will be of 128 dimensions.

Now all the data point values are clustered using the k-means clustering algorithm [Hartigan and Wong (1979)] with k=3 and boxes are made using each cluster. We now have the outside-the-box prerequisites and run novelty detection using the boxes to get evaluation metrics such as the accuracy of the novelty detector, the accuracy of the model's predictions after the novelty detection, and the run time of the algorithm.

The method of omitting dimensions deemed unnecessary mentioned in section 4 is then applied with the number of bins set to 30 and an entropy threshold $T = 3.0$ to reduce the dimensions of the boxes of each class. The evaluation is then rerun using the updated boxes.

5.1 Results

This section will present the different results found while conducting the experiments on both the outside-the-box method and the results for the experiments after conducting the entropy dimensionality reduction.

Novelty detection results

For the first results, we have the total amount of novelties reported using the boxes before (marked as out_of_box) and after the dimensionality reductions (marked as entropy-reduced boxes). These can be seen in Figure 3.

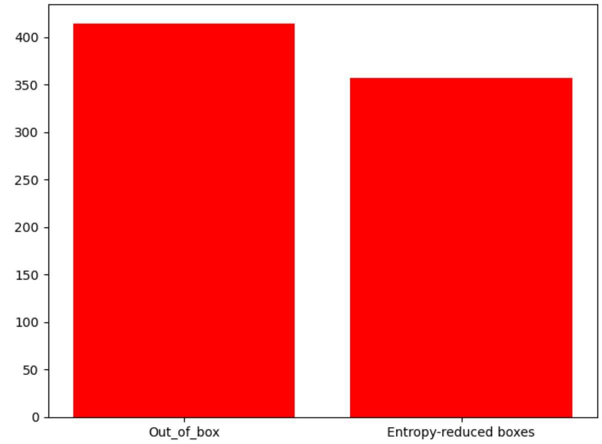

Figure 3. Bar charts for the amount of novelty detections

These indicate a loss of total novelty detections after the entropy reduction, which is somewhat expected as with fewer dimensions to check for novelties, fewer novelty detections will be made. It could also mean that some of the dimensions removed using the entropy reduction had such a high uncertainty that without those dimensions removed, data points would regularly fall outside the boxes of those dimensions.

To compare the quality of the novelty detections of both before and after the reductions the outcomes will be categorized into 3 different categories: false positive, false negative, and true positive. The meaning of these are as follows:

- **False Positive:** When a data point that would otherwise have been a right prediction is marked as a novelty

- **False Negative:** When a data point is predicted wrongly and not marked as a novelty
- **True Positive:** When a data point that would otherwise have been a wrong prediction is marked as a novelty

It stands to reason that the goal is to have as many true positive results as possible. As for the false positives and false negatives, both are considered a fail and which of these is the best depends on what system the novelty detection method is applied to.

The results of the novelty detection tests in the form of grouped bar charts can be seen in Figure 4 and in the form of a stacked bar chart can be seen in Figure 5.

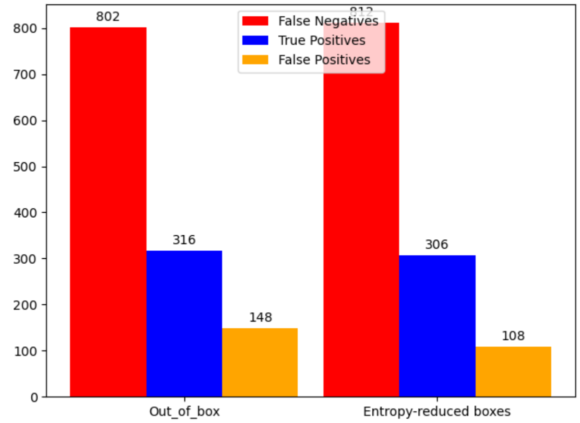

Figure 4. Grouped bar charts of the two novelty detection tests

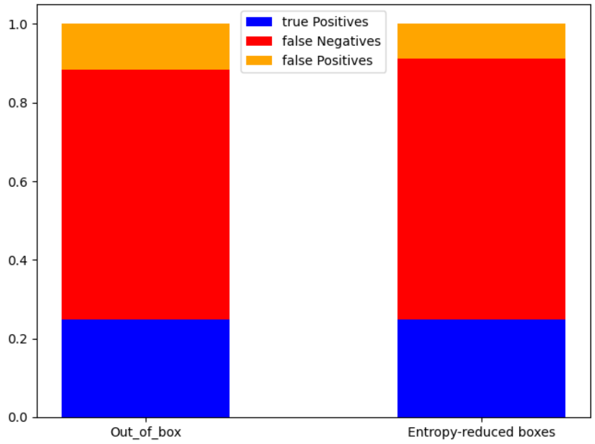

Figure 5. Stacked bar charts of the two novelty detection tests

The fall in the amount of False positives and true positives when using entropy-reduced boxes is most likely because as the total amount of novelty detections falls, fewer inputs that would otherwise have been right or wrong predictions are marked as novelties.

