Toward Statistical Risk Analysis for Aerospace Systems with AI/ML Components

–

Yuning He[1] and Konstantin Dmitriev[2]

[1] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
yuning.he@nasa.gov

[2] *Institute of Flight Systems Dynamics, Technische Universität München, Germany*
konstantin.dmitriev@tum.de

ABSTRACT

Aerospace systems increasingly rely on machine learning (ML) and AI components. Such safety-critical applications require careful verification and validation (V&V), and certification. A central task in safety certification is risk assessment: for a large number of off-nominal failure conditions, it must be demonstrated that they do not pose undue risk during operation. Events with major, hazardous, or catastrophic consequences must only occur extremely infrequently.

This paper addresses the statistical modeling and analysis of rare events in AI/ML-based components at both the system- and component-level within the SYSAI analysis framework. Here, we focus on typical perception-related functional tasks that are found in unmanned aerial vehicles (UAVs) or supporting aircraft operations, specifically an Autonomous Centerline Tracking (ACT) system and an Autonomous Emergency Braking System (AEBS). We provide an overview of the SYSAI statistical learning and analysis framework and discuss challenges for the analysis of failure modes and rare events in systems with AI/ML components. We describe how SYSAI can efficiently generate and evaluate scenarios that are likely to contain rare events. These experiments support the estimation of reliable probability distributions for such events and the analysis of temporal dependencies between individual input images.

1. INTRODUCTION

As machine learning (ML) and AI components become increasingly integrated into safety-critical aerospace systems, ensuring system safety and reliability through rigorous certification processes is essential. Standards for certification of ML-based components and systems, like the EASA guidelines (*EASA Concept Paper: First usable guidance for Level 1 machine learning applications*, 2021; *EASA Concept Paper: First usable guidance for Level 1&2 machine learning applications*, 2023) as well as approaches based upon traditional system development standards, for example, following DO-178C (RTCA, 2011), are under development.

A central component of safety certification is risk assessment. For a large number of off-nominal failure conditions it must be demonstrated that they do not pose an undue risk during operation. Typically, the risk of a failure condition is given by its likelihood and consequence. The consequences are grouped into categories of “minor”, “major”, “hazardous”, and “catastrophic”, depending on the impact the failure has on the aircraft, crew, and passengers. For the successful certification of a safety-critical system, it must be shown that events with major, hazardous, or catastrophic consequences only occur extremely infrequently.

A complex system like an aircraft, which we are considering here, is comprised of numerous components, both realized in hardware as well as implemented in software. Faults in any of these components might contribute to a failure of the overall system. Fault Tree Analysis (FTA) (Vesely, 2002) is a well-established methodology to study the impact of faults within single or multiple components on the entire system. While FTA and associated probabilistic methods are mature for hardware components, their application to software and ML-enabled components poses fundamental challenges due to non-standard failure distributions, opaque and potentially adaptive decision boundaries, and the presence of rare but safety-critical ML-related failure events such as false negatives.

This paper addresses the statistical modeling and analysis of rare events in ML-based components within a system-level

safety framework. In this work, we are focusing on typical perception-related functional tasks that are found in unmanned Aerial Vehicles (UAVs) or that support aircraft operations. Such components usually use one or multiple Deep Neural Networks (DNNs) that have been trained and tested on given data sets prior to deployment. Examples include camera based obstacle detection on runways, detection and classification of runway signs at airports, automatic detection of runways for auto-landing, or vision-based taxiing.

Even when trained with suitable data, an ML-component might show unexpected or erroneous behavior during operation. Such an event might be triggered by a potentially unexpected environmental condition (e.g., fog or heavy snowfall) or distorted inputs due to camera failures (e.g., partial sensor failure). In addition, tiny changes in the input image might cause a substantially wrong result of the output of the ML component. Exposure to adversarial data might, for example, lead to a mis-classification of the relevant information, inability to detect the condition ("false negative") or raising a false alarm ("false positive"). That behavior is specific to components that use neural networks or other statistical ML methods. Such component-level events can potentially lead to system-level events with catastrophic consequences. A truck crossing the street, which was misclassified as bright sky (Yadron & Tynan, 2016) and leading to a fatal accident, is a typical example. Accurately characterizing and integrating such events into the system safety and risk analysis is essential but non-trivial.

