Towards Improving Confidence in Autonomous Vehicle Software: A Study on Traffic Sign Recognition Systems

Koorosh Aslansefat, Sohag Kabir, Amr Abdullatif, Vinod Vasudevan Nair, and Yiannis Papadopoulos

***Abstract*—The application of artificial intelligence (AI) and data-driven decision-making systems in autonomous vehicles is growing rapidly. As autonomous vehicles operate in dynamic environments, the risk that they can face an unknown observation is relatively high due to insufficient training data, distributional shift, or cyber-security attack. Thus, AI-based algorithms should make dependable decisions to improve their interpretation of the environment, lower the risk of autonomous driving, and avoid catastrophic accidents. This paper proposes an approach named SafeML II, which applies empirical cumulative distribution function (ECDF)-based statistical distance measures in a designed human-in-the-loop procedure to ensure the safety of machine learning-based classifiers in autonomous vehicle software. The approach is model-agnostic and it can cover various machine learning and deep learning classifiers. The German Traffic Sign Recognition Benchmark (GTSRB) is used to illustrate the capabilities of the proposed approach.**

***Index Terms*—Autonomous Systems, Safety Assurance, AI Safety, Statistical Distance Measure, SafeML, Safe Machine Learning**

I. Introduction

The rise of artificial intelligence and the advancement of technologies have paved the way for autonomous systems such as autonomous vehicles to enter our everyday life. Such systems have the potential to make an enormous societal and economic impact. For instance, as mentioned in Waymo Safety Report [1], when human drivers are involved in driving, around 1.35 million lives have been lost due to traffic crashes worldwide in 2016 and 836 billion dollars have been lost annually due to loss of lives and injuries caused by crashes. For each person, there is a 67% chance of getting involved in drunk driving crashes. In the US, 94% of crashes involve human choice or error. Therefore, dependable and reliable autonomous vehicles can help to save lives and decrease economic losses by reducing the number of traffic crashes by eliminating human involvement in driving.

Autonomous vehicles are increasingly given autonomous decision-making power such that while performing safety-critical tasks within human vicinity, they can autonomously make their own decisions and take actions with minimal human intervention. To be able to do so, an autonomous vehicle has to cooperate with other vehicles, road-side infrastructures (e.g. traffic sign), smart traffic light systems, etc. Consequently, using AI/machine-learning (ML) such systems continuously learn from their operation and dynamically reconfigure in response to changes such as unexpected failures of components/subsystems, the continuous change in the context of operation, variable workloads, and physical infrastructures. A key challenge for software-intensive, AI-enabled self-adaptive autonomous systems is to provide assurance about their safety and reliability.

For traditional non-autonomous systems, assurance is provided through design and development activities including verification, validation, testing, conformance to standards and certification. Safety assurances are often provided through safety arguments where safety goals are defined, and rationales for believing in these goals are designed to be dependent on a variety of assumptions. These assumptions may include aspects like failure semantics and failure rates of both hardware and software components, operating context, the efficiency of the human operator to respond to events, etc [2]. In operation, the physical system and its operating environments are monitored to see if any of these safety assumptions are violated, and thereby notify the users about the potential changes in the assurance and take necessary actions to achieve fail-safe behaviour. For example, in a car, when transient errors in hardware like sensors affect the functionality of the software like that for cruise control, an error detection unit (monitoring function) can detect the error and degrade the system by appropriate warnings and allowing the driver to take over. Therefore, the integrity of the monitoring knowledge plays a crucial role in providing accurate runtime assurances.

The issue of continuous assurance provision is further complicated for autonomous systems where important pieces of evidence are collected through ML/AI components. Due to the blackbox nature of these components, the confidence in the evidence provided by these components will directly affect the confidence in the overall assurance. For instance, consider the ML-based traffic sign recognition (TSR) system in an autonomous car, which is responsible for identifying different traffic signs and thus assisting in assuring safe driving. TSR for autonomous vehicles have several shortcomings and a survey of such shortcomings is available in [3]. Therefore, it is likely that in some cases, evidence/inputs received from a TSR could be misleading. If this misleading information is considered while providing safety assurance then it is highly likely that a false assurance could be provided, resulting in an autonomous

K. Aslansefat and Y. Papadopoulos are with the Department of Computer Science and Technology, University of Hull, Hull, HU6 7RX, UK (e-mail: k.Aslansefat-2018@hull.ac.uk and y.i.papadopoulos@hull.ac.uk), and S. Kabir, A. Abdullatif, and V. Nair are with the Faculty of Engineering and Informatics, University of Bradford, Bradford, BD7 1DP, UK (e-mail: s.kabir2@bradford.ac.uk, A.R.A.Abdullatif@bradford.ac.uk, V.Vasudevan@bradford.ac.uk)

vehicle driving with false assurance. In a worst-case scenario, this could lead to a catastrophic accident. Therefore, it is important to improve the confidence in the output generated by such software components in autonomous vehicles.

