Keep Your Distance: Determining Sampling and Distance Thresholds in Machine Learning Monitoring

Al-Harith Farhad[1], Ioannis Sorokos[2(✉)], Andreas Schmidt[2], Mohammed Naveed Akram[2(✉)], Koorosh Aslansefat[3], and Daniel Schneider[2]

[1] University of Mannheim, Schloss, 68131 Mannheim, Germany
afarhad@mail.uni-mannheim.de
[2] Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
{ioannis.sorokos,andreas.schmidt,naveed.akram, daniel.schneider}@iese.fraunhofer.de
[3] University of Hull, Cottingham Road, Hull HU6 7RX, UK
k.aslansefat@hull.ac.uk

Abstract. Machine Learning (ML) has provided promising results in recent years across different applications and domains. However, in many cases, qualities such as reliability or even safety need to be ensured. To this end, one important aspect is to determine whether or not ML components are deployed in situations that are appropriate for their application scope. For components whose environments are open and variable, for instance those found in autonomous vehicles, it is therefore important to monitor their operational situation in order to determine its distance from the ML components' trained scope. If that distance is deemed too great, the application may choose to consider the ML component outcome unreliable and switch to alternatives, e.g. using human operator input instead. SafeML is a model-agnostic approach for performing such monitoring, using distance measures based on statistical testing of the training and operational datasets. Limitations in setting SafeML up properly include the lack of a systematic approach for determining, for a given application, how many operational samples are needed to yield reliable distance information as well as to determine an appropriate distance threshold. In this work, we address these limitations by providing a practical approach and demonstrate its use in a well known traffic sign recognition problem, and on an example using the CARLA open-source automotive simulator.

Keywords: Machine Learning · Monitoring · Safety · Uncertainty

1 Introduction

The continuous expansion of the application fields of *Machine Learning* (ML) into safety-critical domains, such as autonomous vehicles, entails an increasing

C. Seguin et al. (Eds.): IMBSA 2022, LNCS 13525, pp. 219–234, 2022.
https://doi.org/10.1007/978-3-031-15842-1_16

need for suitable safety assurance approaches. One key aspect in this regard is getting a grasp on the confidence associated with the output of an ML component. While some ML models provide a probabilistic output that can be interpreted as a level of confidence, such output alone is not sufficient to establish overall trust. Significant progress has been made towards addressing this question, with approaches that introduce more sophisticated evaluation of a given model's outputs. Model-specific approaches base their evaluation on understanding of the internals of the given ML model, e.g. [23] focus on the second-to-last layer of a given deep neural network. On the other hand, model-agnostic approaches treat models as black-boxes, basing their evaluation on properties that can be examined externally, e.g. in [16], surrogate models are constructed during training to later provide uncertainty estimates of the ML model in question. An additional concern for evaluating ML models, is that the evaluation must also satisfy the application requirements, in particular with regards to performance. For instance, the authors of [25] propose auxiliary networks for evaluation, but the computational capacity needed to estimate them hinders their roll-out into real-time systems. On a general note, A safety argument for a system with ML components will typically be very specific for a given application and its context and comprise of a diverse range of measures and assumptions, many of which we would expect to include both development-time approaches and runtime approaches, with ours falling under the latter category.

SafeML, proposed in [2] and improved in [1], is a runtime approach for evaluating ML model outputs. In brief, SafeML compares training and operational data of the ML model in question and determines whether they are statistically 'too distant' to yield a trustworthy answer. The work in [1] further demonstrates a bootstrap-based p-value estimation extension to improve confidence in measurements. However, the existing literature does not explain how to address specific challenges for practical application of SafeML.

Our contribution is to identify these limitations and propose an approach that enables a systematic application of SafeML and overcomes these limitations. In the remainder of Sect. 1, we provide a more detailed description of previous work on SafeML. We then discuss what its practical limitations are, provide the motivation behind our approach, and then further detail our contributions.

1.1 SafeML

SafeML is a collection of measures that estimate the statistical distance between training and operational datasets based on the *Empirical Cumulative Distribution Function* (ECDF). In [2], the estimated distance has been shown to negatively correlate with a corresponding ML model's accuracy. In the same paper, a plausible workflow of applying SafeML for monitoring ML was also proposed. The workflow allows an ML task to be divided into two phases, an offline/training phase and an online/application phase. In the training phase, it is assumed that we have a trusted dataset and there is no uncertainty associated with its labels. An ML model, such as a deep neural network or a support vector machine, can be trained using the trusted data for classification or regression tasks.