Such events must only occur extremely infrequently to be able to ensure safety and risk requirements. These rare events also have specific statistical properties, which typical standardized statistical analyses might not be able to describe them properly or even handle them. For example, when assuming a Gaussian distribution, such events are typically at least 5σ distant from the mean. Describing such a distribution just by (μ, σ) would ignore rare events. However, when considered during a FTA, their probability, albeit low, might substantially contribute to the likelihood of an unwanted and catastrophic failure.

In this paper, we present SYSAI (He, 2015), a framework for the statistical analysis of complex AI-based systems on ML-component and system (e.g., aircraft) level. We leverage the capabilities of SYSAI and have extended the tool to be able to systematically model, generate, and analyze rare-event scenarios. Specifically, SYSAI is capable of identifying regions in the input space where rare events occur that can be associated with elevated risk. In particular, the efficient characterization of boundaries to safe regions can provide important information for the refinement of design requirements, definition of Operational Design Domains (ODD), and certification. SYSAI is also designed to generate synthetic data and scenarios targeting these rare regions, both in static and temporal contexts, supporting analysis over time series. SYSAI can statistically characterize these regions to establish probability distributions that can properly describe and model rare events.

Our approach bridges the gap between component-level rare event modeling and system-level risk estimation, incorporating key safety concepts such as the Operational Design Domain (ODD), Out-of-Distribution (OoD) detection, and Model Operational Context (MOC). By explicitly modeling rare events in the context of these safety constructs, we ensure a more rigorous and complete risk picture.

In this paper, we illustrate our approach using selected artifacts from three recent case studies: (1) ACT (autonomous centerline tracking) (He & Schumann, 2024, 2020), which uses a camera-based DNN to guide an autonomous aircraft down the runway, (2) AEBS (Autonomous Emergency Braking System) (Dmitriev et al., 2024) for the detection of airport no-entry signs and can issue an emergency braking if the pilot does not react, and (3) a visual landing guidance system (*Concepts of Design Assurance for Neural Networks (CoDANN)*, 2020). In each of the systems, images taken from a forward-facing camera on the plane are processed by a DNN-based component. The pre-trained DNN extracts important information about the current scenery, which is then fed into by a controller to elicit the required behavior: stay on the runway, stop at a no-entry airport sign if pilot is non-responsive, or facilitate safe auto-landing. Obviously, all these systems are safety-critical and are thus good candidates for the analysis of dramatic but infrequent events.

With SYSAI, we efficiently generate synthetic images and simulated scenarios and explore regions in the high-dimensional state- and failure-space that are likely to contain unwanted but rare events. Using SYSAI's customized statistical algorithms, their probabilistic characteristics can be properly described.

Our results demonstrate that modeling rare events using statistically sound techniques not only enhances the fidelity of safety analysis but also provides actionable insight for ML model and requirements refinement and system-level risk mitigation. We conclude with a discussion of open challenges and directions for future work, including real-time integration with verification pipelines, adaptive safety envelopes, and the formalization of rare event hierarchies in AI-enabled systems.

2. CASE STUDIES

In this paper, we will discuss our approach using two case studies from the aerospace domain: the Autonomous Emergency Braking system AEBS and the Autonomous Centerline tracking System ACT.

2.1. AEBS: Autonomous Emergency Brake System

The AEBS has been designed to serve as a case study for applicability of the DO-178C standard for low-criticality AI applications. The AEBS is an on-board system that, during taxi, can warn the pilot if a "no-entry" runway sign is visible and, in case the pilot does not react, issue an emergency braking. Its intended functionality makes it a safety-critical system of DAL-C. The AEBS system (Dmitriev et al., 2024, 2025) uses a pre-trained Deep neural network for camera-based detection of the runway sign. Figure 1 shows its architecture: a forward-facing camera on the aircraft provides images to the system. Two independent Deep neural networks (YoloV2 and F-RCNN) aim to detect if a standardized "no-entry" runway sign is limited. The Emergency Braking Controller (EBC) software then decides if a warning is to be issued to the pilot, or an emergency braking needs to be initiated to avoid entering forbidden regions in case the pilot fails to react.