To address the above-mentioned issue with the autonomous vehicle software, TSR in particular, in this paper, we have proposed a novel approach called SafeML II which has the following features:

- It ensures the safety of a machine learning-based TSR system using modified state-of-the-art empirical statistical distance measures and can work with a variety of distribution functions, especially exponential families.
- The implemented bootstrap p-value calculation in the SafeML II functions improves the accuracy and validity of its results.
- It utilises a human-in-the-loop procedure that can use human intelligence and avoid catastrophic accidents.
- It is a model-agnostic approach that works with a variety of Machine learning and deep learning classifiers.

The effectiveness of the approach is illustrated via an application to the real-world German Traffic Sign Recognition Benchmark (GTSRB) dataset.

II. Safety Assurance Challenges of AI/ML in Automotive Domain

In 2011, the International Organization for Standardization (ISO) proposed the ISO 26262 standard to regulate functional safety for road vehicles. It includes requirements and recommendations for the entire lifecycle of car manufacturing, from the concept phase to operation and service. The main aim of ISO 26262 was to help the automotive industry address functional safety issues more systematically. However, it was defined without considering ML since the first version of ISO 26262 was published before the boom of AI. This eventually leads to a challenging issue today for car manufacturers and suppliers who are determined to incorporate ML for self-driving cars. Therefore, conventional safety assurance methods suggested by the ISO 26262 standard are insufficient or inapplicable for the assurance of ML [4]. In [5], Salay *et al.* presented an analysis of ISO-26262 part-6 methods with respect to safety of ML models. Their assessment of the applicability of the software safety methods on ML algorithms (as software unit design) shows about 40% of software safety methods do not apply to ML models.

The AI community have recently produced several papers on the problems of *'AI safety'*, e.g. [6]. One of the more influential papers [7] identifies 'concrete problems in AI' and according to this paper AI safety issues for autonomous vehicles can be categorized in five domains including I) Safe exploration, II) Scalable oversight, III) Avoiding "reward hacking" and "wire heading", IV) Avoiding negative side effects and V) Robustness to distributional shift. Efforts have been made to assure safety and improve the safety performance of ML components in autonomous vehicles. For instance, in [8], the safety assurance process for ML models in safety-critical applications has been described focusing on an explicit definition of safety requirements for ML components with respect to the safety requirements of the overall system. The approach has been illustrated via an application to a pedestrian detection system in autonomous cars.

In 2019, Kläs *et al.* [9] has emphasized on the distributional shift in the dataset and proposed an uncertainty wrapper based on Wilson Interval Confidence. The conceptual idea was explained for the German traffic sign recognition example without reporting any experimental or numerical results. In another research, they have improved their previous approach considering the impact of additional inputs like rain amount, wind direction, wind speed, and vehicle orientation on the confidence results specifically for traffic sign recognition [10], [11]. A year later, they have proposed a framework for generating an uncertainty wrapper for data-processing models and their dataflow in [12]. In all three research works, the drawback was the lack of a designed safety mechanism after measuring confidence. While in the SafeML approach [13], three different scenarios including I) repeating the measurement or requesting additional data, II) providing a human-in-the-loop procedure and III) trusting the machine learning decisions and providing confidence report are considered based on empirical cumulative distribution function (ECDF)-based statistical distance measures. The SafeML approach was not able to work with images particularly for the Convolutional Neural Network (CNN) based classifiers and more importantly the lack of consideration of p-values of statistical distance measures in the procedure could lead to a wrong decision. In other words, there are some cases where a statistical distance exists but based on an invalid associated p-value it should not be considered for the confidence evaluation. In SafeML II, the ECDF-based statistical distance measure functions have been improved with a bootstrap-based p-value evaluation. It means that in the confidence evaluation of SafeML II only the measured statistical distance value with a valid p-value will be considered and the others will be dropped from the list. Moreover, by converting the images to flatten vectors, SafeML II is able to do a pixel-wise ECDF-based statistical distance measure and generate the confidence that will be explained in the next sections.

III. ML Safety Approach

In this paper, we extend the initial idea of SafeML [13] to propose SafeML II for I) image-based classification problems and II) dealing with outliers in data. Fig. 1(a) illustrates the flowchart of SafeML II. It has two main phases: the training phase is an offline procedure and the application phase is an online procedure. In the training phase, the procedure starts with loading the trusted dataset. It is assumed that the dataset covers the majority of situations, the dataset labelling has been done perfectly and the dataset is relatively balanced. Having loaded the trusted dataset, a classifier will be trained with those data and its performance will be evaluated accordingly. In this part of the procedure, standard methods for cross-validation and explainability should also be considered. If the accuracy of the classifier and its explainability were high enough (e.g. more than 95% accuracy), the classifier will be selected and the procedure goes to the next step. Otherwise, other classifiers or