After its validation, in the online/application phase, the same trained model and a buffer are provided to gather a sufficient number of samples from inputs. The number of buffered samples should be large enough that the distance determination can be relied upon, but the existing approach does not provide further guidance on how this number should be specified. When a large enough number of samples is obtained, the ECDF of each feature and each class is calculated based on the trained classifier decisions. The ECDF-based statistical distance measures are used to evaluate the differences between the trusted dataset and the buffered data. To ensure that the statistical measures are valid, a bootstrap-based p-value evaluation method is added to the measurements, as in [1]. The user of the method must then specify a minimal distance threshold (and optionally additional ones) for the distance measures. The proposed workflow suggests that if the outcome is slightly above the minimal threshold, additional data can be requested. On the other hand, if the outcome is significantly above the threshold value (or a specified additional threshold), alternative actions can be taken, e.g. operator intervention. If the outcome is below the minimal threshold (or a specified additional threshold), the decision of the Machine Learning algorithm can be trusted and the statistical distance measures can be stored to be reported.

As SafeML is model-agnostic, it can be flexibly deployed in numerous applications. In [1,2], Aslansefat et al. already presented experimental applications of SafeML for security attack detection [27], and *German Traffic Sign Recognition Benchmark* (GTSRB) examples [29]. For security intrusion detection, SafeML measures were used to compare the statistical distances against the accuracy of classifier. In the GTSRB example, the model was trained, and the incorrectly classified set of images was compared against randomly selected input images from the training set.

1.2 Motivation

As mentioned in Sect. 1.1, applying SafeML requires the specification of the number of runtime samples that needed to be acquired, and at least the minimal distance threshold for acceptance/rejection. Both parameters must be defined during development time, as they need to be known by the time the ML model is in operation. Existing work on SafeML does not investigate nor provide guidance for establishing these parameters, leaving it up to the user to find reasonable values.

However, this is not a trivial matter, as identifying appropriate thresholds has application-related implications. As will be highlighted further in Sect. 3, an inadequate number of runtime samples may result in low statistical power of the SafeML-based evaluation, whereas collecting too many samples can be inefficient and limit application performance. Addressing these limitations is the focus of this publication.

Statistical power is the probability of a correctly rejected null-hypothesis test, i.e., the probability of a true positive, given a large enough population [7]. Conversely, by presetting a required level of statistical power, the population size needed to correctly distinguish two distributions can be calculated through power analysis. Similarly, distance thresholds that are too low can lead to flooding the host application with false positive alarms, whereas distance thresholds that are too high can lead to potentially critical conditions being overlooked. Concretely, we establish the following research questions:

RQ1: Dissimilarity-Accuracy Correlation. Can we confirm that data points seen during operation that are dissimilar to training data impact the model's performance in terms of accuracy?

RQ2: Sample Size Dependency. Can we determine whether the sample size affects the accuracy of the SafeML distance estimation?

1.3 Paper Contribution and Outline

The contribution of this paper is three-fold. First, we use power analysis to specify sampling requirements for SafeML monitoring at runtime. Second, we systematically determine appropriate SafeML distance thresholds. Finally, we apply the above method in the context of an example automotive simulation.

The remainder of the paper is structured as follows: In Sect. 2, we discuss background and related work, including approaches both similar to and different from SafeML. In Sect. 3, we describe our approach for systematically applying SafeML and determining relevant thresholds, as well as our experimental setup. In Sect. 4, we discuss our experimental results, before recapping our key points and discussing future work in Sect. 5.

2 Background and Related Work

To briefly recap, in [1,2] the authors propose statistical distance measures to compare the distributions of the training and operational datasets; the measures are based on established two-sample statistical testing methods, including the Kolmogorov-Smirnov, Anderson-Darling, Cramer von Mises [8], and Wasserstein methods [24]. The statistical distance measures used by SafeML capture the dissimilarity between two different distributions, but the approach itself does not propose an explicit threshold at which those distributions are not equivalent, nor a means for determining one systematically.

Setting meaningful thresholds is a reoccurring problem in ML and data-driven applications. A method based on the 3-sigma rule was shown to provide suitable threshold criteria in Hidden Markov Models under the assumption of normal distribution [6]. Our approach is similar in the sense that we used the same principle, but we did not assume that our datasets are normally distributed. Therefore, instead of a 3-sigma rule, we opted for a gradual increase of the threshold based on the sigma value. We will elaborate on this further in Sect. 3.

A prerequisite for the transition of AI applications to safety- and security-critical systems is the existence of guarantees and guidelines to assure underlying system dependability. A method was proposed in [25] to assure a model's operation within the intended context in a model-agnostic manner, with an additional autoencoder-based network being used to detect semantic novelty.

However, the innate problem of using neural networks, including autoencoders, is their black-box nature with respect to explainability, which inhibits the establishment of dependability guarantees. Hence, the use of a more explainable statistical method could serve as a solution to this issue. This includes our proposed approach, as the ECDF-based distance to the training set could provide additional insight into the model's decision.

In [23], the authors propose a commonality metric that, inspects the second-to-last layer of a *Deep Neural Network* (DNN). The proposed metric expresses the ratio between the activation of the neurons in the last layer during training (across all training instances) versus their activation during operation, for the given operational input. The approach shares common ideas with SafeML, but diverges in terms of being model-specific, as the metric directly samples the last layer's neurons. In contrast, SafeML does not consider model internals and makes no assumption on the distribution of the training and operational data.

Efforts have been made to ensure a dependable and consistent behavior in AI-based applications. These have taken various forms, from providing generative models, whose outputs can be interpreted as confidence in the predictions, to the aforementioned novelty detection. Design-time safety measures are introduced in [28], where the robustness of neural networks could be certified through a novel abstract domain, before deployment. Similarly, a feature-guided safety testing method for neural networks is proposed in [30] to evaluate the robustness of neural networks by feeding them through adversarial examples. Markov decision processes have also been proposed to be paired with neural networks to verify their robustness through statistical model checking [12].

Uncertainty wrappers are another notable concept [13–16]. This mathematical concept distinguishes ML uncertainty into three layers I) model performance, II) input quality, and III) scope compliance, and provides a set of useful functions for evaluating the existing uncertainties in each step. The uncertainty wrapper can be compared to SafeML in the third layer (scope compliance). Both of them are model-agnostic.

Safeguard AI [17] proposes calculating the likelihood of *out-of-distribution* (OOD) inputs and adding it to the loss function of the ML/DL model. This approach also uses a *Generative Adversarial Network* (GAN) to produce boundary data in order to create a more accurate OOD. In comparison to SafeML, the approach is model-specific and cannot be evaluated at runtime.

Another common theme across approaches for safeguarding ML models is the investigation of all conceivable input perturbations to produce robust, safe, and abstract interpretable solutions and certifications for ML/DL models [9,10,18–20,26]. These approaches are also model-specific and do not provide runtime solutions. Similar to previous approaches, *DeepImportance* is a model-specific solution that presents a new *Importance-Driven Criteria* (IDC) as a layer-wise function to be assessed during the test procedure and provides a systematic framework for ML testing [11]. Regarding the reliability evaluation of ML models, only a small number of solutions have been provided so far. One of these is ReAsDL, which divides the input space into tiny cells and evaluates the

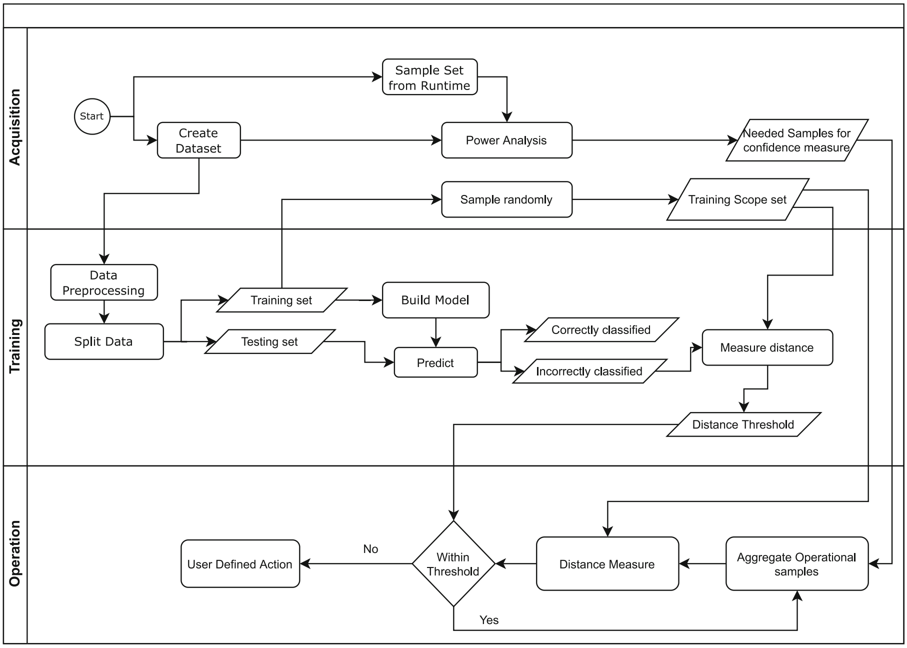

Fig. 1. Process flowchart

ML/DL reliability based on the cells' robustness and operational profile probability [31,32]. This solution is model-agnostic and focuses on classification tasks similar to SafeML. The NN-Dependability-kit suggests a new set of dependability measures to assess the impact of uncertainty reduction in the ML/DL life cycle. The authors also included a formal reasoning engine to ensure that the ML/DL dependability is guaranteed. The approach can be used for runtime purposes [3].

3 Methodology

In this section, we present our refined approach for applying SafeML, in the form of a proposed workflow, and address the question of how to determine the sampling and distance thresholds. To validate our approach, we applied SafeML to ML monitoring during simulation and, also used it against an existing dataset, the GTSRB. In the next section, we will describe the experimental design for our empirical evaluation of the proposed approach.

3.1 Process Workflow

The process workflow for determining the needed number of samples as well as the distance threshold is divided into three stages, as shown in Fig. 1.