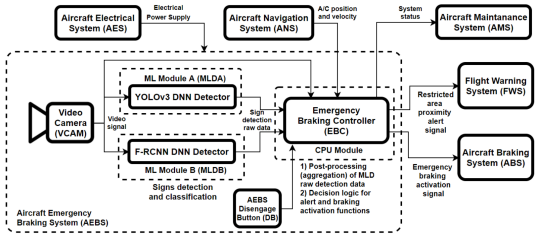

Figure 1. AEBS System architecture (from (Dmitriev et al., 2025))

A certification attempt of the system according to DO178-C, DAL-C is described in (Dmitriev et al., 2025). Of major importance for the certification is the estimation of failure rates for the network components. Only if the failure rates are low enough, the safety goals can be reached.

2.2. ACT: Autonomous Centerline Tracking

The Autonomous Centerline tracking system ACT is a demonstration system for the use of AI systems for autonomous aircraft operations. Its architecture is shown in Figure 2: a forward-facing camera on the wing of the aircraft provides images of the scenery in front of the aircraft to the ACT system. These images are processed individually by a deep neural network, which has been trained to estimate control inputs to a steering controller to taxi the aircraft at a low speed down a runway along its centerline. Obviously, the aircraft must not leave the runway, which makes the ACT system safety-critical. We are using the X-plane simulator[1] to simulate the aircraft and to generate the images for the neural network input. Safety analyses for ACT have been described in (He & Schumann, 2024).

[1] x-plane.com

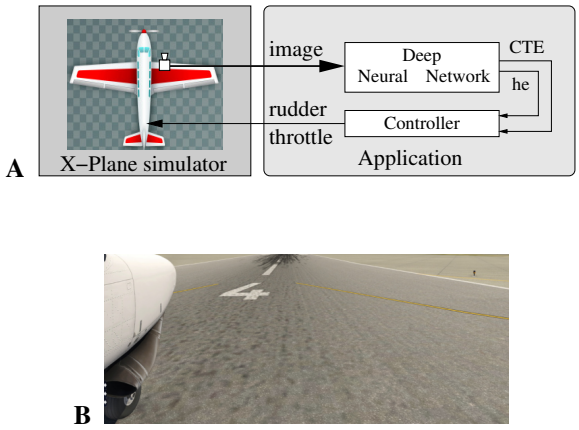

Figure 2. **A**: High-level architecture of the ACT system. **B**: Example image for input to the DNN

3. BACKGROUND: THE SYSAI FRAMEWORK

SYSAI (System Analysis using Statistical AI) (He & Schumann, 2020) is a flexible statistical learning framework for V&V and the analysis of complex and high-dimensional cyber-physical systems with AI components. Figure 3 shows the high-level architecture of SYSAI analysis framework. On the left-hand side, we have the "system under test" (SuT), which in our case is the ACT system and the X-Plane simulator, as described in the previous section. The SuT is executed given a set of parameters provided by the statistical learning model of SYSAI. The result of the test run is then used to incrementally construct the statistical model.

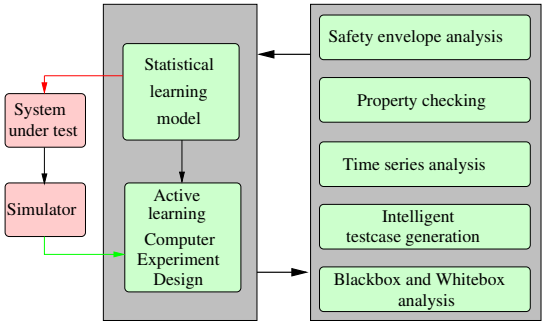

Figure 3. SYSAI architecture

For the representation and construction of the statistical model, SYSAI uses Dynamic Regression Trees (DynaTrees (Taddy, Gramacy, & Polson, 2011; Gramacy & Polson, 2011)), a dynamic Gaussian process model based upon Particle Filters. DynaTrees are regression and classification learning models with complicated response surfaces in on-line application settings. DynaTrees create a sequential tree model whose state changes over time with the accumulation of new data, and provide particle learning algorithms that allow for the efficient on-line posterior filtering of tree-states. A major advantage of DynaTrees is that they allow for the use of sim-

ple models within each partition. The models also facilitate a natural division in sequential particle-based inference: tree dynamics are defined through a few potential changes that are local to each newly arrived observation, while global uncertainty is captured by the ensemble of particles.

This surrogate model is initialized with available training data and incrementally refined using candidate data points that are produced by our active learning module. It evaluates the current surrogate model using a customized active-learning heuristics and suggests candidate data points that provide most information for model refinement. For these candidate points, the ground truth is obtained by executing the SuT.

