Towards Multi-timescale Online Monitoring of AI Models: Principles and Preliminary Results

Fateh Kaakai, Paul-Marie Raffi

Towards Multi-timescale Online Monitoring of AI Models: Principles and Preliminary Results

Fateh Kaakai[1,2,*], Paul-Marie Raffi[2]

[1] *Thales Research & Technology France - 1, avenue Augustin Fresnel Palaiseau 91767 Cedex France*
[2] *IRT SystemX, Nano-INNOV – Bât 863, 2, Boulevard Thomas Gobert, 91120, Palaiseau*

Abstract

Online monitoring is an architectural pattern well-known to safety engineers, but it had to be adapted to AI technologies. In this paper, an innovative multi-time scale online monitoring architecture is presented. The main idea is to combine several monitoring timescales - Present-Time Monitoring (PTM), Near-Past Monitoring (NPM), and Near-Future Monitoring (NFM) - on different monitoring assets (inputs, internal states, and outputs of the AI model) to ensure a high anomaly detection rate by design of the online monitor.

Keywords

Online monitoring, Multi-timescale, AI, Machine Learning, Model, Safety, Anomaly Detection

1. Introduction

In the industry, it is commonly established that the main objective of online monitoring of AI models (also called Run Time Assurance in [21] or safety control structure in [25]) is to detect (*i*) any deviation of the AI component deployed in production from the expected behavior (i.e., intent specified at the system level and allocated to the AI model), and (*ii*) precursors of the occurrence of failure conditions (i.e., feared events at the system boundaries) based on a predefined set of safety properties. Deploying a monitoring component running in parallel with the AI model is a practical way to manage the risk induced by a model for which it is not possible or feasible to formally demonstrate the achievement of the performance and the safety objectives resulting from the system analyses. Online monitoring is an architectural pattern well-known to safety engineers, but it had to be adapted to AI technologies. In an ideal world, the AI model can perform its prediction over its entire Operational Design Domain (ODD) with the expected level of performance (e.g., 99.9% correct predictions, and this accuracy is maintained over time in operation). However, in practice, if we consider for example machine learning models in many recent papers, most of the time it is very difficult to achieve more than 99% accuracy (see for example the tables of results in [22, 23, 24]), which is an average of one wrong prediction out of 100 inferences in production. But should we hastily conclude that 1% of bad prediction systematically triggers unexpected behavior leading to a system failure condition? In practice, from the industrial experience of the authors and for a wide range of industrial applications, a single error does not directly lead to hazardous or catastrophic events, because the system design has eliminated Single Points Of Failure (SPOF) (e.g., application of the following guidelines ARP4754A [26] and ARP4761 [27] in the aeronautical domain). Therefore, based on this assumption (no SPOF in the system), it implies that a single failure of an AI component (i.e., an incorrect prediction at a given time) cannot

SafeAI2023: The AAAI's Workshop on Artificial Intelligence Safety,February 13–14, 2023, Washington, D.C.
*Corresponding author
@: fateh.kaakai@thalesgroup.com (Fateh Kaakai) ; paul-marie.raffi@ext.irt-systemx.fr (Paul-Marie Raffi)
†The paper has been entirely written by Fateh KAAKAI. Paul-Marie RAFFI has contributed to the online monitor development and the use case study.

directly lead to hazardous or catastrophic events. However, what about the case where the AI component has persistent failures (i.e., the model fails to infer the correct prediction at a given point in time, and continues to fail during a subsequent time interval)? This could increase the residual risk due to a higher probability of having (during that time interval where the AI component continues to fail) a combination of multiple internal failures in the system leading to a system failure condition. To detect persistent failures, considering the dynamics of the system and thus the "time" variable is a major issue. In this paper, an innovative multi-time scale online monitoring architecture is presented. The main idea is to combine several monitoring timescales - Present-Time Monitoring (PTM), Near-Past Monitoring (NPM), and Near-Future Monitoring (NFM) - on different monitoring assets (inputs, internal states, and outputs of the AI model) to ensure a high anomaly detection rate by design of the online monitor.