As for the rise in false positives when using entropy-reduced boxes, this is likely due to the fall in the total number of novelty detections resulting in fewer wrongly predicted inputs being marked as novelties (meaning more false negatives).

In the stacked bar chart (Figure 5), the ratio of the results can be seen. These are calculated by taking the number of each of the result categories and dividing it by the total number of results. In the figure it can be hard to see the difference between the two results, except for the false negatives ratio being a bit higher for the entropy-reduced boxes results and the false positives a bit lower.

If you include a fourth outcome of **True Negatives**, meaning all outcomes where the prediction is true and not marked as a novelty, the figures become as can be seen in Figures 6 and 7.

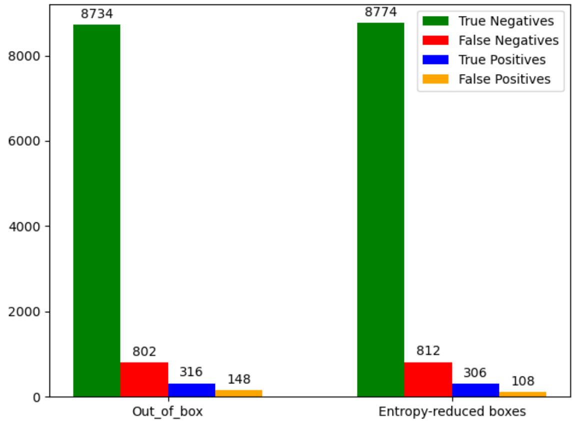

Figure 6. Grouped bar charts of the two novelty detection tests with true negative outcome

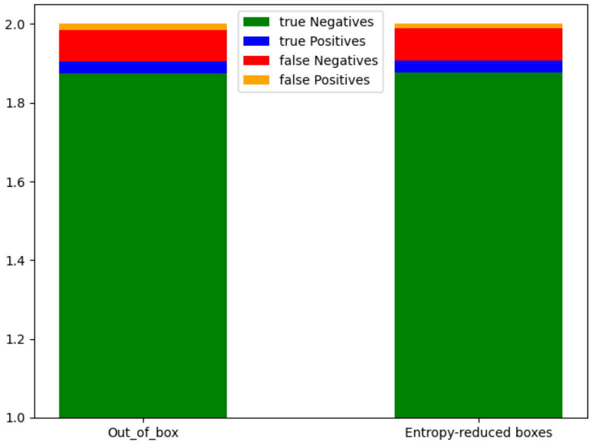

Figure 7. Stacked bar charts of the two novelty detection tests with true negative outcome

These figures show an increase in true negatives after the entropy reduction, which again is probably a result of the lesser amount of novelty detections, resulting in some of the false positives becoming true negatives as less right predictions are marked as novelties.

The results of these tests seem to indicate that at the chosen entropy threshold, the dimensionality reduction does not have any major impact on the novelty detection ratio although it does have an impact on the total amount of novelty detections made.

Accuracy results

When looking at the accuracy results, two things will be measured:

1. The prediction accuracy of any inputs not marked as novelties.

2. The novelty accuracy of detecting any input of an unknown class (These defer from true positives as those are any inputs that would be wrongly classified)

The accuracy results can be seen in Figure 8

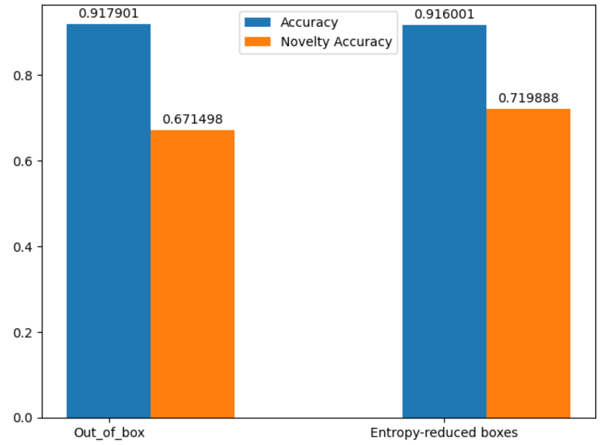

Figure 8. Accuracy bar charts

This figure indicates that the prediction accuracy after novelty detection stays the same after reducing the number of dimensions in the boxes.

The Novelty Accuracy increase after the dimensionality reduction seems to indicate that the important dimensions for detecting the actual unknown classes (and not just the true positives, which are any inputs that would be wrongly classified) are kept after the reduction. This in turn with the overall fall in novelty detections, as seen in the novelty detection results, presented earlier, means that the overall novelty accuracy increases.