- **Acquisition:** In this stage, two datasets are involved, a training dataset and a testing dataset. In our empirical experiments (see Sect. 3.2), these datasets are generated from the simulation, but they should generally be derived during development. At this point, power analysis is used to find the number of samples to determine the difference between the operational dataset and the training set. This factor can be calibrated for the application at hand, as it determines an additional number of samples beyond the minimum needed to achieve the determined test power. The effect size for the power analysis is established between the training set and the testing set, using Cohen's d coefficient [4].
- **Training:** The training dataset is processed and split into a training set and a testing set. A sub-sample of the smaller training set is uniformly sampled to represent the *Training Scope Set* (TSS) in the calculation of statistical distances, which maintain its features in order to reduce computational complexity during runtime. A model is then built from the smaller training set and used to predict the outputs of the testing set. The result is further distinguished into correctly and incorrectly classified outputs, where SafeML measures evaluate the statistical distance between the incorrectly classified outputs and the TSS. The resulting distances are finally used as the initial distance threshold. This initial distance threshold is then increased gradually by a factor of the standard deviation until a user-defined safety performance level is met.
- **Operation:** Once the trained model is in operation, the value obtained in the 'Acquisition' stage is used to aggregate operational data points into an operational set. SafeML measures evaluate the statistical distance between this operational set and the TSS. If the value falls within the defined threshold, the model continues its operation normally, otherwise, a signal is sent to run a user-defined action.

3.2 Experiment Setup

We performed experiments on the German Traffic Sign Recognition Benchmark (GTSRB) [29] and on a synthetic example dataset in the CARLA simulator[1] [5] to evaluate our approach. CARLA is an automotive simulator used for the development, training, and validation of autonomous driving systems. The dataset generated from CARLA was used to evaluate the confidence level of SafeML predictions and the autopilot decisions of the simulated vehicle. The GTSRB dataset is a collection of traffic sign images, along with their labels used for benchmarking the ML algorithms. It was first used in 2011. The dataset is a good representation of the safety-critical application of ML-components. Hence, it was also considered in this work for the evaluation of the presented approach.

The CARLA setup allows us to identify a systematic method for estimating the minimum number of required samples and the distance acceptance threshold though a fixed-point iteration, as well as to determine their implication on the

[1] https://carla.org.

model's prediction and how they correlate to the model's performance. It also offers multiple maps called Towns, with different sizes and properties, which allows for the experiment to be repeated. A simple model was built from a dataset sampled from CARLA, using a vehicle autopilot with varying driver profiles (shown in Table 1). This corresponds to the 'Acquisition' step in section Sect. 3.1. Three types of driving profiles were considered: safe, moderate, and dangerous. We should note that the profiles (and the model) were not designed with the aim to provide an accurate risk behavior estimation, but rather as a source of plausible ground truth for evaluating SafeML. A collection of classifiers were trained as the subject ML models for the CARLA dataset with results shown in Table 2. The models' inputs are the three location coordinates and the outputs are ordinally-encoded speed levels at the given coordinates (0: slow, 1: moderate, 2: fast).

As the dataset for GTSRB is already available, the creation of the dataset was assumed to be complete from the 'Acquisition' phase. Then a network was built to classify the GTSRB dataset. We built a simple convolutional neural network, as such networks are known for their superior performance on image applications. We then applied the above mentioned approach. This allows obtaining the minimum number of required samples and the distance acceptance threshold for this application.

Table 1. Properties of driver profiles

Property/driving profile	Safe	Moderate	Dangerous
Max speed	30% below limit	At limit	30% above limit
Traffic signs	Abide by all	Ignore 50%	Ignore 100%
Automatic lane-change	No	Yes	Yes
Distance to other cars	5 m	3 m	0 m

Table 2. Performance of trained models on the simulated CARLA dataset

Model	Class	Recall	Precision	F1-Score
kNN	**0**	0.89	0.95	0.92
	1	0.96	0.90	0.93
	2	0.96	0.95	0.96
Random Forest	**0**	0.83	0.52	0.64
	1	0.81	0.88	0.84
	2	0.72	0.92	0.81
LSTM	**0**	0.92	0.99	0.96
	1	0.99	0.91	0.95
	2	1.00	1.00	1.00

We trained a CNN network. The network was able to achieve an accuracy of around 99.73%. We remind readers that SafeML is model-agnostic, and other ML models could also have been used. This high accuracy resulted in very few incorrect samples for testing SafeML. Thus, one of the minority classes was excluded in order to be considered as an out-of-scope class, reducing accuracy to 97.5%. This added greater disparity to enable validation of SafeML.

In [2], SafeML distance measures have been shown to negatively correlate with the accuracy of the model. From this fact, and according to the first research question established in Sect. 1.2, we hypothesize that misclassified points would have a higher distance than correctly classified data points due to their dissimilarity to the training set.

Furthermore, from principles of statistical analysis, it is established that, if an insufficient number of samples is used during hypothesis testing, there is a risk of the statistical tests not achieving sufficient power. According to our second research question in Sect. 1.2, our corresponding hypothesis is that the number of samples correlates with confidence of dissimilarity (the magnitude of the distance).

The experiment concluded by following the 'Operation' step of the process workflow explained in Sect. 3.1. In the CARLA example, the same experiment was reproduced in different environment setups to ensure consistency of the results. In GTSRB, this was performed on the test set, which can be replaced by runtime dataset, at runtime.

Table 3. Mean and standard deviation of the statistical distances of the entire test set (CVM: Cramer von Mises, AD: Anderson-Darling, KS: Kolmogorov-Smirnov, WS: Wasserstein)

	Prediction	CVM	AD	KS	WS
kNN	**Correct**	1569.71, 617.60	8.577, 3.03	0.0193, 0.0043	3.192e−05, 1.153e−05
	Incorrect	5743.45, 2085.75	35.35, 11.12	0.083, 0.0139	1.430e−04, 5.264e−05
Random Forest	**Correct**	3780.74, 227.29	18.59, 0.97	0.0341, 0.0007	1.238e−04, 1.875e−05
	Incorrect	10478.63, 1147.64	56.73, 4.78	0.1068, 0.0161	4.368-04, 6.654e−05
LSTM	**Correct**	2744.89, 895.56	13.63, 3.26	0.0578, 0.0034	4.356e−05, 2.276-05
	Incorrect	7892.06, 1033.94	43.24, 3.23	0.1772, 0.0871	2.134e−04, 1.033e−04

4 Results

4.1 Preliminary Findings

Before continuing with the workflow of the simulation, an analysis of the trained model was used to test the hypotheses predefined in Sect. 3.2, namely:

RQ1: Dissimilarity-Accuracy Correlation was tested by calculating the statistical distance between the correctly classified data points and the TSS, as well the incorrectly classified data points and the training scope. Table 3 shows the mean and standard deviation of each of the statistical distance measures used. It shows that the incorrectly classified points are highly dissimilar to the TSS (higher distance), supporting the corresponding hypothesis.

RQ2: Sample Size Dependence: Due to the model's accuracy of 95%, the number of correctly classified data points was significantly larger than that of incorrectly classified points when the distances in Table 3 were calculated. To account for the number of samples, the distances were calculated over a varying number of randomly sampled points of each group. As shown in Fig. 2, the distance of incorrectly classified points is always larger than the distance of correctly classified points and increases with increasing number of samples. This can be attributed to several factors, such as: (a) increased distinction between the distributions and (b) a shift of the average value of the distances when the number of available samples increases, which removes skewness in the distribution.

4.2 Experiment Results

Following the process workflow presented in Sect. 3.1, each stage produced its corresponding values after being executed on the "Town 1" standard map from CARLA. In the 'Acquisition' stage, power analysis was used on each of the driver profiles. The highest number of samples returned was 91. Multiplying this by an additional factor of 1.3 yielded a final number of samples of 120, which aligned with our sampling batches; the operational samples were collected in batches over 4 s with a simulation resolution of 30 frames per second. The performance of the trained model is shown in Table 2, where the kNN model was used in the evaluation of the results due to its simplicity and high reported performance. The resulting threshold values for SafeML are shown in Table 4.

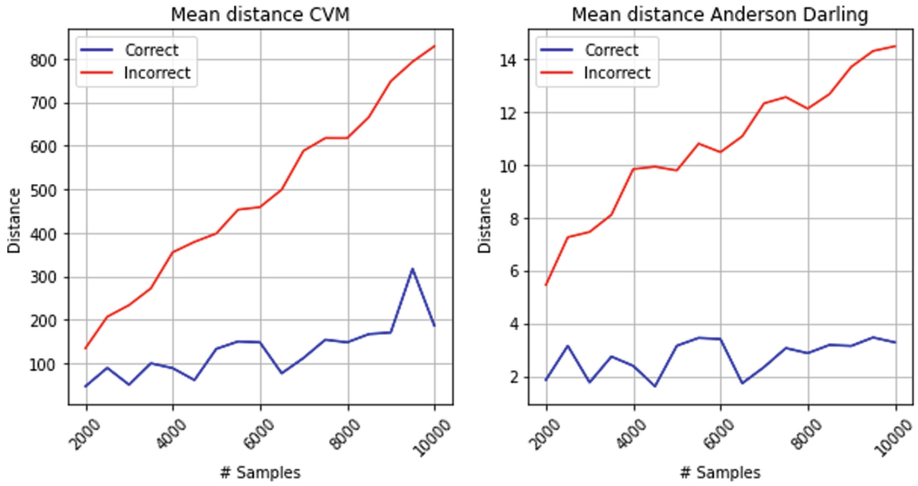

Fig. 2. Statistical distance over varying sampling sizes

Table 4. Threshold parameters used for Town 1 (CVM: Cramer von Mises, AD: Anderson-Darling, KS: Kolmogorov-Smirnov, WS: Wasserstein)

Prediction	CVM	AD	KS	WS
Mean	387.83	9.64	0.087	1.38e−4
Standard deviation	171.57	3.61	0.02	6.22e−5

The acceptable performance of the ML-model is a design decision obtained from the application requirements specified. In our example, let us consider the correctness over a batch. Since each batch contains multiple frames, let us assume a batch is considered correctly classified if its overall accuracy is 0.8 (96 correct points out of 120). Consequently, a batch is assumed to be incorrectly classified if its overall accuracy is 0 (focusing on worst-case scenarios), with all of its members being misclassified. This high limit was chosen to represent an extreme scenario that minimizes the number of false alarms.

The performance of each of the distance measures in SafeML was evaluated on different driver profiles as shown in Figs. 3 and 4, where the true positive rate (batches with 0 accuracy that were above the threshold) and the false positive rate (batches with 0.8 accuracy that were above the threshold) were plotted over a varying increase in the threshold in increments of 0.1 of the standard deviation.

Figure 3 shows the standard deviation factor by which the threshold should be increased to yield reliable identification by SafeML. The plot compares incorrect

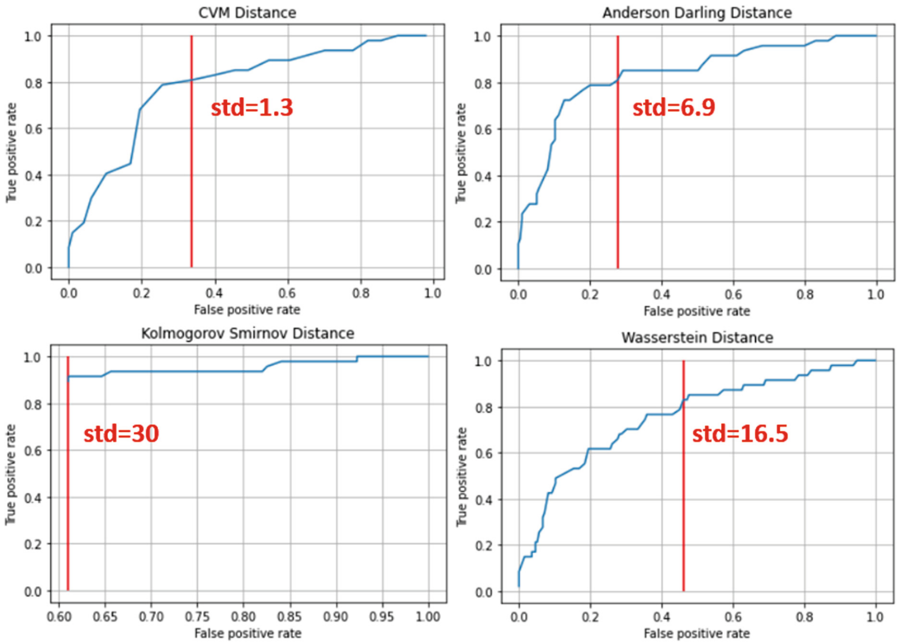

Fig. 3. SafeML performance on Town 1 with moderate driver profiles

(i.e., false positive rate) versus correct SafeML alarms (true positive rate), set to a threshold of 0.8 (as mentioned previously, this threshold can be determined based on application-level requirements). Through this method, a suitable factor for the distance measures was found, with the exception of Kolmogorov Smirnov, where a similar percentage of false positive rates was achieved for the distance measures.

The same process was repeated for the dangerous driver profile shown in Fig. 4, where similar plot curves were observed, and the threshold points could be established following similar steps as for the moderate profile. However, the performance ratio between true and false positive rate is exceptionally bad. The experiment was repeated on "Town 2" and "Town 4" with similar results.

Repeating the process workflow on the GTSRB shows quite a similar trend, where the correct classification and the incorrect classification are completely separable by setting a suitable distance threshold, as shown in Fig. 5. The number of samples (with each sample being an image) required can be seen on the x-axis. In this case, the majority of the incorrect classifications represent an out-of-scope class. The distance was calculated using features derived from the last layer of the CNN instead of from the raw pixels. More detailed results can be found in the git repo.

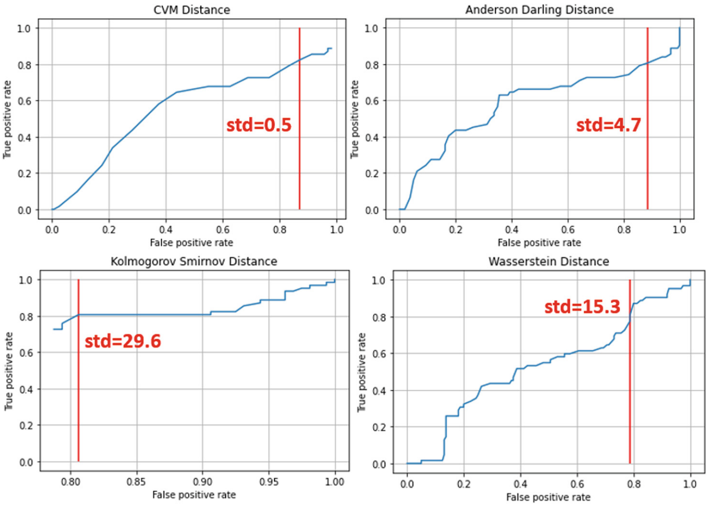

Fig. 4. SafeML performance on Town 1 with dangerous driver profiles

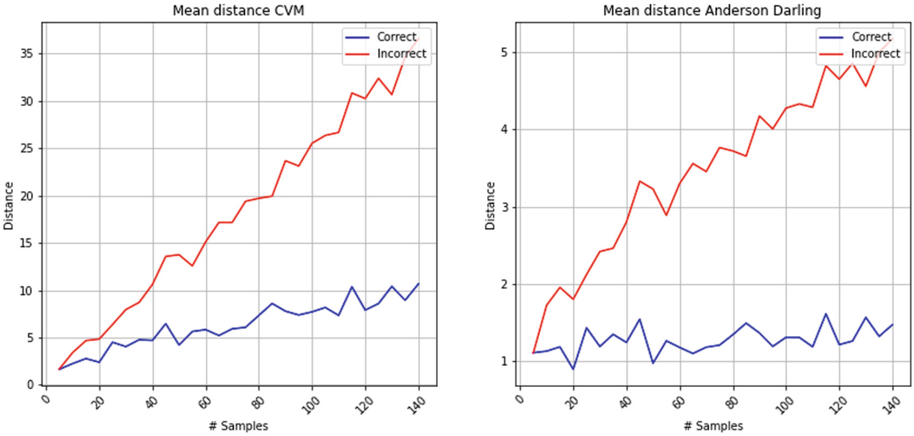

Fig. 5. Statistical distance over varying sampling sizes for GTSRB

5 Conclusion and Future Work

In this paper, we addressed the challenge of determining sampling and distance thresholds for SafeML, a model-agnostic, assessment tool for scope compliance. Our approach incorporates power sampling during the development stage of the subject ML model in order to determine the number of samples necessary to achieve sufficient statistical power while applying the SafeML distance evaluation during the runtime stage. Furthermore, we proposed means of identifying appropriate distance thresholds, based on the observed performance of the ML model during development-time simulation. We validated our approach experimentally, using a scenario developed in the CARLA automotive simulator as well as the publicly available GTSRB dataset.

Apart from the SafeML applications discussed earlier in Sect. 2, at the time of writing, additional examples are being researched, such as using SafeML for cancer detection via x-ray imaging as well as for pedestrian detection, financial investment, and predictive maintenance.

Regarding future work, we are considering further directions to improve SafeML, including investigating the effect of outlier data' and the effect of dataset characteristics (see [22]), using dimensionality reduction, accounting for uncertainty in the dataset labels (see [21]), and expanding the scope towards graph, quantum, and time-series datasets.

Code Availability

Regarding the reproducibility of our research, codes and functions supporting this paper have been published online at: https://tinyurl.com/4a76z2xs.

Acknowledgements. This work was supported by the Secure and Safe Multi-Robot Systems (SESAME) H2020 project under grant agreement 101017258 and the German Federal Ministry for Economic Affairs and Climate Action (BMWK) within the research project "FabOS" under grant no. 01MK20010A.

References

1. Aslansefat, K., Kabir, S., Abdullatif, A., Vasudevan, V., Papadopoulos, Y.: Toward improving confidence in autonomous vehicle software: a study on traffic sign recognition systems. Computer **54**(8), 66–76 (2021)
2. Aslansefat, K., Sorokos, I., Whiting, D., Tavakoli Kolagari, R., Papadopoulos, Y.: SafeML: safety monitoring of machine learning classifiers through statistical difference measures. In: Zeller, M., Höfig, K. (eds.) IMBSA 2020. LNCS, vol. 12297, pp. 197–211. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58920-2_13
3. Cheng, C.H., Huang, C.H., Nührenberg, G.: nn-dependability-kit: engineering neural networks for safety-critical autonomous driving systems. In: International Conference on Computer-Aided Design (ICCAD), pp. 1–6. IEEE (2019)
4. Cohen, J.: A power primer. Psychol. Bull. **112**(1), 155 (1992)
5. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: an open urban driving simulator. In: 1st Annual Conference on Robot Learning (2017)
6. Duan, J., Zeng, J., Zhang, D.: A method for determination on HMM distance threshold. In: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 1, pp. 387–391 (2009). https://doi.org/10.1109/FSKD.2009.732
7. Ellis, P.D.: The Essential Guide to Effect Sizes: Statistical Power, Meta-Analysis, and the Interpretation of Research Results. Cambridge University Press, Cambridge (2010). https://doi.org/10.1017/CBO9780511761676
8. Evans, D.L., Drew, J.H., Leemis, L.M.: The distribution of the Kolmogorov–Smirnov, Cramer–von Mises, and Anderson–Darling test statistics for exponential populations with estimated parameters. In: Glen, A.G., Leemis, L.M. (eds.) Computational Probability Applications. ISORMS, vol. 247, pp. 165–190. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-43317-2_13
9. Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., Vechev, M.: Dl2: training and querying neural networks with logic. In: International Conference on Machine Learning, pp. 1931–1941. PMLR (2019)
10. Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.: Ai2: safety and robustness certification of neural networks with abstract interpretation. In: Symposium on Security and Privacy (SP). IEEE (2018)
11. Gerasimou, S., Eniser, H.F., Sen, A., Cakan, A.: Importance-driven deep learning system testing. In: 42nd International Conference on Software Engineering (ICSE). IEEE (2020)
12. Gros, T.P., Hermanns, H., Hoffmann, J., Klauck, M., Steinmetz, M.: Deep statistical model checking. In: Gotsman, A., Sokolova, A. (eds.) FORTE 2020. LNCS, vol. 12136, pp. 96–114. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50086-3_6
13. Jöckel, L., Kläs, M.: Increasing trust in data-driven model validation. In: Romanovsky, A., Troubitsyna, E., Bitsch, F. (eds.) SAFECOMP 2019. LNCS, vol. 11698, pp. 155–164. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26601-1_11