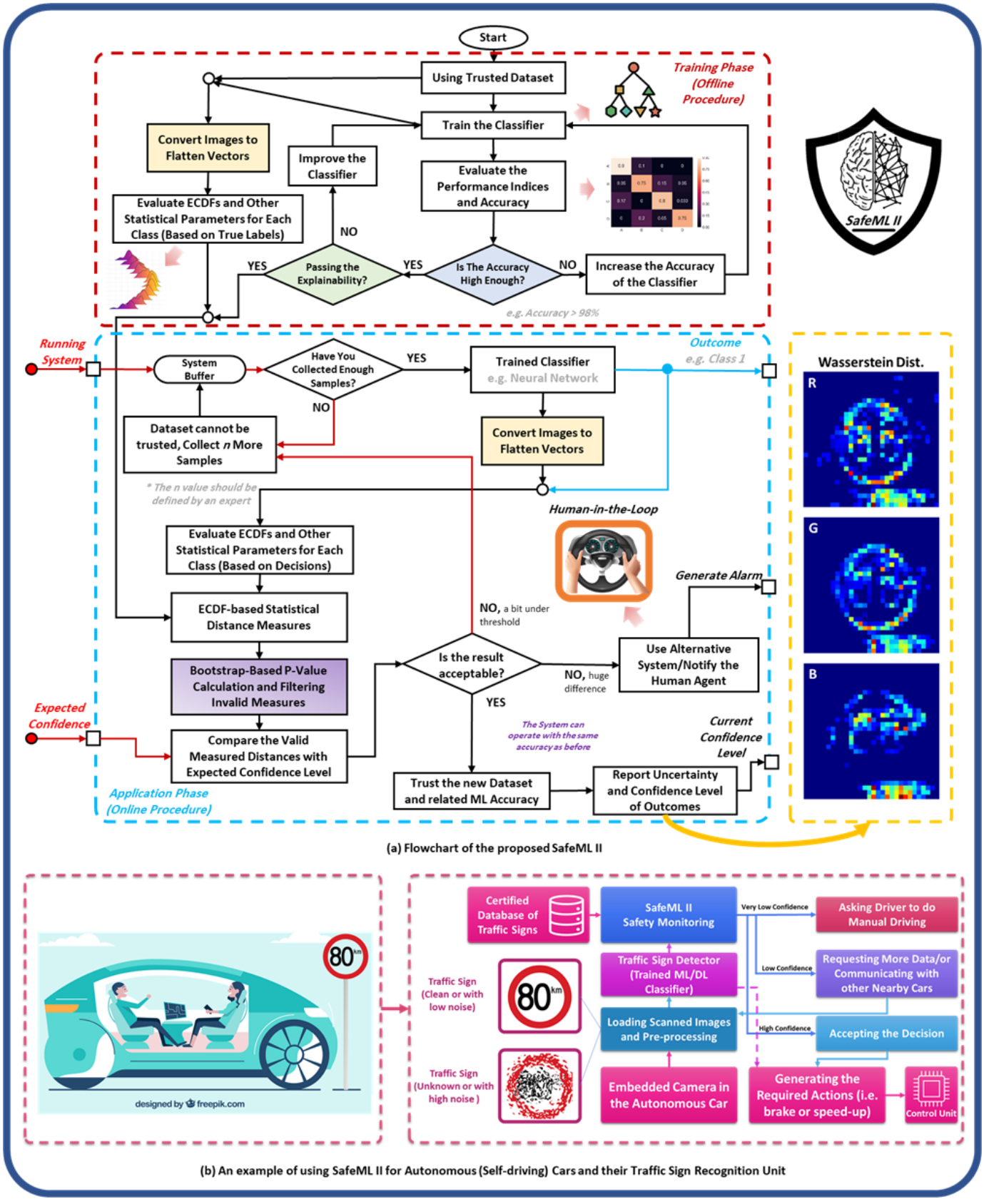

Fig. 1. SafeML II - Flowchart and Application Block Diagram

even data refinement will be needed to achieve a certain level of accuracy. Having selected the appropriate classifier, the statistical parameters of each feature in each class including cumulative distribution function, mean and variance will be

stored to be used for comparison in the next phase.

In the application phase, there would be a buffer to collect enough samples. The buffer size should be defined at design time by an expert in a way that the collected data contain the statistical characteristics of their class. Note that these upcoming data are not labelled. Having collected enough samples, the trained and tested classifier in the previous phase will be used and based on its decisions, the data will be labelled. Based on classifier decision, the statistical parameters of buffered data will be collected and compared with training dataset through empirical cumulative distribution function (ECDF)-based statistical distance measures such as Kolmogorov-Smirnov, Kuiper, Anderson-Darling, Cramer-Von Mises, and Wasserstein [14]. Moreover, in the design time, an expected confidence threshold should be defined for each statistical distance measure. The confidence level will be calculated based on the aforementioned comparison and will be again compared with the expected confidence threshold. Three different scenarios have been considered; I) when the confidence is a bit lower than the threshold, the system should collect more data, II) when the confidence has a huge difference in comparison to the predefined threshold, then it is assumed that the upcoming data have not been seen by the classifier before and a human-in-the-loop procedure should be taken into consideration, and III) when the confidence is higher than the predefined threshold the results of the classifier will be accepted and a report of the statistical comparison will be stored in the system.