Runtime results

As for the runtime results of both approaches, the runtime of the novelty detection part of the approaches on the test data of 10000 mnist images was measured. The runtime results in seconds can be seen in Figure 9.

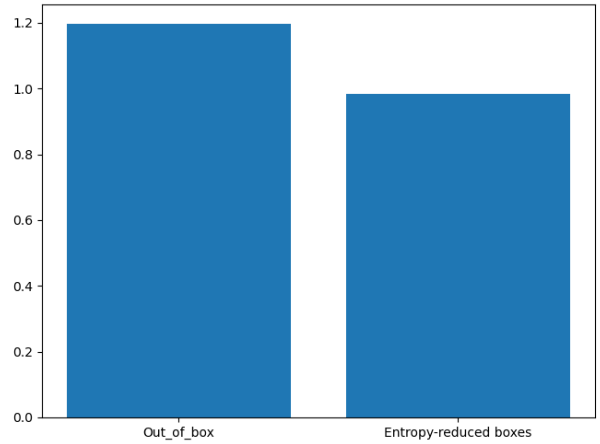

Figure 9. The runtime of the novelty detection methods

These results indicate a reduction in the runtime of the novelty detection (outside box test) of the dimensionality-reduced boxes. This is as expected, as the logic for checking if a given point is within the box margins is skipped on some dimensions.

It should be noted that the overall run time has not changed much in the experiments as it is still required to run the inputs on most of the model to get the output of layer ℓ. This takes far more time than the novelty detection itself when running with a big model such as ResNet50V2 used for the experiments. The overall runtime including getting the inputs from ℓ can be seen in Figure 10.

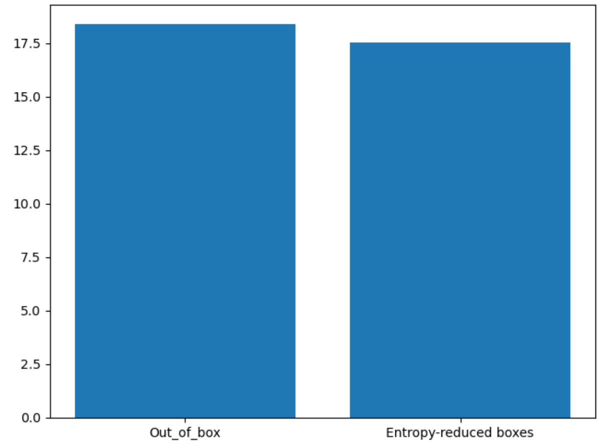

Figure 10. The overall runtime of both methods

It should also be noted that the time measurement is also dependent on a lot of other things such as what else the computer is doing at the run time of each experiment and computer hardware specifications, therefore results can vary.

6 CONCLUSION

With the widespread use of machine-learned tools, the importance of high-quality methods of novelty detection is more important than ever. The proposed method in this paper provides one method of novelty detection working on any type of input to the system, though a limitation is it is dependent on having the correct entropy threshold.

The results in this paper also demonstrates the effectiveness of using entropy as a criterion for dimensionality reduction in novelty detection frameworks, offering an increase in performance that can be important in systems where computational efficiency is crucial such as real-time applications.

6.1 Future work

Dynamic Entropy Threshold

As stated one of the limitations of the proposed method is that it is based on an entropy threshold where the optimal value will change based on a lot of parameters such as how well the model is trained. One way to fix this could be to decide the entropy threshold based on the training data.

This would work by looking at the different entropy values of all the dimensions of all the boxes and then calculating the threshold based on that data. One simple way to calculate could be to take some percentile of the entropy data and then set the threshold to that.

REFERENCES

Bereziński, P., Jasiul, B., and Szpyrka, M. (2015). An entropy-based network anomaly detection method. *Entropy*, 17(4):2367–2408.

Cheng, C., Xu, W., Bai, X., Feng, B., and Liu, W. (2020). Maximum entropy regularization and chinese text recognition. *CoRR*, abs/2007.04651.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Hartigan, J. A. and Wong, M. A. (1979). A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. *CoRR*, abs/1603.05027.

Henzinger, T. A., Lukina, A., and Schilling, C. (2020). Outside the box: Abstraction-based monitoring of neural networks. *ECAI 2020*.

Kueffner, K., Lukina, A., Schilling, C., and Henzinger, T. A. (2023). Into the unknown: active monitoring of neural networks (extended version). *Int. J. Softw. Tools Technol. Transf.*, 25(4):575–592.

Lai, C.-H., Zou, D., and Lerman, G. (2023). Novelty detection via robust variational autoencoding.

Lukina, A., Schilling, C., and Henzinger, T. A. (2021). Into the unknown: Active monitoring of neural networks. In Feng, L. and Fisman, D., editors, *RV*, volume 12974 of *LNCS*, pages 42–61. Springer.

Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99:215–249.