How to Make Your Classifier Safe

A story about the accuracy estimation of Machine Learning/Deep Learning classifiers based on statistical distance measures (SafeML, and SafeDL) — Part I.

Koorosh Aslansefat

June 2020

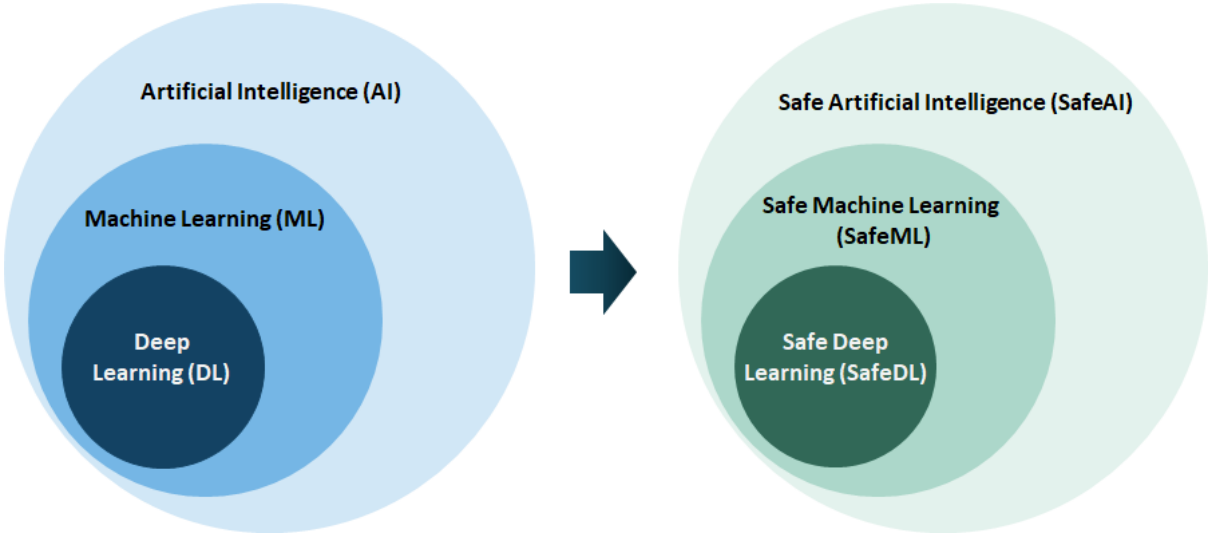
Table of Contents

1 Introduction

Nowadays, artificial intelligence (AI) is growing rapidly and its applications dominating many different subjects. In parallel with this rapid growth, the concerns about AI safety is also raising. For safety-critical systems where human life, environment, money and privacy is under risk, the AI safety cannot be ignored. (Amodei et al., 2016) have discussed different existing issues of certifying the modern machine learning systems operating in the field. As shown in the following figure, safety issues can be categorised into five categories including A) Safe exploration, B) Robustness to distributional shift, C) Avoiding negative side effects, D) Avoiding "reward hacking" and "wire heading", E) Scalable oversight.

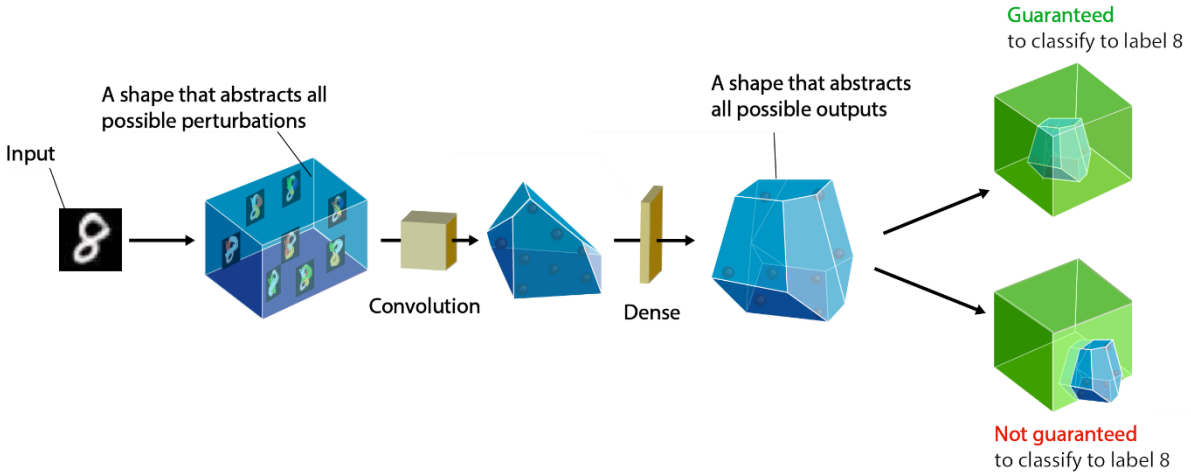
Five main categories of AI Safety

The topic of this story can be considered as "robustness to distributional shift" issue in AI safety. Based on the following figure, the SafeML approach that we are going to explain includes Safe Machine Learning (SafeML) and Safe Deep Learning (SafeDL).