2. Summary of Related Works

To tackle the topic of monitoring AI models, some works started to define a taxonomy of anomalies that are specific to AI technologies [1, 2, 3] but, to the best of our knowledge, no taxonomy is an undisputed reference. Other works tried approaches to perform runtime verification such as Monitoring Based on Past Experiences [4, 5], and Monitoring Based on Inconsistencies During Inference [6]. We can also find many papers on Out of Distribution Detection using either Data-Driven Out-of-Distribution Detection [7, 8, 9], Detection by reconstruction [8, 10], Detection by test-time adversarial attacks [11, 12], or Anomaly Detection for Time Series [13]. Another group of work is dedicated to Uncertainty Prediction including, Bayesian Neural Networks [15, 16], MC Dropout [17], Ensemble Methods [18], and Single-forward uncertainty estimation [19, 20].

3. Multi-timescale Monitoring

The context of the online monitoring function is described in Figure 1 below.
The AI-based product consisting of one or several integrated AI models is depicted in the black oval. This product can be an item (i.e., a component), or a subsystem of an entire system. The generic term "product" is used in the following. The product receives at inference time operational data from sensors.

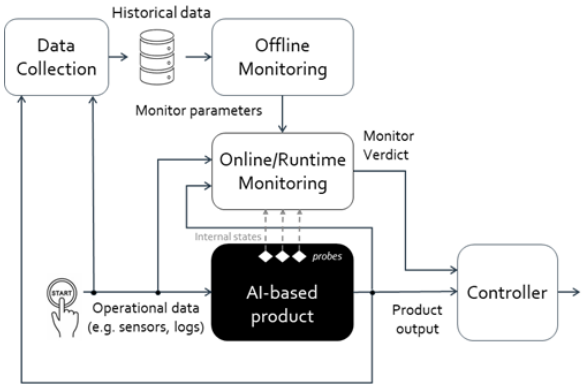


Figure 1: Context of the Online Monitor function

Above the product, in a white oval, the online monitoring item receives both the external inputs and outputs of the product, as well as some information about the internal states of the product using pre-designed probes placed in the product's software code or hardware. At the top of Figure 1, an item in charge of continuously collecting all relevant operational data is usually required to feed a complementary offline monitoring function. The offline monitoring function may have several objectives according to the use case such as (but not limited to): (*i*) calculating offline metrics, (*ii*) fine tune some of the online monitor parameters, (*iii*) detecting data and concept drifts, and (*iv*) act as a hypervisor of the online monitor. At the bottom right-hand side of Figure 1, a controller is responsible for synthesizing the output produced by the product and the verdict of the monitor to compute, based on certain business logic (that is in general specific to the use case), the final output, which is so-called the “safe output”.
To illustrate the product to be monitored, consider the very simplified didactic example in Figure 2, which represents a linear physical phenomenon $y = f(t) = a \cdot t + b$ to be approximated by an AI model $\hat{y} = \hat{f}(t_k)$, where t_k is the system clock which also clocks the monitoring item (at each time t_k the device acquires data to produce a verdict; t_0 is the product start-up time).

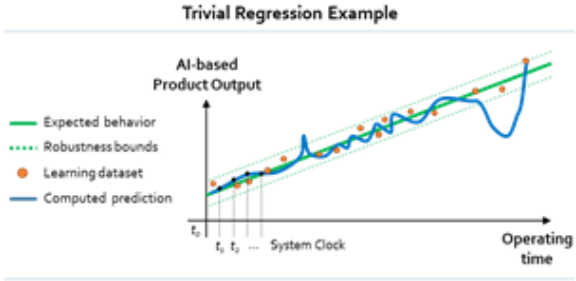


Figure 2: Illustration of the AI-based Product

Let's also assume that system requirements specify a set of properties to be satisfied by the product in operation. These properties are business-driven and thus specific to each use case. In general, these properties may be (but are not limited to) (*i*) functional properties related to the nominal expected behavior of the product like performance requirements, (*ii*) safety properties identified by safety risk analyses, (*iii*) security properties determined by security risk analyses, (*iv*) explainability properties coming from human factor analyses. To keep the logic of a simplified didactic example, consider that there is only one general property materialized by robustness bounds depicted by the green dashed segments in Figure 2. The area bounded by these two green dashed segments defines the validity domain $\mho$ of the product output $\hat{y}$. The very simplified general safety property[2] can therefore be expressed as follows:

$$\forall t_k \geq t_0, \hat{f}(t) \in \mho \qquad (1)$$

Regarding the design of the product, since sufficient data were collected and are available to characterize the physical phenomenon to be modeled, it has been decided to use Machine Learning (ML) technology to design the product (e.g., using an artificial neural network). The ML model is obtained after several iterations of learning and is depicted by the blue curve in Figure 2. It is deliberately not perfect. Indeed, it is possible to observe several operating points of the ML model output $\hat{y}$ fall outside the validity domain $\mho$ and do not satisfy the safety property (1). The online monitoring function aims to detect all operating points that violate the system properties – let's call them by the generic term anomaly in the rest of this paper. To be efficient, the online monitor should ensure a sufficient anomaly detection rate, and this is precisely the ultimate goal of the multi-scale monitoring framework which is the main contribution of this paper.

The principle of multi-scale monitoring is described in Figure 3. It consists in combining several monitoring timescales: monitoring of the product at the *present time*, monitoring over a configurable time window in the *near past*, and monitoring over a configurable time window in the *near future*.

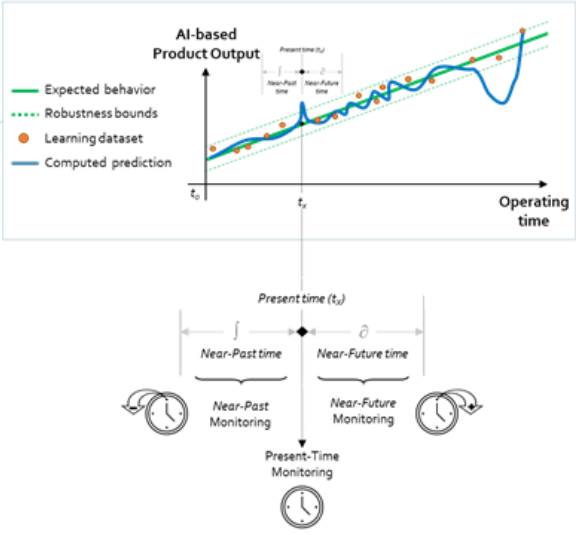
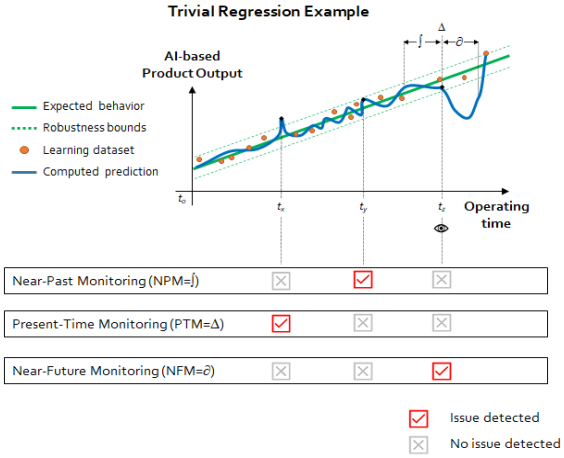


Figure 3: Principle of the multi-timescale online monitoring

To illustrate the combination of these three different timescale monitoring functions, let us continue the discussion on the trivial example of Figure 4. At time t_x, the ML model output $\hat{y}$ overpasses the robustness boundaries, and it is expected that the Present Time Monitoring (PTM) will be able to detect such abnormal behavior. Between times t_x and t_y, it is possible to observe that $\hat{y}$ starts to unexpectedly oscillate. It is an unintended behavior that could be a precursor of a failure of the ML model. Since this oscillation phenomenon should be observed and confirmed on several clock cycles, it is expected that the Near Past Monitoring (NPM) will be the appropriate monitoring timescale to detect such oscillation anomaly.



[2] In section 5 (Application), an industrial use case is presented with more complex properties to be monitored.

Figure 4: Combination of the multi-timescale monitoring functions PTM, NPM and NFM

At time t_z, $\hat{y}$ has an abrupt trend that will make it overpass the robustness boundaries at the next clock cycles. Here, the Near-Future Monitoring (NFM) is the most appropriate monitoring function to detect such potentially abnormal behavior since it is based on trend analysis.Through this didactic example, one can observe that an efficient combination of these three different monitoring timescales – NPM, PTM, and NFM – allows one to detect several classes of anomalies and to achieve this by designing a high online detection rate when the AI model is in production.

4. Industrial Design Principles

In the previous sections, a first design principle has been presented through the new multi-timescale monitoring framework that aims at increasing by design the anomaly detection rate. However, there are many other design principles of online monitors that are important as well. Below is a synthesis of the main industrial design principles collected and formalized by major international industrial groups within the frame of the French research program Confiance.ai[3]. All these design principles are not detailed in this paper since each of them would require a full technical paper to be comprehensively presented.

- *Design Principle 1*: The monitoring function should by design ensure completeness of anomaly detection while minimizing false alarms
- *Design Principle 2*: The sophistication of the monitoring function should be proportionate to the criticality level of the AI function
- *Design Principle 3*: The monitoring function should be smart to manage complexity and performance issues
- *Design Principe 4*: The monitoring function should not have any safety adversarial common mode of failure with the monitored AI function
- *Design Principle 5*: The monitoring function itself should not has an unacceptable impact on the system safety and security (innocuity)

In the next section, an industrial use case from the automotive domain is presented to illustrate some of the concepts presented earlier.

5. Application

The application used to present some results related to multi-timescale online monitoring is called the Renault Welding Use Case.

Figure 5: Renault Welding Use Case

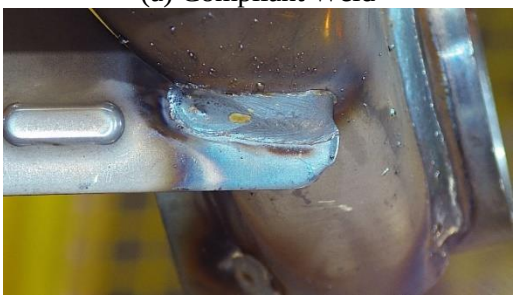
The industrial context is a plant producing mechanical components used for the ground connection of motor vehicles and the mechanical parts of interest in this use case are only the parts of the rear axle. During the manufacturing process shown at the top of Figure 5 (see Operational Platform (OP) #120), metal parts are welded together. The mechanical quality of the final component depends on the quality of the weld.

Until now, a systematic inspection of the weld is carried out by a specialized human operator on a screen like the one at the bottom of Figure 5 (see Display OP#120). The screen displays different

[3] See www.confiance.ai involving more than 40 partners including large industrial groups such as: Airbus, Air Liquide, Atos, Naval Group, Renault Group, Safran, Sopra Steria, Thales, Valeo, and others (full list on the web site).

photos of the same weld taken by different cameras from different angles. Based on its experience, the human operator classifies the weld as “compliant”, “not compliant” or “unknown” (see examples of welds in **Figure 6**). This last status "unknown" leads to a further deeper technical evaluation of the manufacturing process.

(a) Compliant Weld



(b) Non-Compliant Weld

Figure 6: Examples of welds

In practice, the overwhelming majority of welds are compliant (robotized welding), and from a human factor perspective, this situation is likely to decrease the attention of the operator in charge of quality control. To mitigate this risk, Renault launched a project to develop an AI-based system to assist the operator in charge of the quality control of welds as depicted in Figure 7.

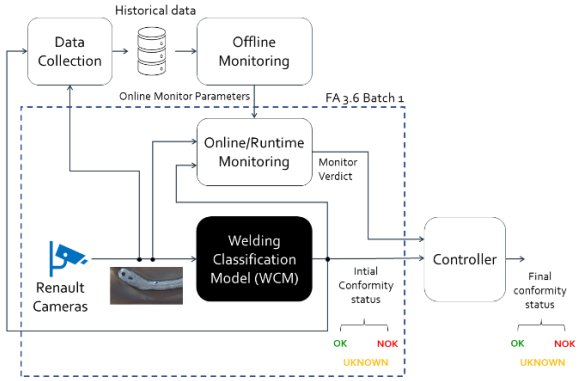


Figure 7: Online Monitoring of the Welding Classification Model

The AI-based product in Figure 7 is called Welding Classification Model (WCM) and it performs an automated preliminary conformity assessment of the weld quality. The WCM is developed using supervised ML technology based on labeled datasets containing historical data of compliant and non-compliant welds. The design details of the WCM are not important in this paper since it is considered a black box by the online monitor that only looks at its inputs and its outputs and not at its internal states as shown in Figure 7). The WCM provided by Renault reaches very good performance (measured with an f1 score) but only on a given domain, called ODD, that is characterized according to operational parameters. Based on a dedicated study of the ODD done with Renault representatives, two operational parameters have been considered in this study as the most impacting the performance of the ML Model (and therefore of the correctness weld conformity classification): (*i*) image brightness and (*ii*) image blur. Thus, the 2 properties to be monitored are expressed as follows:

$$\forall t_k \geq t_0, im_k \in \mho_{Brightness} \quad (2)$$

$$\forall t_k \geq t_0, im_k \in \mho_{Blur} \quad (3)$$

Where im_k is the image received by the WCM at time t_k and $\mho_{Brightness}$ and $\mho_{Blur}$ are respectively a projection of the full WCM ODD on the two targeted operational parameters – i.e., image brightness and image blur. Besides, $\mho_{Brightness}$ and $\mho_{Blur}$ are not calculated theoretically, they are determined based on test campaigns with augmented data as illustrated for the brightness ODD in Figure 8.

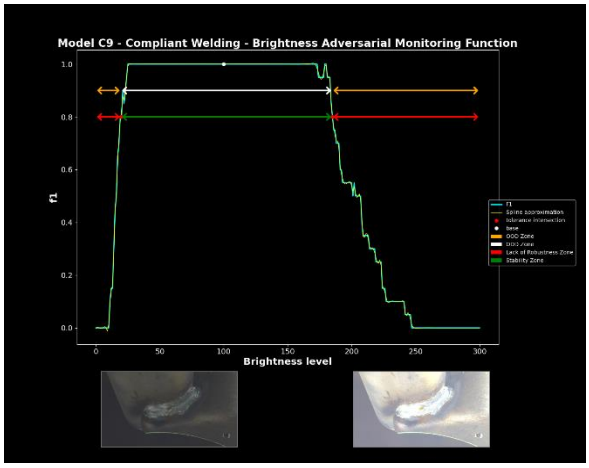


Figure 8: WCM Brightness ODD characterization

Once $\mho_{\text{Brightness}}$ and $\mho_{\text{Blur}}$ are determined, it is possible to develop dedicated monitoring functions to detect Out-Of-Distribution (OOD) input images since we can observe in Figure 8 that the f1 performance score of WCM drops sharply outside $\mho_{\text{Brightness}}$ (depicted with the white arrow at the top of Figure 8). And it is the same for the blur (not represented in the paper to avoid overloaded information). The rules-based design of the online PTM monitoring functions for brightness and blur OOD detection are detailed in Tables 1, 2, and 3. Examples of anomalies detected in the Welding use case are represented in Figure 10 (brightness anomalies) and Figure 9 (blur anomalies).

Table 1

Standard Brightness Detection

Description		Design & Implementation	
Objective Extract the degree of brightness from an image. Return a status telling if this degree of brightness can impact the prediction of the model on this image.		**Design principles** • The Area of Interest is extracted from the image. • A formula of perceived brightness is applied on the mean of RGB values of the Area of Interest • This perceived brightness is compared with the values min and max of reference tuned for this category of images • Compute an estimation of the degree of brightness: batch 2	**Detailed design** Perceived brightness formula: $\text{Sqrt}(0.299 \times \text{mean}(R)^2 + 0.587 \times \text{mean}(G)^2 + 0.114 \times \text{mean}(B)^2)$
Monitoring Class Rules-based Present-Time Monitoring (PTM) Subclass: Out of Distribution Monitoring (ODM)	**Recommended usage / Limitations** The first value of reference comes from the analysis of: • the Robustness • Graphs of distribution of brightness by category of images		

Figure 9: Brightness Anomaly Detection on the Welding Use Case based on PTM monitoring

Table 2

Standard Blur Detection

Description		Design & Implementation	
Objective Extract the degree of blur from an image. Return a status telling if this degree of blur can impact the prediction of the model on this image.		**Design principles** • The Area of Interest is extracted from the image. • Area of Interest is converted to gray. • Variance of Laplacian is computed on the color values of this gray Area of Interest. • This variance is compared with the value of reference tuned for this category of images • Compute an estimation of the degree of blur: batch 2	**Detailed design** The Laplacian is computed by filtering the gray image with the following 3×3 aperture: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ The variance of the Laplacian is a measure of the spread of its distribution.
Monitoring Class Rules-based Present-Time Monitoring (PTM) Subclass: Out of Distribution Monitoring (ODM)	**Recommended usage / Limitations** The first value of reference comes from the analysis of: • the Robustness • the distance between the camera and the object • Graphs of distribution of blur by category of images		

Figure 10: Blur Anomaly Detection on the Welding Use Case based on PTM monitoring

The performance of the developed PTM monitoring functions has been evaluated by the LNE[4] which is an independent partner of the Confiance.ai Program specializing in calibration, testing, and certification under the trusteeship of the French Ministry for the Economy and Finance with oversight for Industry.

LNE randomly selected 11,000 images for the evaluation set, and identified the following evaluation metrics:

- Analysis of the classification of the monitor compared to the noise for each image:
  - True positive: the image has a medium noise or important noise, and the monitor raised an alarm
  - True negative: the image has a slight noise, or no noise, and the monitor did not raise an alarm
  - False positive: the image has a slight noise, or no noise and the monitor raised an alarm
  - False negative: the image has a medium noise or important noise, and the monitor did not raise an alarm
- Using these four values, the precision, recall, and f-measure are computed
  - Precision: total of true positives by the total of detected positives (true and false)
  - Recall: the total of true positives by the total of real positives (true positives and false negatives)
  - F-measure: harmonic mean of the precision and recall

Table 3
LNE results by type on anomaly

Noise	Precision/Recall	F-measure
V motion blur[5]	0.68/**0.90**	0.77
H motion blur[6]	0.68/**0.96**	0.80
Brightness dark	0.79/**0.99**	0.88
Brightness light	0.75/**1.0**	0.85

The recall metric is very important in domains such as automotive quality controls where you want to minimize the chance of missing positive (i.e., missing to detect a non-compliant weld) by predicting false negatives (i.e., a non-compliant weld is predicted as a compliant one and there is no alarm sent by the monitor). These are typically cases where missing a positive case has a much bigger safety impact than wrongly classifying something as positive.

These evaluation results show that the rule-based PTM OOD functions have a good or a very good recall ($90\% \leq \text{recall} \leq 100\%$). However, the precision results (and thus the f-measure scores) show that the number of false positive alarms is still high and needs to be reduced in a further version of the monitoring functions.

There is no result presented in this paper on NPM and NFM functions since the development of these monitoring functions are in progress within the Confiance.ai Program. The results of ongoing research on NPM and NFM will be published in future papers, as well as the results of the integrated multi-timescale monitor combining PTM, NPM, and NFM into a single monitoring item.

6. Acknowledgments

This work has been supported by the French government under the "France 2030" program, as part of the SystemX Technological Research Institute within the Confiance.ai Program (www.confiance.ai).

A special thanks to Guillaume BERNARD from LNE who contributed to the independent evaluation of the monitor, and to Dominique TACHET & Meriem LAFOU from Renault who provided valuable support on the Renault Welding use case.

7. References

[1] Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE.

[2] Meinke, A. and Hein, M. (2020). Towards neural networks that provably know when they don't know. In ICLR.

[3] Ahmed, F. and Courville, A. (2020). Detecting semantic anomalies. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 3154–3162.

[4] https://www.lne.fr/en
[5] Vertical motion blur
[6] Horizontal motion blur

[4] Mohseni, S., Pitale, M., Singh, V., and Wang, Z. (2019). Practical solutions for machine learning safety in autonomous vehicles. arXiv preprint arXiv:1912.09630.

[5] Hecker, S., Dai, D., and Van Gool, L. (2018). Failure prediction for autonomous driving. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 1792–1799. IEEE.

[6] Zhou, W., Berrio, J. S., Worrall, S., and Nebot, E. (2019). Automated evaluation of semantic segmentation robustness for autonomous driving. IEEE Transactions on Intelligent Transportation Systems, 21(5):1951–1963.

[7] Mohammadi, B., Fathy, M., and Sabokrou, M. (2021). Image/video deep anomaly detection: A survey. arXiv preprint arXiv:2103.01739.

[8] Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.

[9] Daxberger, E. and Hernández-Lobato, J. M. (2019). Bayesian variational autoencoders for unsupervised out-of-distribution detection. arXiv preprint arXiv:1912.05651.

[10] Xia, Y., Zhang, Y., Liu, F., Shen, W., and Yuille, A. (2020). Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In ECCV.

[11] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[12] Lee, K., Lee, K., Lee, H., and Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Advances in Neural Information Processing Systems, pages 7167–7177.

[13] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

[14] Beggel, L., Kausler, B. X., Schiegg, M., Pfeiffer, M., and Bischl, B. (2019). Time series anomaly detection based on shapelet learning. Computational Statistics, 34(3):945– 976.

[15] Graves, A. (2011). Practical variational inference for neural networks. In NeurIPS.

[16] MacKay, D. J. (1992). A practical Bayesian framework for backpropagation networks. Neural Computation, 4(3).

[17] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In ICML.

[18] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In NeurIPS.

[19] Malinin, A. and Gales, M. (2018). Predictive uncertainty estimation via prior networks. In NeurIPS.

[20] Sensoy, M., Kaplan, L. M., and Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. In NeurIPS.

[21] Peterson, E. M., DeVore, M., Cooper, J., and Carr, G. (2020). Run Time Assurance as an Alternate Concept to Contemporary Development Assurance Processes, NASA report NASA/CR–2020-220586, https://ntrs.nasa.gov/citations/20200003114.

[22] Ahmed, M., Hashmi K. A., Pagani, A., Liwicki, M., Stricker, D., and Afzal, M. Z. (2021) .Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments, In Sensors. https://pdfs.semanticscholar.org/5040/0b478dda3eebb966f71e8d8f90718a0e2854.pdf

[23] Schmarje, L., Santarossa, M., Schröder, S-M., and Koch, R. (2020). A Survey on Semi-, Self- and Unsupervised Learning in Image Classification. In IEEE Acess. https://arxiv.org/pdf/2002.08721.pdf

[24] D. Dakshayani Himabindu and S. Praveen Kumar (2021). Survey on Computer Vision Architectures for Large Scale Image Classification using Deep Learning. In International Journal of Advanced Computer Science and Applications. https://pdfs.semanticscholar.org/03e1/3f250da93bcaf1d760fe40f97e465e5083fa.pdf

[25] Dobbe, R. I. J. (2022). System Safety and Artificial Intelligence. In The Oxford Handbook of AI Governance. https://academic.oup.com/edited-volume/41989/chapter/377785597

[26] SAE International (2010). Guidelines for Development of Civil Aircraft and Systems, ARP4754A.

[27] SAE International (1996). Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, ARP4761.