SYSAI features customizable heuristics that allow the active learning to focus on particular characteristics of the model. Classical algorithms like ALM (MacKay, 1992) or ALC (Cohn, 1996) focus on under-explored regions in general of the domain space. Inspired by (Jones, Schonlau, & Welch, 1998) and work on contour finding algorithms, we loosely follow (Ranjan, Bingham, & Michailidis, 2008) and define our boundary-aware metric boundary-EI (He, 2015, 2012) that puts the focus of the search into "interesting" and potentially "troublesome" areas near safety boundaries. Here, the surrogate model therefore exhibits substantially more details than in other areas that are not of interest. This exploration is guided by the selected active learning heuristics and is able to cover the entire input space with a low number of data points. The SYSAI framework and the underlying models and algorithms are described in detail in (He & Schumann, 2020).

SYSAI, in general, supports the following important analysis tasks (Figure 3):

- *statistical analysis of training and test data:* for a detailed statistical analysis of the data sets used for training and evaluation of the DNN.
- *Safety-envelope analysis:* our framework can perform automatic analysis of the safety-envelope, which indicates under which operational conditions the system is behaving safely or not. Geometric shape modeling does not only identify but also characterizes regions with similar behavior and describes those regions in easy to understand geometrical terms. SYSAI thus helps to make the system more explainable.
- *Property checking:* our tool supports the automatic checking and analysis of safety and performance requirements.
- *Time series analysis:* SYSAI can perform advanced time-series analysis in a high-dimensional parameter and state space. This analysis provides a deeper understanding of the system behavior and its dynamics. The tool also supports event prediction.
- *Intelligent test-case and scenario generation:* SYSAI can efficiently generate relevant test cases in high-dimensional spaces.
- *Blackbox and Whitebox analysis:* with our SYSAI tool, system-wide and component-based analyses can be carried out. The SYSAI interface allows to access data inside of components, thus enabling Whitebox analysis.

4. MODELING AND ANALYSIS OF ISOLATED RARE EVENTS

Safety-critical systems like AEBS or ACT must ensure that severe or catastrophic failures only occur with extremely low probability p. In aerospace applications, acceptable failure probabilities are often below 10^{-5} and can reach levels as low as 10^{-9}. Such *rare events* are defined as events with $p < 10^{-5}$ in the general statistical literature (Rubino & Tuffin, 2009; Bucklew, 2004).

4.1. Statistical Challenges

Estimating the probability of rare events poses unique challenges. Empirical data often contain no direct observations of failure, rendering naive frequency estimates unreliable. Standard Monte Carlo approaches are computationally infeasible, as the number of required samples grows inversely with the event probability. Specialized techniques—such as importance sampling, subset simulation, and extreme value theory—are therefore essential for efficient estimation of tail probabilities (Au & Beck, 2001; Juneja & Shahabuddin, 2006).

4.2. Failure Modes in AI-Based Components

While failures of hardware components are well-characterized and understood, failures of (traditional) software and AI/ML-based components introduce qualitatively different uncertainty sources:

- **Hardware:** failures are typically physical, intermittent, or continuous, and often modeled via exponential or Weibull distributions. Typically these probabilities are expressed in Mean Time between Failures (MTBF) or Mean Time To Failure (MTTF).
- **Conventional software:** is treated as error-free once verified and considered to be deterministic, so no probabilistic failure model is defined.
- **AI/ML software:** AI/ML components can react in a probabilistic way, even if the underlying execution algorithms (e.g., DNN evaluation) are deterministic and does contain any random number generator. Probabilistic behavior comes in through the training regime and data. So, even a correctly implemented software may still yield erroneous or uncertain outputs.

In the AI/ML systems that we are considering here, failure

events correspond to incorrect or unsafe outputs. These can be incorrect numerical values for DNN outputs cte or he in the ACT system, or a misclassification, erroneous detection (false positive), or non-detection (false negative) of the No-entry sign in the AEBS. The likelihood of such failure events depends on multiple interacting factors: (a) model and network architecture, (b) data quality and coverage, (c) training regime, and (d) operational input distribution. Consequently, failure behavior is typically non-stationary and data-dependent, often exhibiting sharp probability gradients or adversarial sensitivities (Goodfellow, Shlens, & Szegedy, 2014; Amodei et al., 2016).