To have a better understanding of the idea, the illustrated example in Fig. 1(b) is used. In this example, it is assumed that there is an autonomous vehicle and there is a specific module in the vehicle software for traffic sign recognition based on the machine learning algorithm. The main task for the machine learning algorithm is to classify the upcoming images from the vehicle's embedded camera(s) and based on a look-up table a required action will be generated to be used in the control unit. It can simply be a brake or acceleration command. The main question is "How one can make sure that the decision is always correct?" The idea of SafeML II can be a solution to this question. As an example, consider there is an 80 Km sign in the road and the vehicle's embedded camera reads it. Most of the time it is expected to be a clear image but in rare conditions such as having a faulty camera, heavy rain, fog, or a cyber-attack the image may not be clear. In such rare cases, the SafeML II can compare the images with the trusted dataset and creates confidence. For the very low confidence situation, it means that the input is not something that the trained ML algorithm has seen before and it is better to be handled by the driver (human-in-the-loop procedure). In autonomous vehicles that do not have a wheel to control such as the car like Amazon Zoox, it is suggested that a human agent from the control centre control the car remotely. It should be noted that the needed reaction time and possibilities to involve the human in the loop can be another research subject to be investigated in the future. When the confidence is low, the SafeML II may ask for more data or communicate with surrounding cars and increase the level of confidence. If the confidence is high and the upcoming images are statistically similar to the trusted dataset, the decisions can be accepted. Having a high confidence decision, the needed control command will be generated to be sent to the main control unit. All confidence reports should be stored in the system to be used for system improvement.

IV. Numerical Results

In this section, numerical results comparing the proposed approach and existing approaches in the literature are presented for a German Traffic Sign Recognition (GTSR) dataset [15]. The dataset has been released in 2011 and it includes 43 different traffic signs. The dataset is unbalanced and the number of samples for some classes can be more than the others. Regarding the cross-validation, the hold-out method is used to split 80% of the data for training and 20% for validation. It should be noted that the dataset has a separate folder for test data.

As mentioned before, the SafeML II is a model-agnostic approach that can be used on top of any machine learning classifier regardless of its structure. In this paper, a Deep Convolutional Neural Network (CNN) classifier is used because of its reputation on image classification. The following structure is used as the configuration of CNN. The input has a 2D convolution layer (Conv2D) with a filter size of 32, kernel size of 5×5 and the relu activation function. The second layer has another Conv2D with a filter size of 64, kernel size 3×3 and the relu activation function. Then, a max pooling layer with a size of 2×2 and a dropout layer with a rate of 0.25 is used. After that, another Conv2D layer with a filter size of 64, kernel size of 3×3 and relu activation function is added. A max pooling with a size of 2×2 and a dropout with the rate of 0.25 is applied on top of it. A flattened and dense layer with a size of 256, and relu activation function, with 0.5 percent dropout is used. Finally, for the output, a dense layer with the size of 43 and Softmax activation function is considered. Moreover, the Adaptive Moment Estimation (ADAM) optimiser and the cross-entropy loss function are used in the training procedure.

Using the above configuration, the performance of the CNN classifier was 0.9797 on the test dataset. The next level is to check whether the achieved accuracy is high enough or not? This part was not considered in the first version of the SafeML and it could reduce the precision of the proposed approach when a poor classifier is chosen in the offline phase. In the case of having a poor classifier, the loop should be repeated until reaching a certain level of satisfaction for accuracy. It is also possible to consider explainability approaches to make sure the trained classifier behave reasonably and focuses on the right part of the image. Assuming that the level of achieved accuracy is acceptable for safety experts, the images of each class will be separated to R, G, and B matrix and converted to the flatten vectors accordingly. As the size of each image is 30×30, the equivalent vector will be 1×900. The ECDFs of each class will be generated and stored for use in the next phase. In the online phase, the buffer size is considered as 15. In a practical scenario, the buffer size should be defined by safety experts and designers. As there was no real-time data, the test data are considered as the upcoming data and we are going to see

how the proposed approach will react to the wrong decisions. To have better visualization, class number three is chosen. This class is related to the 60 Km speed limit sign and it has 1410 images in the training dataset and 450 images in the test dataset. Various risks can be considered for miss-classification of this sign like having a lower speed and blocking the road or having a higher speed and increasing the probability of hitting pedestrians passing the street. The associated risk for miss-classification of each class can be investigated in a separate research study. The accuracy of the classifier for this class specifically was 0.9655. In other words, 435 images are detected correctly but 15 images are detected as the other classes. Based on the SafeML II procedure, the R, G, B matrix of test images are converted to flatten vectors and their ECDFs have been generated. Furthermore, using the ECDF-based statistical distance measures such as Kolmogorov-Smirnov (KS), Kuiper (K), Anderson-Darling (AD), Cramer-Von Mises (CVM), and Wasserstein (W), the statistical distances will be obtained. The first version of SafeML will jump to a comparison between statistical distance measures and the pre-defined expected confidence threshold. However, in SafeML II, a bootstrap algorithm with 1000 iterations is used to obtain the P-value and validate the measures [16]. Thus, the measures with a P-value lower than 0.05 are stored and others will be omitted. The validated statistical distance measure can be compared with the expected confidence level. It should be noted that for each ECDF-based statistical distance measure, there should be a particular expected confidence threshold predefined by a safety expert. The decision of the machine learning classifier is accepted and trusted if the distance measure is higher than the predefined threshold. Additionally, a report of the statistical distance measure will be stored in a database to be used for the further development of the system. In the situation that the statistical distance measure is 5% lower than the predefined threshold, the system may ask for further data. It should also be mentioned that in that situation, the autonomous vehicle can use other existing sources of information to validate the decision. For example, the autonomous vehicle can communicate with nearby vehicles or use GPS and pre-loaded map data. The mentioned percentage can also be changed based on the safety experts' and system designers' opinion. At the moment there is no published standard to define these levels but in the future, these parameters can be defined using the published standards. The worst scenario is that the statistical distance measure is hugely different from the expected threshold, meaning the upcoming data has not been seen by the classifier before and there is a risk of missed classification. The SafeML II idea is to put human-in-the-loop and ask the driver to make the decision. It is assumed that the driver has enough time for making the decision. However, there might be some cases where the time is restricted and SafeML II cannot be used. As mentioned before, the autonomous vehicles that do not have the wheel-based driving capability, it is suggested that a human agent from the control centre control the car remotely. The first row of Fig. 2 illustrates the Wasserstein distance measure of the 60 Km traffic sign (Class 3) for R, G, and B part of the images. As can be seen, the middle of the image has more statistical differences in all three colour layers. Besides, the blue part of the image has less statistical distance in comparison to the red and green parts of the image. As can be seen, in the first layer, a previous version of SafeML is used which has a lack of P-value based distance validation while in the second row the SafeML II is used that has the embedded P-value distance validation. Comparing the first and second row of this figure, it is clear that SafeML II has a better statistical distance representation and it does not catch the background areas of the signs. The third row of this figure illustrates a sample image where the classifier has correctly detected the sign, while the fourth row shows a sample image where the classifier was not able to detect the sign correctly. However, it seems that it can be detected by a human with careful observation. Therefore, in these cases, human-in-the-loop can help the system to make the right decision and also learn it to make better decisions in the future. The AI system can be considered as a talented and clever child that needs to work in parallel with human and become mature over time. This figure also demonstrates how the ECDF-based Wasserstein is calculated for a pixel in the image.

In Fig. 2, it is shown how SafeML can be used for image-based classification problems and how it can provide a statistical representation and explanation between wrong predictions and the ground truth. It is also shown that P-value consideration can improve the statistical explanations (illustrated in the second row of Fig. 2). The difference in results given by application of SafeML's four ECDF-based distance measures to two different datasets is illustrated in Fig. 3 (a-d). As can be seen, the KSD measures the maximum value between two ECDF.The KS distance cannot detect which ECDF has a higher value while Kuiper distance can measure two maximum up and maximum down. In a situation where two sets have the same mean value and different variances like spiral and circle benchmarks, the Kuiper distance has a better measure over KS distance. As illustrated in Fig. 3 (c), Wasserstein distance (WD) can somehow calculate the area between two ECDFs. Thus, WD will be more sensitive to a change in the geometry of the distributions. The CVM distance has similar functionality to WD, and it can perform faster. If we reduce the step size in the CVM algorithm, the results will be close to the WD ones. To have more detail on ECDF distance measures one can refer to [17]. Fig. 3 (e) provides a comparison between true accuracy, estimated accuracy by SafeML II, and Wilson Interval Confidence (WIC) bound from Klas et al. [9]. For the WIC, the z-score is chosen to be 3.29053 to gain a 99.99% confidence level. The WIC usually provides both upper bound and lower bound. To ensure the maximum safety level, only the lower bound is considered. From the existing 43 classes in the GTSRB, 5 safety-critical related classes have been chosen for the comparison. The results show that in most cases the Wasserstein-based accuracy estimation has less error. For two cases the Wasserstein algorithm was not successful: for class number 11 (Cross Road Ahead), the Anderson-Darling estimation has less error and for class number 13 (Yield), the low band Wilson Interval has better accuracy. It should be noted that the WD, CVMD and ADD are not always bounded between 0-1. However, based on our experiments they are always correlated with accuracy. To

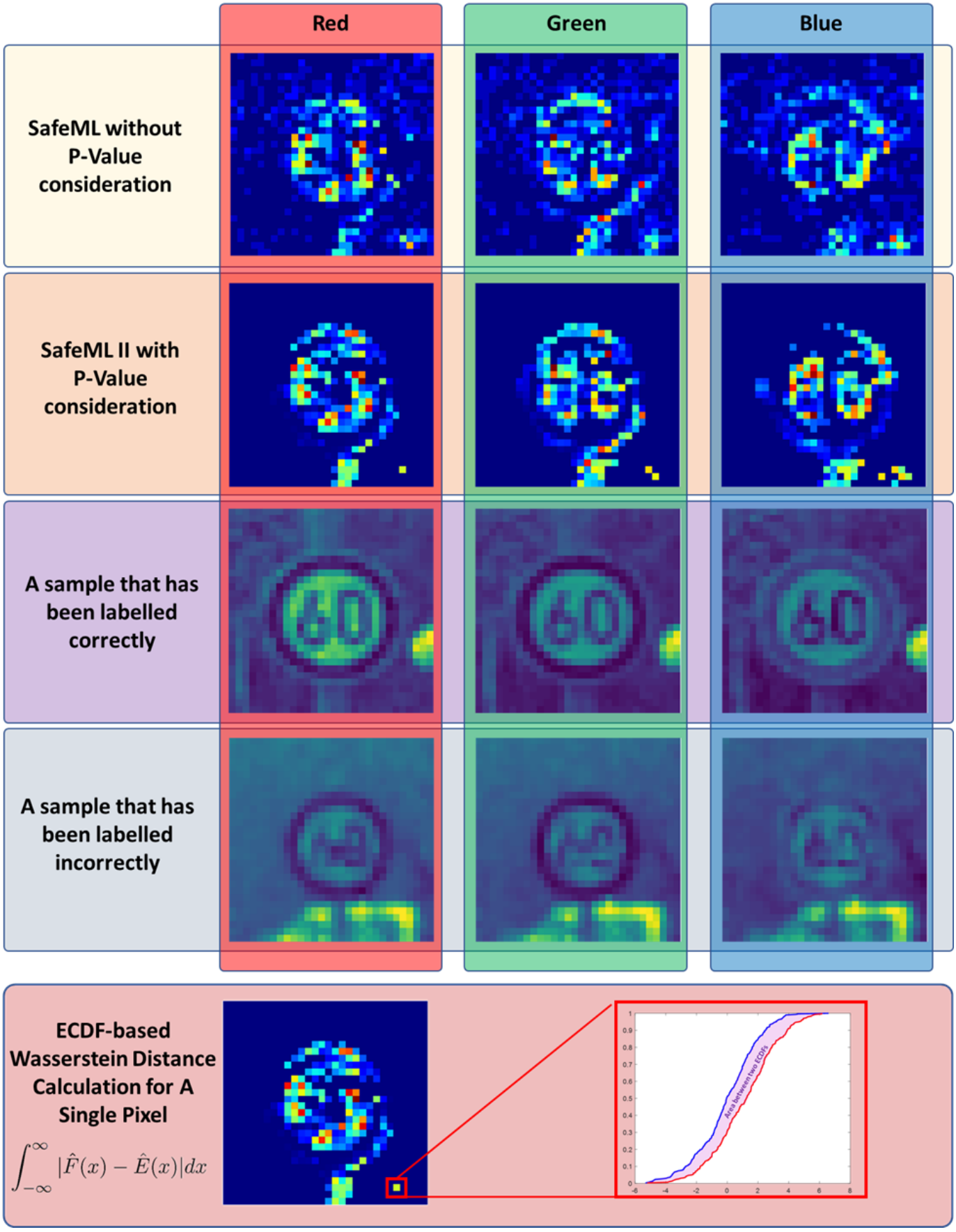

Fig. 2. Sample results of SafeML II with Wasserstein Distance and considering p-values (class number 3)

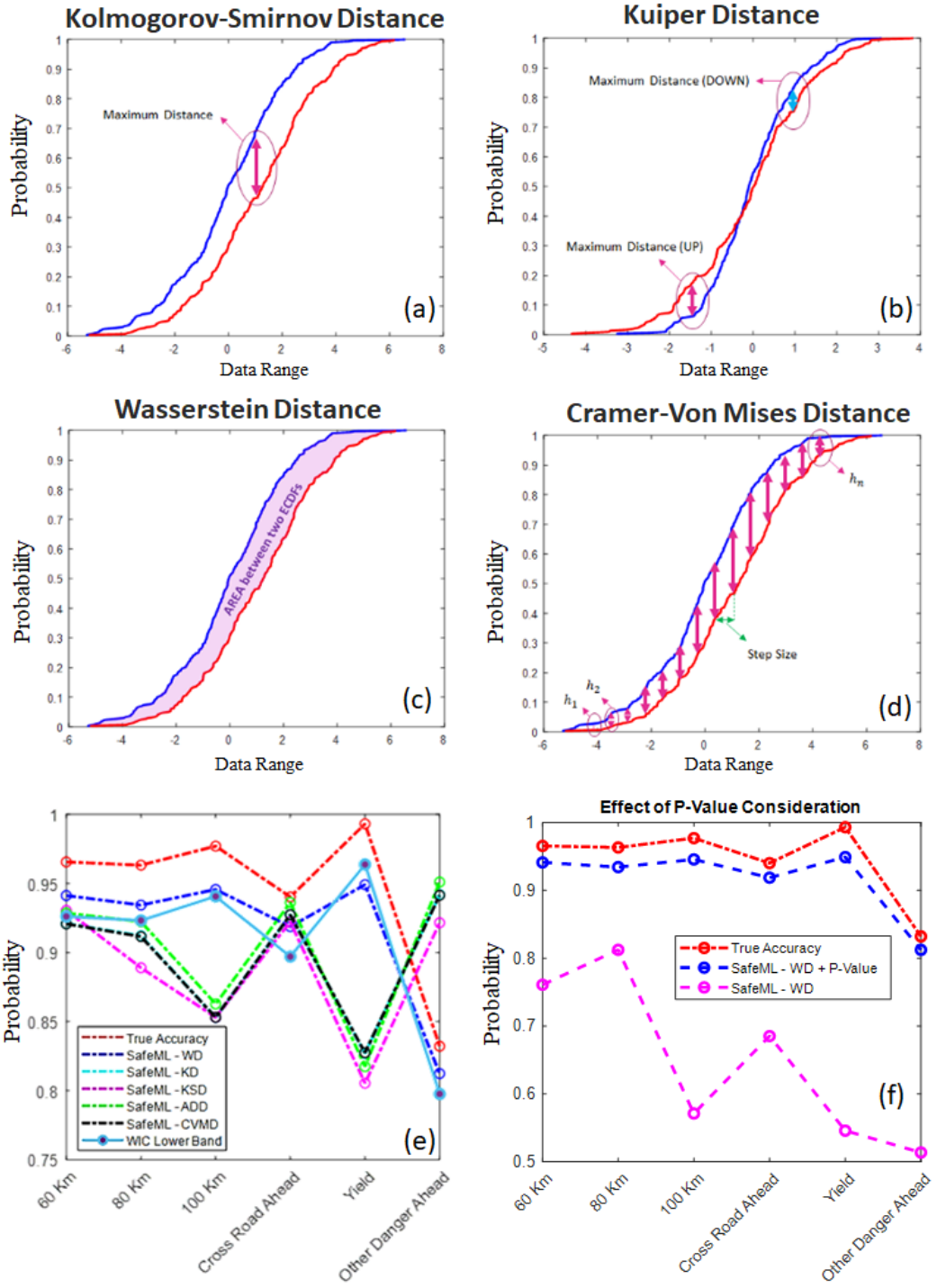

Fig. 3. (a) Kolmogorov-Smirnov Distance, (b) Kuiper Distance, (c) Wasserstein Distance, (d) Cramer-Von Mises Distance, (e) Comparison between true accuracy, estimated accuracy by SafeML II and Klas et al. [9], (f) Comparison between true accuracy, WD with and without P-Value consideration)

clarify the effect of P-Value consideration, the WD algorithm is selected as the best performing measure for GTSRB and its results with and without P-Value consideration are compared with true accuracy as shown in Fig. 3 (f). The original SafeML [13] was successful for feature-based dataset. However, our experiments show that it is not always successful. For example, in GTSRB, the WD measure without P-Value consideration has failed to detect true accuracy changes while WD with P-Value consideration was successful. Generally, using ECDF-based distance measures with P-Value consideration are more reliable and less noisy especially when the results are going to be used for statistical explainability purposes.

In this paper, we have only focused on traffic sign recognition and the idea can be integrated with other safety-related parts of autonomous vehicle software to cover wider safety perspectives. For example, in [18], it was explained how to build an integrated safety model and consider different components of a cooperative operation scenario of autonomous vehicles. The results of SafeML II can be used as an input in the proposed safety model in that work to improve confidence in the provided assurance. It should be noted that the SafeML II concept has some limitations. For example, it can only work with Machine Learning classifiers, while having the SafeML II concept to work for prediction and regression algorithms is still an open research question. Moreover, we currently investigate what specific characteristics of a dataset can lead to a better ECDF-based statistical accuracy estimation in run time. Due to the use of the buffering technique, in some time-critical applications, the proposed approach may not be able to handle a sudden shift of data efficiently within a very short period of time. Generally, for safety-critical systems, it is crucial to limit the possibility of making unsafe decisions and actions that may be caused by a sudden shift in the data. A potential solution to track sudden changes in the incoming data is to use the soft clustering models [19], which offer a way to evaluate the changes through a natural measure by computing it directly from models. Moreover, in this paper, we introduce the model-agnostic version of SafeML where we are unable to go inside any ML/DL algorithm. In our future research work, we will address the model-specific version of SafeML where we will be able to utilize CNN's middle layer to avoid pixel-level alignment requirements.

V. Conclusion

The rapid growth of artificial intelligence applications in various domains and particularly in autonomous vehicle software raise concerns in different perspectives such as AI safety, AI responsibility, AI explainability and interpretability, human-in-the-loop AI, and AI trustworthiness. This paper addressed the issue of distributional shift and its implications for the safety of machine learning or deep learning classification tasks in autonomous vehicle software. The paper proposed SafeML II by extending SafeML to improve its capabilities for the human-in-the-loop procedure and ECDF-based statistical distance measures, and applies them to image-based classification algorithms in a model-agnostic way. SafeML II improves the ECDF-based statistical distance measure functions using bootstrap-based p-value calculation. The proposed SafeML II approach is generic in nature, therefore, we believe it can be integrated with traditional safety assurance methods to enable them to provide assurance for ML/AI models and also to increase confidence in the provided assurance.

Code availability

Regarding the research reproducibility, codes and functions supporting this paper are published online at GitHub: https://github.com/ISorokos/SafeML.

Acknowledgements

This work was supported by the Secure and Safe Multi-Robot Systems (SESAME) H2020 Project under Grant Agreement 101017258.

References

[1] Waymo LLC, "Waymo safety report," pp. 1–48, 2020.
[2] S. Kabir and Y. Papadopoulos, "Computational Intelligence for Safety Assurance of Cooperative Systems of Systems," *Computer*, vol. 53, no. 12, pp. 24–34, 2020.
[3] A. F. Magnussen, N. Le, L. Hu, and W. E. Wong, "A survey of the inadequacies in traffic sign recognition systems for autonomous vehicles," *International Journal of Performability Engineering*, vol. 16, no. 10, 2020.
[4] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars: chances and challenges," in *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, 2018, pp. 35–38.
[5] R. Salay, R. Queiroz, and K. Czarnecki, "An analysis of iso 26262: Using machine learning safely in automotive software," *arXiv preprint arXiv:1709.02435*, pp. 1–6, 2017.
[6] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.
[7] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*, pp. 1–29, 2016.
[8] L. Gauerhof, R. D. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, and I. Habli, "Assuring the Safety of Machine Learning for Pedestrian Detection at Crossings," in *Proceedings of the 39th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*. Springer Nature, 2020, pp. 197–212.
[9] M. Kläs and L. Sembach, "Uncertainty wrappers for data-driven models," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2019, pp. 358–364.
[10] L. Jöckel, M. Kläs, and S. Martínez-Fernández, "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)*. IEEE, 2019, pp. 540–541.
[11] L. Jöckel and M. Kläs, "Increasing trust in data-driven model validation," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2019, pp. 155–164.
[12] M. Kläs and L. Jöckel, "A framework for building uncertainty wrappers for ai/ml-based data-driven components," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2020, pp. 315–327.
[13] K. Aslansefat, I. Sorokos, D. Whiting, R. T. Kolagari, and Y. Papadopoulos, "SafeML: Safety Monitoring of Machine Learning Classifiers Through Statistical Difference Measures," in *Model-Based Safety and Assessment: 7th International Symposium, IMBSA 2020, Lisbon, Portugal, September 14-16, 2020, Proceedings*, vol. 12297. Springer Nature, 2020, p. 197.
[14] M. M. Deza and E. Deza, "Distances in probability theory," in *Encyclopedia of distances*. Springer, 2009, pp. 1–583.
[15] "German Traffic Sign Recognition Benchmarks." [Online]. Available: https://benchmark.ini.rub.de/?section=gtsrb
[16] E. Gilleland, "Bootstrap methods for statistical inference. part ii: Extreme-value analysis," *Journal of Atmospheric and Oceanic Technology*, vol. 37, no. 11, pp. 2135–2144, 2020.

[17] K. Aslansefat. (2020) How to make your classifier safe. [Online]. Available: https://towardsdatascience.com/how-to-make-your-classifier-safe-46d55f39f1ad

[18] S. Kabir, I. Sorokos, K. Aslansefat, Y. Papadopoulos, Y. Gheraibia, J. Reich, M. Saimler, and R. Wei, "A runtime safety analysis concept for open adaptive systems," in *International Symposium on Model-Based Safety and Assessment*. Springer, 2019, pp. 332–346.

[19] A. Abdullatif, F. Masulli, and S. Rovetta, "Clustering of nonstationary data streams: A survey of fuzzy partitional methods," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1258, 2018.