Left: Artificial Intelligence (AI) vs. Machine Learning (ML) vs. Deep Learning (DL). Right: Safe Artificial Intelligence (SafeAI) vs. Safe Machine Learning (SafeML) vs. Deep Learning (SafeDL).

There are different existing approaches for increasing the safety and robustness of ML algorithms. Some papers investigate the uncertainty evaluation of results in a classifier while others focus on the improvement of robustness against uncertainties. As an example, the following figure shows the ETH Robustness Analyzer for Neural Networks (ERAN) that uses possible perturbations for input "8" and tries to create a shape that abstracts all possible outputs. If the created shape violates the defined boundary and the results cannot be certified. Otherwise, the outputs will be guaranteed. For more details please check (Gehr, T., et al. 2018 and Balunovic, M., et al. 2019).



ETH Robustness Analyzer for Neural Networks (ERAN) [https://github.com/eth-sri/eran]

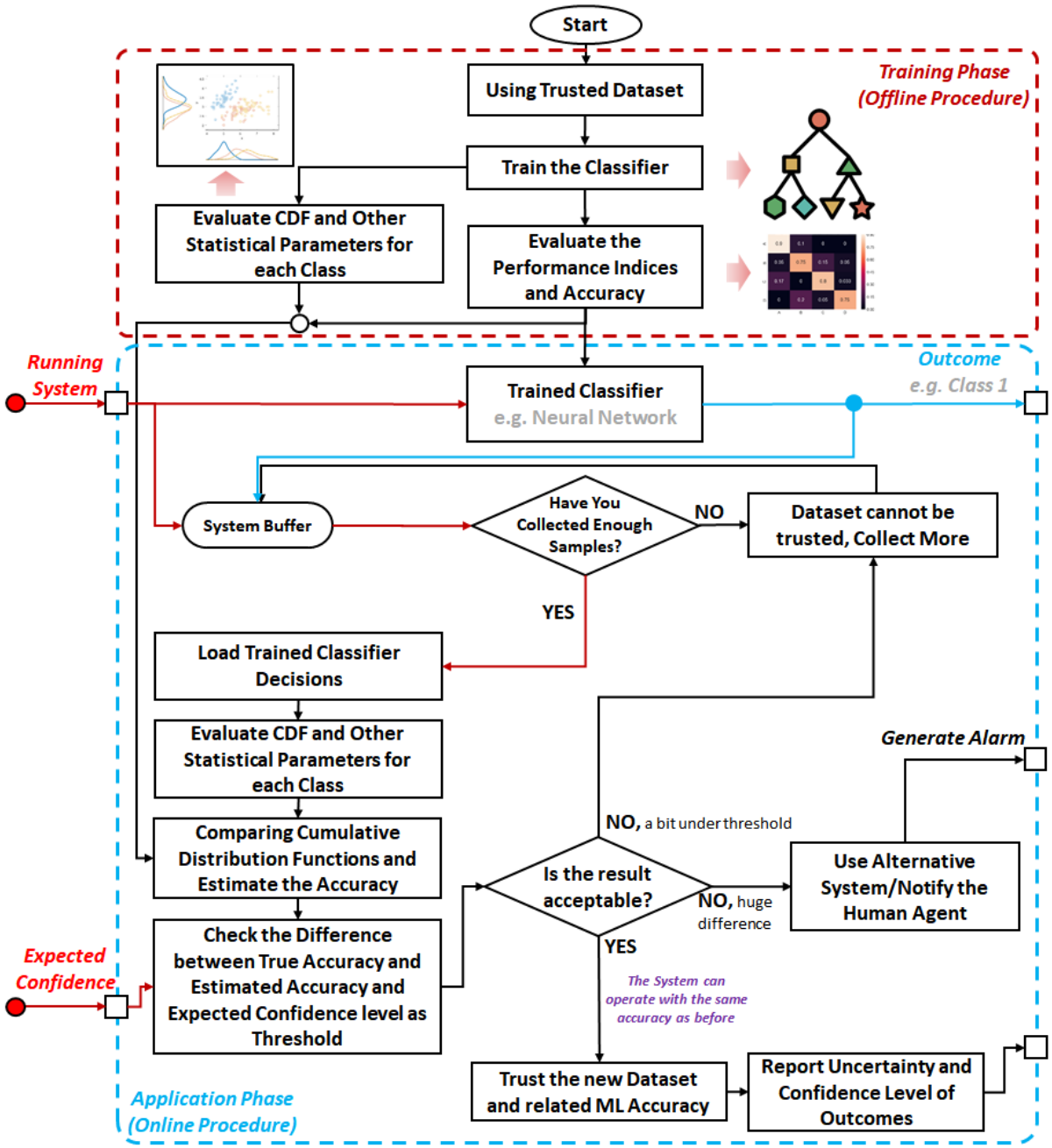
In section 2, the idea of SafeML is discussed briefly, and section three addresses the application of statistical difference measures with some python example. A short conclusion is provided in section 5. Some of the related medium posts and Github projects are suggested at the end of the story.