4.3. Rare Event Modeling for AI Components

Quantifying rare failures in AI/ML systems remains is a challenging research problem. Since expected error probabilities may be extremely small, rare event simulation and statistical reliability methods are required to characterize the low-probability tails of model behavior. Promising directions include adaptive importance sampling (de Koning et al., 2005) for learned decision boundaries, extreme value modeling of prediction confidence, and synthetic data generation for stress testing. Integrating such approaches into certification-oriented reliability analysis is key for the safe deployment of AI in safety-critical domains.

With SYSAI, we can generate test cases and scenarios that focus on a detailed exploration of high-dimensional decision boundaries. Scenarios are generated under various environmental conditions, for example, different weather situations or time-of-the-day, or failure conditions, e.g., camera failures.

SYSAI supports the analysis on system level as well as on component level. For example, the entire AEBS can be evaluated for regions, where a safe stop is possible (safety region). On the component level, on the other hand, SYSAI only evaluates the AI/ML component itself. In our case study this would correspond to the DNN receiving a generated image, and its estimated output values or the correctness of the sign recognition. For the ACT case study, we evaluated the performance of two DNNs with different architecture. Figure 4 shows the distributions for estimated CTE values for different ranges of CTE. The red line in Figure 4 indicates the ideal performance. For negative values of CTE, DNN A shows a larger bias than DNN B. However, for positive CTE values, DNN B shows a substantial bias, which could render this DNN unfit for the purpose. However, system level experiments with SYSAI revealed that for nominal operations, both DNNs were capable of steering the AC along the runway, most likely due to the controller robustness. This also indicates that for failure and risk analysis both, system-level and component level analysis must be performed.

Figure 5 shows, how the probability of the detection rate for a no-entry sign varies over the distance between the camera on

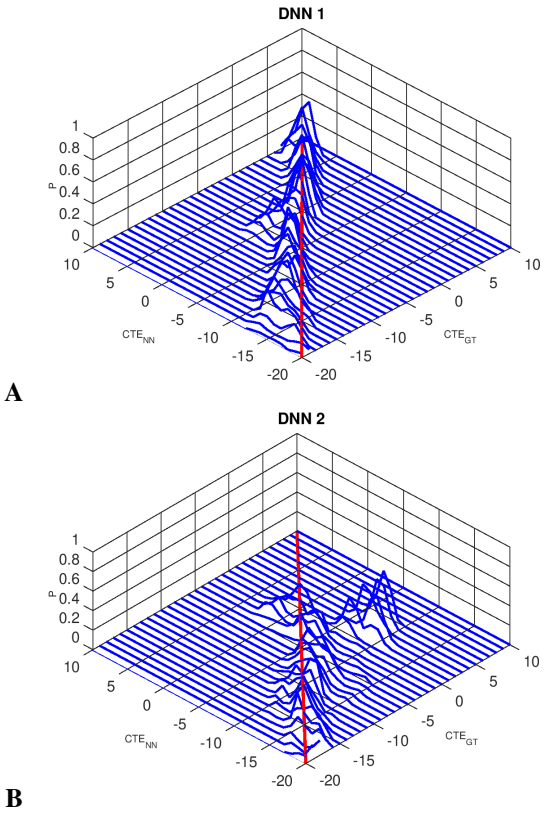

Figure 4. Distribution of CTE outputs over ground truth for two different DNN architectures used for ACT

the aircraft and the sign. For larger distances, the recognition-rate is low, because the sign appears very small on the image, which makes it detection difficult. Getting closer to the sign, there exists a region, where the sign can be detected reasonably reliably. Note that, however, the detection probability is still considerably lower at around 90-95% than needed for a safety-critical application (e.g., 0.999). When getting even closer to the sign, the detection reliability again drops substantially. Causes might include lateral offsets of parts of the sign moving outside the frame.

5. NON-INDEPENDENCE OF EVENTS

The previous section discussed isolated rare events occurring independently and infrequently. While such models are often sufficient for hardware components, they are inadequate for AI and ML-based components. Here, we assume that the software itself does not fail in an implementation sense, but that it may produce incorrect or uncertain outputs. This distinction has major implications for reliability analysis, as AI-induced errors are often more frequent, structured, and temporally correlated than classical random failures.