14. Jöckel, L., Kläs, M., Martínez-Fernández, S.: Safe traffic sign recognition through data augmentation for autonomous vehicles software. In: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 540–541. IEEE (2019)
15. Kläs, M., Jöckel, L.: A framework for building uncertainty wrappers for AI/ML-based data-driven components. In: Casimiro, A., Ortmeier, F., Schoitsch, E., Bitsch, F., Ferreira, P. (eds.) SAFECOMP 2020. LNCS, vol. 12235, pp. 315–327. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-55583-2_23
16. Kläs, M., Sembach, L.: Uncertainty wrappers for data-driven models. In: Romanovsky, A., Troubitsyna, E., Gashi, I., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2019. LNCS, vol. 11699, pp. 358–364. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26250-1_29
17. Lee, K., Lee, H., Lee, K., Shin, J.: Training confidence-calibrated classifiers for detecting out-of-distribution samples. In: International Conference on Learning Representations (2018)
18. Mirman, M., Gehr, T., Vechev, M.: Differentiable abstract interpretation for provably robust neural networks. In: International Conference on Machine Learning. PMLR (2018)
19. Mirman, M., Singh, G., Vechev, M.: A provable defense for deep residual networks. arXiv preprint arXiv:1903.12519 (2019)
20. Müller, M.N., Makarchuk, G., Singh, G., Püschel, M., Vechev, M.: PRIMA: precise and general neural network certification via multi-neuron convex relaxations. arXiv preprint arXiv:2103.03638 (2021)
21. Northcutt, C.G., Jiang, L., Chuang, I.L.: Confident learning: estimating uncertainty in dataset labels. J. Artif. Intell. Res. (JAIR) **70**, 1373–1411 (2021)
22. Oreski, D., Oreski, S., Klicek, B.: Effects of dataset characteristics on the performance of feature selection techniques. Appl. Soft Comput. **52**, 109–119 (2017)
23. Paterson, C., Calinescu, R., Picardi, C.: Detection and mitigation of rare subclasses in deep neural network classifiers. In: 2021 IEEE International Conference on Artificial Intelligence Testing (AITest), Los Alamitos, CA, USA, pp. 9–16. IEEE Computer Society, August 2021. https://doi.org/10.1109/AITEST52744.2021.00012. https://doi.ieeecomputersociety.org/10.1109/AITEST52744.2021.00012
24. Ramdas, A., Trillos, N.G., Cuturi, M.: On Wasserstein two-sample testing and related families of nonparametric tests. Entropy **19**(2), 47 (2017)
25. Rausch, A., Sedeh, A.M., Zhang, M.: Autoencoder-based semantic novelty detection: towards dependable AI-based systems. Appl. Sci. **11**(21) (2021). https://doi.org/10.3390/app11219881
26. Ruoss, A., Baader, M., Balunović, M., Vechev, M.: Efficient certification of spatial robustness. arXiv preprint arXiv:2009.09318 (2020)
27. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
28. Singh, G., Gehr, T., Püschel, M., Vechev, M.: An abstract domain for certifying neural networks. Proc. ACM Program. Lang. **3**, 1–30 (2019). https://doi.org/10.1145/3290354
29. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw. **32**, 323–332 (2012). https://doi.org/10.1016/j.neunet.2012.02.016. http://www.sciencedirect.com/science/article/pii/S0893608012000457

30. Wicker, M., Huang, X., Kwiatkowska, M.: Feature-guided black-box safety testing of deep neural networks. In: Beyer, D., Huisman, M. (eds.) TACAS 2018. LNCS, vol. 10805, pp. 408–426. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-89960-2_22
31. Zhao, X., et al.: Assessing the reliability of deep learning classifiers through robustness evaluation and operational profiles. arXiv:2106.01258 (2021)
32. Zhao, X., Huang, W., Schewe, S., Dong, Y., Huang, X.: Detecting operational adversarial examples for reliable deep learning. arXiv:2104.06015 (2021)