2 SafeML Idea

SafeML idea has been proposed by (Aslansefat et al., 2020-b), and the goal was to somehow monitor the decisions of the classifiers when there is no available label. The following figure demonstrates the flowchart of the SafeML idea. In this flowchart, there are two main sections including training phase and application phase.

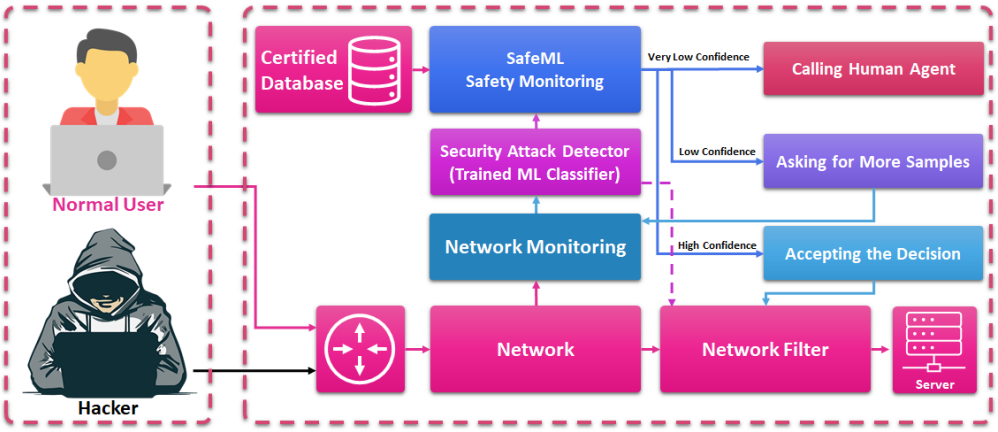
A) **The training phase** is an offline procedure in which a trusted or certified dataset will be used to train the intelligent algorithm that can be a machine learning or deep learning algorithm. Thus, using a trusted dataset the classifier will be trained and its performance will be measured with existing KPIs (e.g. ROC, Accuracy and Kappa). Meanwhile, the statistical parameters and distributions of each class will be estimated and stored to be used for comparison (e.g. Mean values, Variance Values, and the Empirical Cumulative Distribution Functions (ECDFs)).

B) **The application phase** is an online procedure in which real-time and unlabelled data is going to be feed to the system. For example, consider a security attack detector that has been trained to detect different security attacks and it should filter the attacker IPs. Therefore, in the application phase, the trained classifier should distinguish between the normal network traffic and the security attacks (classification task). One important and critical issue in the application phase is that the data does not have any label. So, it cannot be assured that the classifier can operate as accurate as of the training phase.



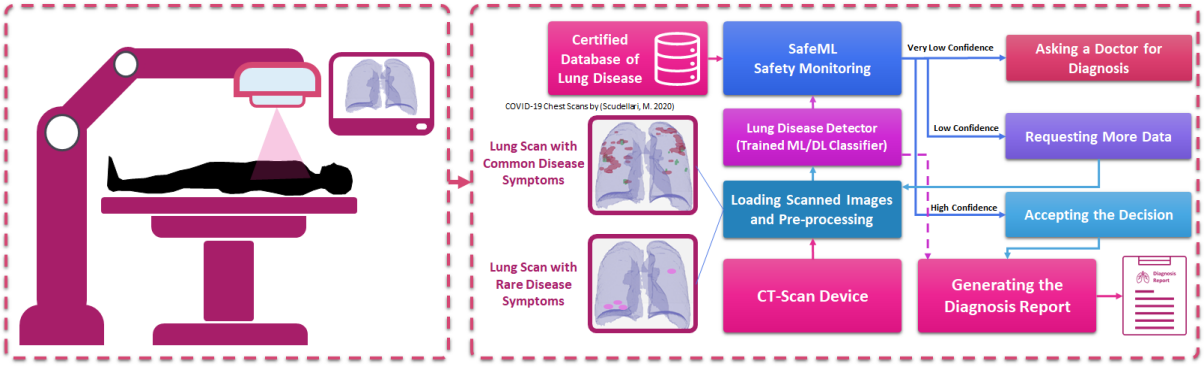
Flowchart of SafeML (Aslansefat et al. 2020-b)