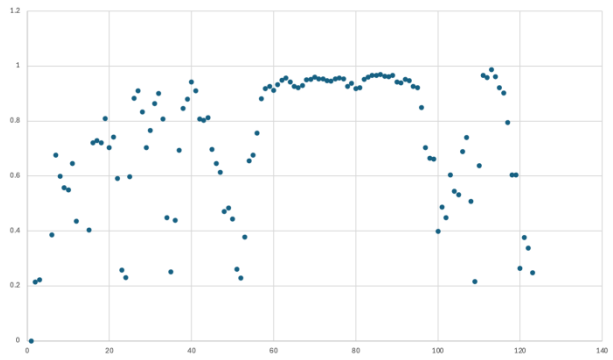

Figure 5. Typical success probability over distances from camera to sign. Based on actual images (Dmitriev et al., 2025)

5.1. Dependence and Temporal Correlation in AI Failures

AI components can exhibit error rates that are several orders of magnitude higher than what is tolerable in safety-critical systems. For example, state-of-the-art visual object detection models commonly report mis-classification rates of a few percent, even under nominal conditions (Redmon & Farhadi, 2018; Janai, Güney, Behl, & Geiger, 2020). As shown in Figure 5, the trained DNN in the AEBS system achieves a detection rate (true positive rate) of around 95%.

A possible way to decrease the failure probability is to base the decision – in AEBS the decision to perform an emergency brake action – not on a single processed image, but on a number of images. AEBS defines a detection window of length l and requires a minimal number of correct answers a during this interval (Dmitriev et al., 2025). There $l = 12$ and $a = 5$.

Other metrics for decision-making can be defined, like a minimal number of consecutive detections, or filter-based approaches that can use the detection confidence for each image. For each of those approaches, maximum failure probabilities under various nominal and off-nominal conditions must be established as a basis for subsequent FTA. This, however, is challenging, as failure distributions of the AI/ML components differ substantially from the exponential models typically used for hardware reliability (O'Connor & Kleyner, 2012). A more critical issue, however, is that consecutive errors are often statistically dependent—violating the independence assumption underlying many conventional reliability and safety models.

Such dependencies can arise from multiple sources:

- **Correlated inputs:** Successive sensor frames or environmental measurements are not independent. Temporal or spatial correlation in the sequence of images can induce correlation in the component's outputs.
- **Contextual or environmental factors:** Adverse conditions such as rain, glare, fog, or occlusions can cause bursts of failures—periods of elevated error probability—rather than isolated events (Hendrycks et al., 2021; Geirhos et al., 2020).

  A typical example in the realm of AEBS could be a truck, crossing in front of the do-not-enter sign, which makes the sign undetectable for a number of consecutive images. Similarly, the sun rising or setting in the vicinity of the sign could generate glare and overexposed images that makes it impossible to properly detect the sign. Those failure situations are not permanent, but could lead to malfunction of the AEBS system if not considered. For the FTA analysis, the probabilities for such failure occurrences need to be taken into account.
- **Systemic biases:** Dataset biases or over-fitting to specific environmental conditions can create structured failure patterns that persist over time (Oakden-Rayner, Dunnmon, Carneiro, & Re, 2020).

  A typical example would be a bias for the DNN output as shown in Figure 4. Here, a large bias for larger cte values in DNN2, but also a small bias for negative cte with DNN1 could potentially lead the AC off the runway. Also, SYSAI analysis detected low ACT performance after 2PM (He & Schumann, 2020), which was caused by insufficient training data coverage.

These failures can be transient *Short-term correlated failures* or *Persistent failures*. Such long-term conditions such as a dirty lens, camera icing, or sensor misalignment can cause continuous undetected states, representing a shift to a high-failure regime.

These temporal dependencies imply that rare-event models assuming identically distributed and independent failures (i.i.d.) will underestimate the overall risk. The effective failure probability over a decision window—such as the braking decision interval—must account for temporal correlation and burstiness of the underlying error process (Leveson, 2012; Kalra & Paddock, 2016).

5.2. Modeling Implications and Statistical Considerations

Accurately characterizing interdependent failures requires modeling approaches that extend beyond scalar failure probabilities. Obviously, calculation of autocorrelation between the different images of the sequence to be processed, is important. Together with a model of the decision-making, this can be used to estimate the overall failure probability.

The autocorrelation between the individual images makes it possible to quantify temporal dependencies. Hidden Markov models, switching models, or techniques like Kalman filters are taking in information from the actual design of the system,

in this case, the EBC (Figure 1) into account and can improve the accuracy of the analysis.

The analysis of image streams with a higher rate than needed can provide useful information about the interdependencies between the individual frames and can thus be used to select a suitable update rate of the AI/ML system. With SYSAI, we efficiently can generate scenarios in nominal and off-nominal cases and under different environmental conditions, like time-of-the-day, weather, or sign placement. The resulting image sequences are then analyzed.

Figure 6 shows the autocorrelation of images over the lag for images within an AEBS scenario under different environmental conditions. Obviously there is a high correlation between subsequent images, which quickly decays for an increasing lag. However, minimal lags to reach a 95% confidence interval are different, as well as the negative correlations for larger lags. This demonstrates that different environmental conditions need to be considered for establishing minimal lags where independence can be assumed with confidence. In a similar way, SYSAI can generate scenarios which include off-nominal rare events, as discussed above.

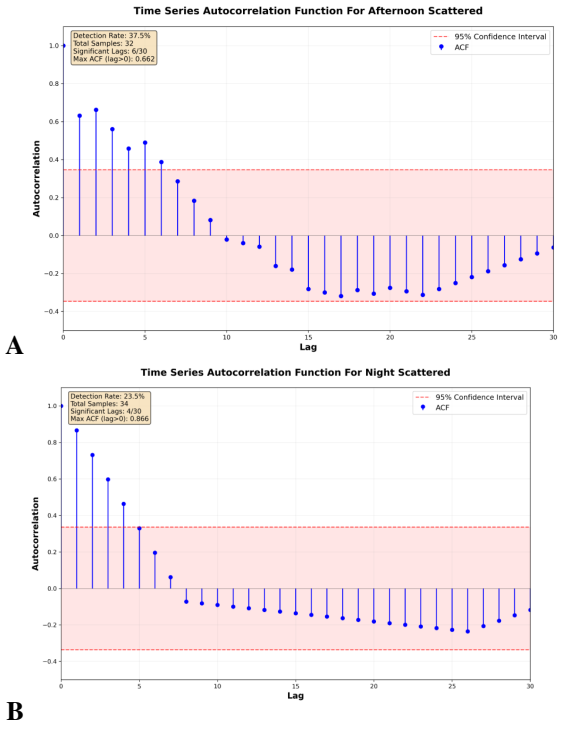

Figure 6. Autocorrelation of images over lag under day-light with scattered clouds (A) and night-time conditions (B)

6. CONCLUSIONS

In this paper, we used two case studies, an Autonomous Emergency Braking System (AEBS) and an Autonomous Centerline Tracking (ACT) system to demonstrate how SYSAI models both isolated and temporally correlated failures. It shows that AI/ML components can exhibit non-independent, "bursty" errors caused by environmental conditions, transient failures, and dataset biases. SYSAI's time-series and boundary-aware analyses capture these dependencies, revealing impacts on system-level risk that would be missed by independence assumptions. This illustrates that statistically rigorous rare-event modeling can provide detailed insights and model refinement for risk analysis and can also improve safety envelopes and failure-rate estimation for aerospace systems with AI/ML components.

REFERENCES

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*. Retrieved from https://arxiv.org/abs/1606.06565

Au, S.-K., & Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. In *Probabilistic engineering mechanics* (Vol. 16, pp. 263–277). doi: 10.1016/S0266-8920(01)00019-4

Bucklew, J. A. (2004). *Introduction to rare event simulation*. Springer. doi: 10.1007/b97424

Cohn, D. A. (1996). Neural network exploration using optimal experimental design. *Advances in Neural Information Processing Systems*, *6*(9), 679–686.

Concepts of design assurance for neural networks (CoDANN) (Tech. Rep.). (2020). European Aviation Safety Agency.

de Koning, M., Cai, W., Sadigh, B., Oppelstrup, T., Kalos, M., & Bulatov, V. (2005, 03). Adaptive importance sampling monte carlo simulation of rare transition events. *The Journal of chemical physics*, *122*, 074103. doi: 10.1063/1.1844352

Dmitriev, K., Rhein, J., Beller, L., Bröcker, J., Huber, E., Schumann, J., & Holzapfel, F. (2024). Safety assessment of a machine learning-based aircraft emergency braking system: A case study. In *2024 aiaa datc/ieee 43rd digital avionics systems conference (dasc)* (p. 1-10). doi: 10.1109/DASC62030.2024.10749696

Dmitriev, K., Rhein, J., Bender, J., Beller, L., He, Y., Schumann, J., & Holzapfel, F. (2025). Certifying machine learning in aviation: An end-to-end dal c case study. In *2025 aiaa datc/ieee 44rd digital avionics systems conference (dasc)*.