In the application phase, the buffered data will be separated based on classifier decision (that is not certified) and the statistical parameters of each class will be stored to be compared with the one in trained phased. Using the statistical distance measures that will be explained in the next section, the distance between features for each class in the trained phase and application phase will be compared. If the calculated distance and expected confidence (defined by an expert) difference was very low, the classifier results and its accuracy can be trusted (the system is acting autonomously), if the difference was low, the system can ask for more data and re-evaluation to make sure about the distance. In case of larger difference, the classifier results and accuracy are no longer valid, and the system should use an alternative approach or notify a human agent (In this example, the system will ask the responsible security agent to check the network traffic manually and make a decision).



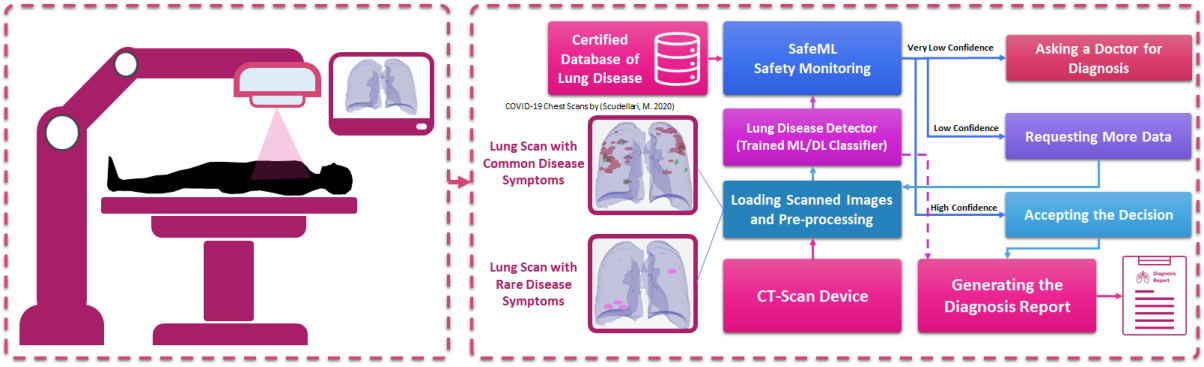
Application of SafeML in security intrusion detection (e.g. datasets like CICIDS2017)

There are countless applications of Machine Learning and Deep Learning for early detection or diagnosis in various kinds of disease. For example, (Scudellari, S. (2020)) wrote about "Hospitals Deploy AI Tools to Detect COVID-19 on Chest Scans". Can this AI tools be fully autonomous in detection or diagnosis? Are they Safe? What is our definition of their safety? It is believed that SafeML or similar approaches can be a possible answer for those questions. The following figure illustrates the application of SafeML in medical diagnosis (e.g. COVID-19 diagnosis using lung scans).



Application of SafeML in medical diagnosis (e.g. COVID-19 Diagnosis using [Lung Scans])

Another example can be a traffic sign detector in an autonomous car or self-driving vehicle that uses Machine Learning or Deep Learning to detect the traffic sign and generate the required action(s). The following block diagram shows how SafeML can be used in this case study. This can be used also for autonomous platoon systems (Kabir, S., et al. (2020)).

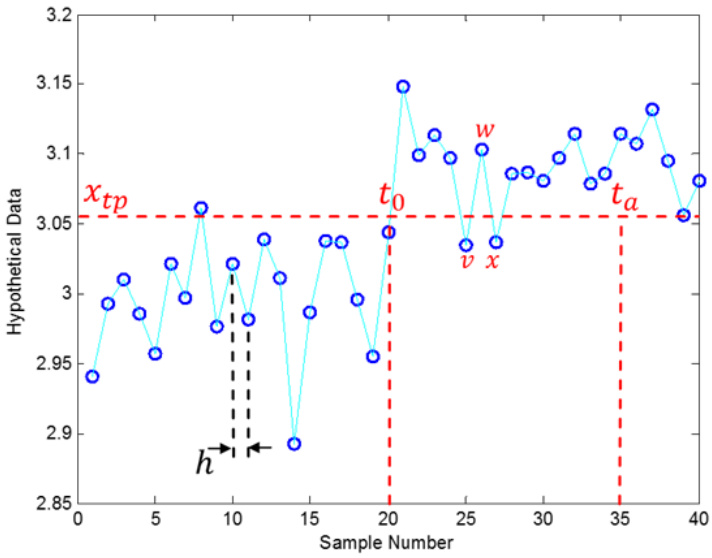


Application of SafeML in self-driving cars (e.g. Traffic Sign Detection) [Car vector created by freepik — www.freepik.com]

Each of the above-mentioned applications will be implemented using SafeML with some well-known case studies in our next stories (Part II, Part III and Part IV).

3 Statistical Distances and their Potential Applications in Accuracy Estimation

A threshold line can be considered as the simplest version of a classifier. Consider the following figure; in this simple classifier any point bellow the threshold line (Xtp) will be considered as class 1 and any point above the threshold line will be counted as class 2. Assume that we know that the points between time 0 to 20 are class 1 and other points are class 2. As can be seen in this simple example, x and v points are misclassified.

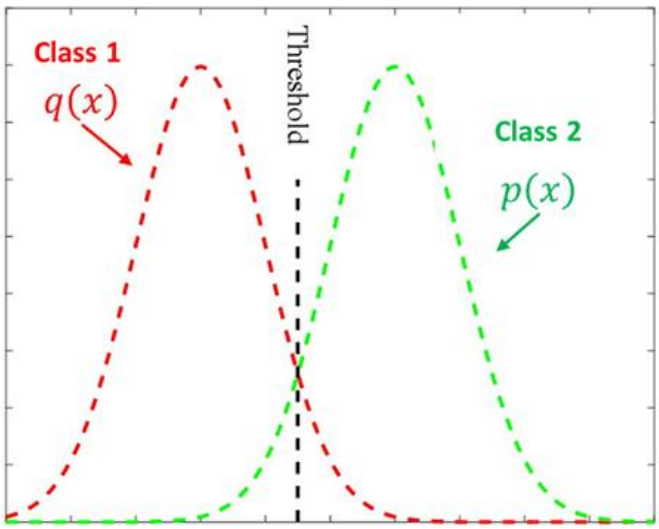


A hypothetical signal and a simple threshold of Xtp (Aslansefat et al., 2020-a)

If we estimate the probability density function of each class (the following figure), then the probability of error can be calculated as:

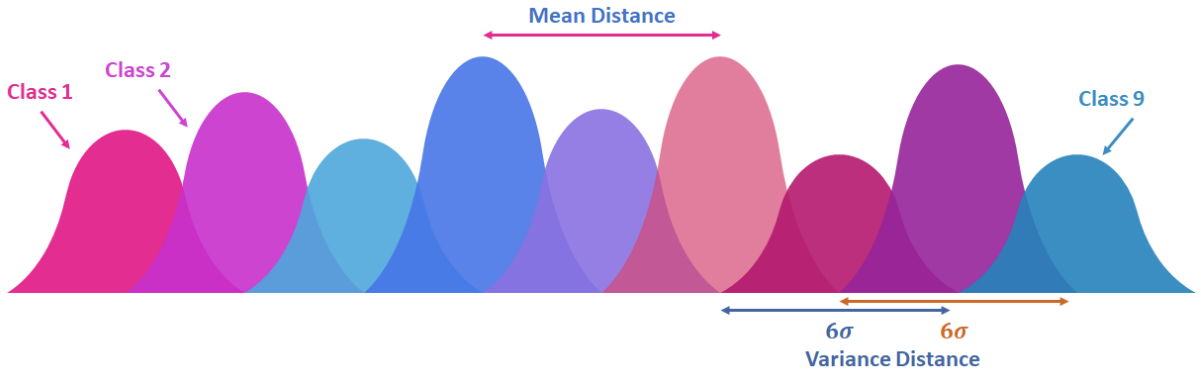
$$P(Error) = \int_{xtp}^{+\infty} q(x)\,dx + \int_{-\infty}^{xtp} \mathrm{p(x)\,dx}$$

The classifier accuracy can be easily obtained using 1-P(error). In this simple classifier, the area that two probability density functions merge causes the error.



Probability density functions of class 1 and class 2

Fukunaga, K. (1990) showed that the upper bound error can be calculated using probability density function (PDF)-based distances like Bhattacharyya distance. The PDF-based distance measures usually rely on mean and variance distance as shown in the following figure. However, some exiting advanced methods can also compare the shape of different PDFs.



Distance measures based on probability density functions (PDFs)

The following figure shows four well-known PDF-based distance measure methods.

Some of the well-known PDF-based distance measures

3.1 Chernoff and Bhattacharyya based Upper Bound Error

The following python example of upper bound error probability estimation based on Chernoff method is provided. In this code, if one considers "s = 0.5", then it will be the Bhattacharyya upper bound error estimation.

https://gist.github.com/koo-ec/67e15b202864d90b8be4f768557c5958

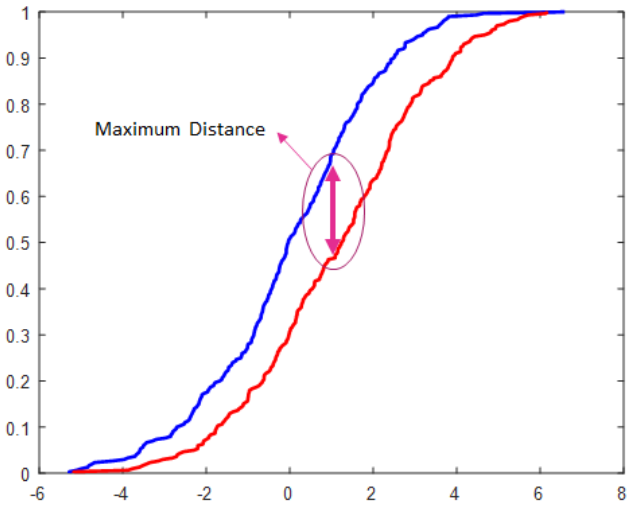
It can be proved that the probability of error is correlated with the distance between the cumulative distribution functions (CDFs) (Aslansefat, K. et al. 2020-b). Some of the well-known CDF-based distance measures can be listed as follows:

Some of the well-known CDF-based distance measures

Sometimes it can be easier to use the empirical cumulative distribution function (ECDF) of features. Python examples of ECDF-based distance measures are provided as follows.

3.2 Kolmogorov-Smirnov Distance

Consider we have a dataset with two classes and one feature. The following figure shows the ECDF of the feature for class 1 (blue) and class 2 (red). The Kolmogorov-Smirnov simply finds the maximum exiting distance between two ECDFs.



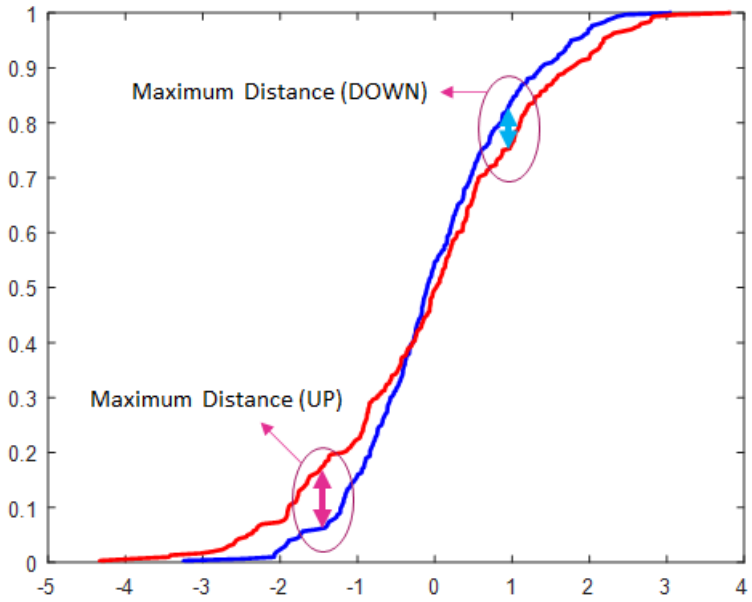
Kolmogorov-Smirnov distance measure Illustration

As an example, you can check the following python code for Kolmogorov-Smirnov distance measure:

https://gist.github.com/koo-ec/4385b7ee08c584ba8c810b39e70d9d5b

3.3 Kuiper Distance

The Kuiper distance has similar functionality to the Kolmogorov-Smirnov distance. However, this method considers two maximum distance as shown below; a) when blue ECDF has a larger value than red ECDF and b) when red ECDF has a larger value than blue ECDF. The Kuiper distance can be obtained by adding two maximum values.



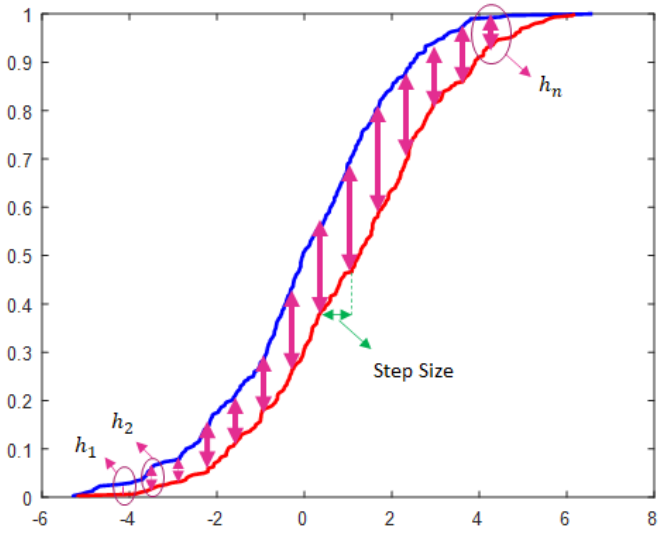
Kuiper distance measure illustration

An example of Kuiper distance measure in python is provided as follows.

https://gist.github.com/koo-ec/9e31d2e31ad32950dda23b769b53fd7f

3.4 Cramer-Von Mises Distance

ECDF is formed by many small steps. Consider the absolute difference of two steps in the same interval as height. If we calculate the summation of all calculated height values for all steps, then we have the Cramer-Von Mises Distance.



Cramer-Von Mises distance measure illustration

You can check the sample python code for Cramer-Von Mises Distance as follows:

https://gist.github.com/koo-ec/40a1773bead7ad56ffcd9b675f85d3be

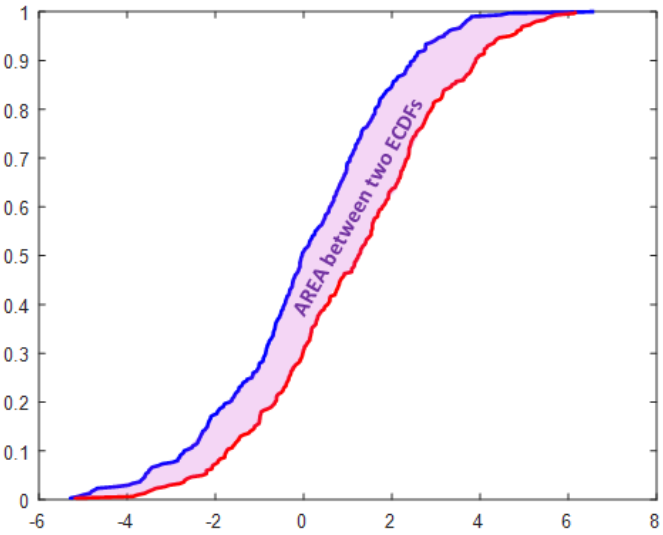
3.5 Anderson-Darling Distance

The Anderson-Darling distance is similar to the Cramer-Von Mises Distance. The only difference is that this approach normalizes height values by their standard deviation (SD). Please check the following python example for Anderson-Darling distance.

https://gist.github.com/koo-ec/f47f81e8fb5b613ad83edda1ece4ac01

3.6 Wasserstein Distance

Wasserstein distance has been used in many applications. For example, it has been used as a loss function in generative adversarial neural networks (GANs) (Gulrajani, I. 2017). The Wasserstein distance we consider both height values and width values for all steps. This method somehow measures the area between two ECDFs when we consider power equal to one. When the power factor is one, Wasserstein distance is equal to Earth Mover Distance.



Wasserstein distance measure illustration

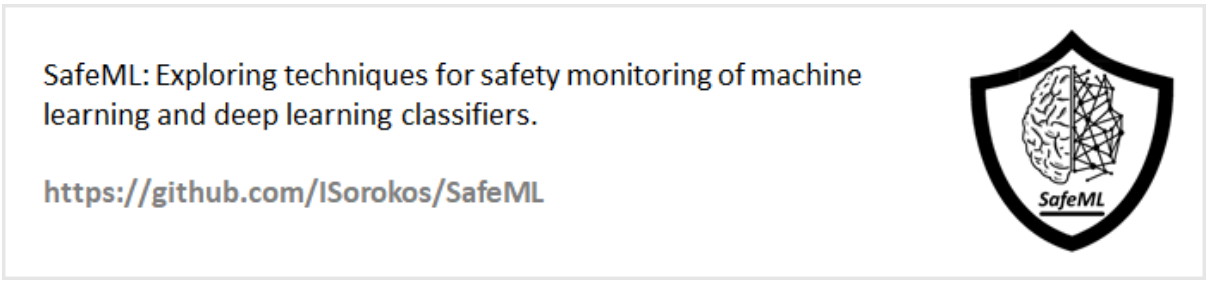
You can check the following python example of Wasserstein distance.

https://gist.github.com/koo-ec/0edce796bf7315ec6da444294ce01776

The above python codes are also available on Google Colab. If one uses R for Programming the twosamples library is suggested. The above Python codes have been rewritten from this library. For MATLAB users, a set of ECDF-based distance measures functions are recommended. To see some examples and case studies with SafeML idea please check the following GitHub project: https://github.com/ISorokos/SafeML

More details about SafeML is available in our recent paper:

[arXiv][ResearchGate][DeepAI][PaperWithCode].



4 Conclusion

In this story, the topic of AI safety has briefly introduced and the idea of SafeML has explained with some possible applications. Some of the well-known ECDF-based distance measures algorithms have been provided with their simple python example. In our next stories, the above-mentioned applications of SafeML will be provided with code implementation. Each of the above-mentioned ECDF-based approaches works well for a certain class of system. Thus, the relation between the system's characteristics and ECDF-based distance measures will be discussed in the next stories. The SafeML is still in an early stage of development and it is aimed to extend it for dealing with time-series data, prediction and regression algorithms (i.e. Schulam, P., et al. (2019)) and domain adaptation (i.e. Shen, J., et al. (2018)). It is also possible to use SafeML as Explainable AI that will be discussed later.

It should be mentioned that in parallel with "AI Safety" research, there are some other research works focusing on the application of AI for improving safety models (Gheraibia, Y., et al. (2019)).

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Aslansefat, K., Gogani, M. B., Kabir, S., Shoorehdeli, M. A., & Yari, M. (2020-a). Performance evaluation and design for variable threshold alarm systems through semi-Markov process. *ISA transactions*, *97*, 282–295. https://doi.org/10.1016/j.isatra.2019.08.015

Aslansefat, K., Sorokos, I., Whiting, D., Kolagari, R. T., & Papadopoulos, Y. (2020-b). SafeML: Safety Monitoring of Machine Learning Classifiers through Statistical Difference Measure. *arXiv preprint arXiv:2005.13166*.

Balunovic, M., Baader, M., Singh, G., Gehr, T., & Vechev, M. (2019). Certifying Geometric Robustness of Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 15287–15297) [Link].

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., & Vechev, M. (2018, May). Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *IEEE Symposium on Security and Privacy (SP)* (pp. 3–18). https://doi.org/10.1109/SP.2018.00058

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of Wasserstein gans. In *Advances in neural information processing systems* (pp. 5767–5777) [Link].

Gheraibia, Y., Kabir, S., Aslansefat, K., Sorokos, I., & Papadopoulos, Y. (2019). Safety+ AI: A Novel Approach to Update Safety Models Using Artificial Intelligence. *IEEE Access*, *7*, 135855-135869. https://doi.org/10.1109/ACCESS.2019.2941566

Kabir, S., Sorokos, I., Aslansefat, K., Papadopoulos, Y., Gheraibia, Y., Reich, J., … & Wei, R. (2019, October). A Runtime Safety Analysis Concept for Open Adaptive Systems. In *International Symposium on Model-Based Safety and Assessment* (pp. 332–346). Springer, Cham. https://doi.org/10.1007/978-3-030-32872-6_22

Scudellari, S. (2020) Hospitals Deploy AI Tools to Detect COVID-19 on Chest Scans, IEEE Spectrum.

Schulam, P., & Saria, S. (2019). Can you trust this prediction? Auditing pointwise reliability after learning. *arXiv preprint arXiv:1901.00403*.

Shen, J., Qu, Y., Zhang, W., & Yu, Y. (2018, April). Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence [Link].*

Related GitHub Projects

SafeML Project: The idea that has been briefly explained in this story.

NN-Dependability-KIT Project: Toolbox for software dependability engineering of artificial neural networks.

Confident-NN Project: Toolbox for empirical confidence estimation in neural networks-based classification.

SafeAI Project: Different toolboxes like DiffAI, DL2 and ERAN from SRILab ETH Zürich focusing on robust, safe and interpretable AI.

Acknowledgement

I would like to thank contributors of SafeML project:

- Professor Yiannis Papadopoulos (University of Hull)
- Dr. Ioannis Sorokos (Fraunhofer Institute for Experimental Software Engineering)
- Dr. Ramin Tavakoli Kolagari (Nuremberg Tech)
- Declan Whiting (University of Hull & APD Communications)

To be continued …