Do-178c: Software considerations in airborne systems and equipment certification (No. RTCA DO-178C). (2011). Washington DC: RTCA, Inc.

EASA concept paper: First usable guidance for level 1&2 machine learning applications (Tech. Rep.). (2023). European Aviation Safety Agency.

EASA concept paper: First usable guidance for level 1 machine learning applications (Tech. Rep.). (2021). Eu-

ropean Aviation Safety Agency.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, *2*, 665–673. doi: 10.1038/s42256-020-00257-z

Goodfellow, I., Shlens, J., & Szegedy, C. (2014, 12). Explaining and harnessing adversarial examples..

Gramacy, R., & Polson, N. (2011). Particle learning of Gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, *20*(1), 467–478.

He, Y. (2012). *Variable-length functional output prediction and boundary detection for an adaptive flight control simulator* (Unpublished doctoral dissertation). University of California at Santa Cruz.

He, Y. (2015). Online detection and modeling of safety boundaries for aerospace applications using active learning and bayesian statistics. In *2015 international joint conference on neural networks, IJCNN 2015, killarney, ireland, july 12-17, 2015* (pp. 1–8). IEEE. Retrieved from https://doi.org/10.1109/IJCNN.2015.7280595 doi: 10.1109/IJCNN.2015.7280595

He, Y., & Schumann, J. (2020). A framework for the analysis of deep neural networks in aerospace applications using bayesian statistics. In *Proc. ijcnn, wcci.*

He, Y., & Schumann, J. (2024). Statistical analysis and runtime monitor- ing for an ai-based autonomous centerline tracking system. *IJPHM*, *15*(3).

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., ... Gilmer, J. (2021). The many faces of robustness: A critical analysis of out-of-distribution generalization.. Retrieved from https://arxiv.org/abs/2006.16241

Janai, J., Güney, F., Behl, A., & Geiger, A. (2020). Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends in Computer Graphics and Vision*, *12*(1–3), 1–308. doi: 10.1561/0600000079

Jones, D., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, *13*, 455–492.

Juneja, S., & Shahabuddin, P. (2006). Rare-event simulation techniques: An introduction and recent advances. In S. Henderson & B. L. Nelson (Eds.), *Handbooks in operations research and management science: Simulation* (Vol. 13, pp. 291–350). Elsevier. doi: 10.1016/S0927-0507(06)13009-3

Kalra, N., & Paddock, S. (2016). *Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?* (Tech. Rep.). RAND Corporation. doi: 10.7249/RR1478

Leveson, N. G. (2012). *Engineering a safer world: Systems thinking applied to safety*. MIT Press.

MacKay, D. J. C. (1992). Information–based objective functions for active data selection. *Neural Computation*, *4*(4), 589–603.

Oakden-Rayner, L., Dunnmon, J., Carneiro, G., & Re, C. (2020). Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL)*, 151–159. doi: 10.1145/3368555.3384468

O'Connor, P. D. T., & Kleyner, A. (2012). *Practical reliability engineering* (5th ed.). Wiley.

Ranjan, P., Bingham, D., & Michailidis, G. (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, *50*(4), 527–541.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. Retrieved from https://arxiv.org/abs/1804.02767

Rubino, G., & Tuffin, B. (2009). *Rare event simulation using monte carlo methods*. John Wiley & Sons. doi: 10.1002/9780470745403

Taddy, M. A., Gramacy, R. B., & Polson, N. G. (2011). Dynamic trees for learning and design. *Journal of the American Statistical Association*, *106*(493), 109-123.

Vesely, W. (2002). *Fault tree handbook with aerospace applications*. Washington, D. C.: NASA Office of Safety and Mission Assurance.

Yadron, D., & Tynan, D. (2016). *Tesla driver dies in first fatal crash while using autopilot mode.* The Guardian. Retrieved from https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk