Runtime Verification and AI: Addressing Pragmatic Regulatory Challenges

Christian Colombo[1][0000−0002−2844−5728], Gordon Pace[1][0000−0003−0743−6272], and Dylan Seychell[2][0000−0002−2377−9833]

[1] Department of Computer Science, University of Malta, Malta
[2] Department of Artificial Intelligence, University of Malta, Malta
{christian.colombo,gordon.pace,dylan.seychell}@um.edu.mt

Abstract. The deployment of AI-driven solutions to increasingly complex tasks with real-world impact raises various challenges in the area of verification. Using the case study of an AI-assisted litter detection being developed for rural areas in Malta, this paper highlights the multi-faceted nature of the risks involved concerning: data issues, functionality correctness, safety concerns, and legal considerations. We place particular focus on the last of these: regulatory challenges.
Drawing inspiration from related works, considering applicable Maltese technology guidelines and EU legislation, against the backdrop of the challenges presented in the case study, the proposed runtime verification architecture brings the pieces together in a comprehensive and pragmatic manner.

Keywords: Runtime Verification · Artificial Intelligence · Regulatory compliance.

1 Introduction

The histories of formal verification techniques and artificial intelligence show a surprising degree of confluence — from Turing's first tentative steps in both areas[3] to the highs of the breakthroughs separated by frugal winters. And although one cannot truly say that now, in the 2020s, verification has become ubiquitous in software design and development to the same extent that the application of AI has become pervasive, one cannot but note that the importance of software functional correctness has increasingly been recognised as a requirement both in economic terms (i.e. measured in financial return, or as mitigation against potential loss) but also from a regulatory perspective, especially for systems used in critical settings.

In this paper, we look at opportunities for formal verification given the current regulatory landscape, particularly those evolving around the use of AI. We argue, that the regulatory recognition of the need for functional correctness for

[3] Turing's 1949 paper on algorithmic verification [32] and his 1950 paper on machines and intelligence [33] are frequently cited as the starting points of the two fields.

AI solutions (particularly high-risk ones) can prove to be an opportunity for the integration of formal verification techniques as part of these compliance processes.

From a regulatory perspective, we focus on the European Union's AI Act [8] and the related regulatory guidelines set up in Malta [20] — as two early models of regulatory requirements addressing AI. In order to illustrate the adoption of formal verification techniques, we consider a real-world AI application currently under development for more efficient litter detection in varying terrain through the use of drones [30, 28].

The application of verification techniques to AI is relatively much younger than either of the two fields. One major challenge encountered in the combination is that much of formal verification is intimately tied to the control structure and flow of the system-under-scrutiny. In human designed systems, such a flow contains much of the abstraction present in the system architects', designers' and developers' mind ensuring correctness (or at least making the system more likely to be correct). In contrast, the unstructured artefacts generated through the more widespread machine learning and statistical techniques (and which have been shown to be most effective over the past years) fail to provide such a handle for verification. In contrast to most verification techniques, runtime verification excels at formal verification of black-box artefacts, acting more on their outputs and less on their internals. This has been recognised in recent work combining runtime verification and AI. As we will argue, based primarily on regulatory requirements (but also pragmatic ones), the use of runtime monitoring is a perfect marriage with compliance and AI.

The paper is organised as follows: we present background and related work in the next section, followed by an overview of the case study on AI-assisted litter detection in Section 3. In Section 4, we review the regulatory landscape surrounding AI in Malta and the EU. Next, we present our proposal in Section 5 which brings together the three strands of this paper: RV, AI, and legislative concerns. Finally, Section 6 concludes the paper.

2 Background and Related Word

The study of enhancing the reliability of complex systems through runtime assurance is far from new e.g. the Simplex architecture proposed in 2001 allows a simpler but safer module to replace a more complex badly-behaving one at runtime [31]. Typical AI systems used for sensitive and non-trivial tasks fit the description of a complex system, while RV explores a wide range of techniques to provide on-the-fly detection and reaction to unexpected behaviour. In this sense, the relationship between RV and AI has already been studied for some time. However, there are aspects of AI systems which can be specifically explored from the RV point of view, depending on whether verification takes a black or white box approach.

Naturally, apart from using RV to monitor AI, RV can also benefit from AI techniques in its quest for pattern recognition (e.g. using anomaly detection

techniques to monitor system execution [15]). Furthermore, correspondence in approach between the two areas has also been highlighted from the point of view of rule-based production systems [11]. However, this paper doesn't focus on the combination of the two fields from this direction.

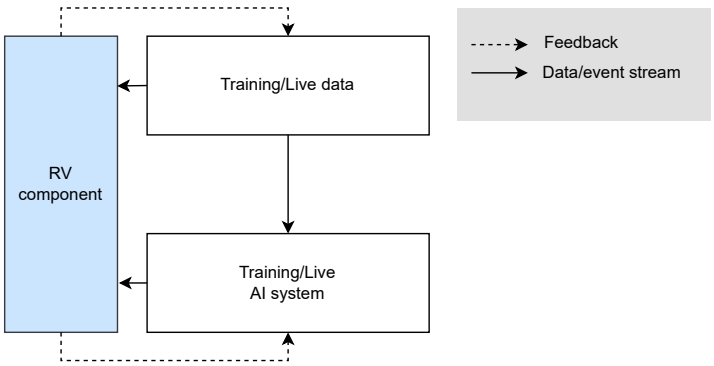

Fig. 1. Architecture of state of the art RV of AI systems.

RV to Monitor AI To catalogue the various ways in which RV has been used to monitor AI, we loosely depict the basic components of AI in Fig. 1 where training data is fed into the AI components during the training phase, involving a feedback loop for learning. Following the training phase, the system receives live data and outputs its verdict. This whole process is usually overseen by the human expert in the field who selects the data, sets the algorithm's parameters, and validates the output of the AI component, particularly during the training phase.

There are several ways in which RV has been employed within this picture (see [29, 14] for comprehensive surveys). Perhaps the most obvious but restricted way to employ RV is to focus on the AI output, i.e., treat the whole process as a black box and check whether the output is within the expected bounds [17]. Also known as *shielding*, this is particularly useful when AI is used within critical applications where wrong output can have serious consequences. Apart from suppressing wrong AI output, shields may also give rewards or punishments to the AI component to learn from the past [1].

Moving a little away from the black box approach, some RV approaches take the training data into consideration when deciding whether the output can be trusted or not. If the live data is significantly different from any of the data the AI was trained on, the RV component can flag the output as dubious [4] or switch to a fallback component [12]. A related approach, involves the human expert ("authority") in the RV process to handle previously unseen input classes [16].

Taking a white box approach, other RV approaches e.g. [4, 13] look directly at the internals of the AI component by also monitoring the hidden layers of the neural network for *outside the box* behaviour.

RV of Legal Concerns The study of legally-binding texts within the context of specification and verification is not new [25], with the focus recently also looking into smart contracts [26]. While the approaches have much in common with typical runtime verification, contracts have their own particularities especially because of the multiple parties usually involved, and due to the complexity of the deontic operators often necessitating conflict analysis. Smart contracts, on the other hand, come with their own set of challenges, particularly because of their immutability and the costs associated with execution (gas) which monitoring contributes to.

The adoption of RV to monitor for adherence to normative texts, has been applied in a wide range of areas: legislation of financial services [3], privacy policies within social networks [27], and airport passenger rights [9]. More recently, the area of normative systems has also been combined with RV and AI by synthesising supervisory modules which enforce "ethical norms" on reinforcement learning agents at runtime [24].

3 Case Study: AI-Assisted Litter Detection

Manual litter collection can be a costly process involving a number of steps: litter needs to be visually located (possibly from some distance) by having personnel searching for it within an area under consideration; after which, the litter needs to be precisely located (possibly requiring a vehicle), picked up, and sorted according to type (paper, plastic, glass, etc).

From such a traditional perspective that assumes no use of technology, efforts in litter collection are typically focused and optimised for urban contexts for a number of practical reasons emanating from the fact that rural areas are typically unstructured, variable, and spread over a relatively large area: (i) visual detection is easier when litter lies on a uniform surface, such as roads or pavements; (ii) infrastructure typically found in an urban context limits the areas where litter might be, thus providing a smaller search space; (iii) litter collection is harder when access to particular areas is limited, particularly for vehicles.

Novel and recent developments in AI provide an opportunity to address the challenges described above to carry out sustainable, efficient, and large-scale litter collection in rural areas. Aerial images captured from different altitudes can be used to automate the search and geolocation process for litter in a vast area. Subsequently, the unmanned aerial vehicle (UAV) can also be used to collect and sort the litter by type.

While our vision for a fully automated process of litter collection is not yet a reality, previous works [30, 28] have demonstrated how parts of the process can be automated, namely the litter searching and detection mechanism from images captured by a human-piloted UAV. As we continue working towards the

full automation of the litter collection process, we are aware that this brings along various technical, safety, and regulatory risks and challenges.

Data risk: Object detection accuracy. The AI model's effectiveness may be limited due to misclassification errors, particularly when analysing images captured from higher altitudes. Accuracy in prediction over multiple classes can be a challenge. This is split into two aspects: the first being errors in classification between classes of litter, and the second is the precision and recall when detecting litter. The performance is directly related to the training and evaluation dataset. It follows that the more diverse and well-annotated a dataset is, the better the performance of the object detector and mitigation of this risk.

Functionality risk: Limited accuracy of GPS prediction. To locate litter via drone observations, the coordinates of detected litter need to be approximated as a relationship between the drone's GPS coordinate, including its altitude, and the object's location in the acquired image. This prediction is subject to precision errors, potentially affecting the efficacy of litter detection and collection. Alternative or complementary geolocation techniques should be investigated to optimise precision and mitigate this risk. Rigorous field trials should determine the optimal drone altitude for maximising GPS accuracy in litter identification.

Regulatory risk: UAV regulations. Deployment of AI-assisted drone-based litter detection systems may be hindered by evolving regulatory landscapes[4]. Additional limitations imposed on drone operations in the 'open' category could significantly impact what is permissible for manually and automatically operated drones. Currently, drones in Malta are mainly restricted by their geolocation, as shown in red in Fig. 2. The two largest restriction areas are the airport and its landing zones, together with the training grounds of the Armed Forces of Malta. There are also other specific sensitive areas, such as governmental executive offices and presidential grounds.

Regulatory risk: Privacy concerns. Real-time image or video analysis by the onboard AI model might capture identifiable data of individuals or vehicles within the drone's field of view. The system should incorporate runtime verification safeguards to mitigate this risk and ensure compliance with regulations such as GDPR. This may be due to storing of data for logging or offline analysis reasons, or to additional functionality that one may eventually add to the basic requirements of a litter-collection system. These safeguards may involve real-time anonymisation techniques, such as blurring or masking detected people or vehicles with solid colour boxes.

Safety risk: Damages to third parties. Drone operations require favourable weather conditions and cannot be conducted on rainy days or days with winds exceeding operational limits. For drones operating in the open category (maximum of 25kg) the wind resistance of the drone is around 28 km/h

[4] In the European Union, the provisions applicable to drone operations in the 'open' and 'specific' categories are described in EU Regulation 2019/945 and EU Regulation 2019/947.

(16 knots). Moreover, (EU)2019/947 stipulates that the drone operator has to provide a statement confirming they comply with the respective national insurance requirements for liability. This risk also includes the guidance by the system of human litter pickers who might be directed to approach dangerous terrain such as proximity to the edge of cliffs.

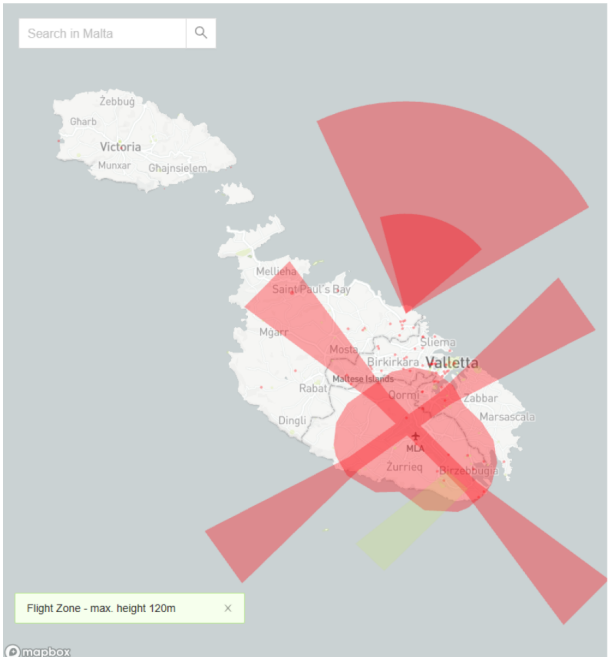

Fig. 2. Geographical Zones across the Maltese Islands. The zones indicate areas where drone operations are restricted due to general/commercial aviation. (Source: Drones Geographical Zones — Transport Malta (gov.mt) https://tmcad.idronect.com/map)

4 The Regulatory Landscape

Over the past years, we have seen an increased awareness of the need for having regulation address technological risks leading to potential operational compliance failure. Whereas before, such risks were typically tied to particular service or production industries (e.g. medical devices, aviation, automobile-industry, financial transactions, privacy), we are starting to see technology-specific regulation being enacted both at a national and an international level, such as the *Innovative Technology Arrangement Act* set up in Malta [18, 5], which addressed distributed ledger technologies and later widened to cover the whole class of critical systems, the EU Cybersecurity Act [7] and more recently, the EU AI Act [8].

The approach taken by such technology-centric legislation has typically been a risk-based and risk-proportionate approach, identifying the degree of technological due diligence required to minimise these risks (both in terms of the actual technology but also of the surrounding operations supporting the technology, such as incident response).

The EU AI Act takes a wide definition of AI: *"machine-based system designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment and that, for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments"* [8]. The focus on the act is that of *trustworthy* AI, shaped to regulate based on the risk level of the AI use: *prohibited artificial intelligence practices* as identified in the Act[5]; *high-risk AI systems* and their operators regulated under specific stringent requirements and obligations; and *certain AI systems* being subject to rules covering transparency, market placement, monitoring, surveillance and enforcement.

Obligations include ones with a data-focus, particularly addressing risk of under-representation of classes of persons and resulting bias, ones with a focus on functional correctness (particularly for safety components), and ones addressing the need for legal and regulatory compliance.

The Act also requires the setting up of national AI regulatory sandboxes, in which systems may be deployed through *"the development of tools and infrastructure for testing, benchmarking, assessing and explaining dimensions of AI systems relevant for regulatory learning, such as accuracy, robustness and cybersecurity as well as measures to mitigate risks to fundamental rights, environment and the society at large"* [8].[6]

Concrete control objectives and measures to be used in conformity assessments and audits in relation to the EU AI Act, are still to be designed and published. In order to enable us to concretely discuss and propose the use of formal methods in such a regulatory environment, we will be focussing and illustrating our proposed approach with reference to the AI regulatory guidelines set up in Malta in 2019 [20], where concrete control objectives were proposed [22] and a technical assurance sandbox was also launched [23].

The approach is much aligned with that of the EU AI Act, focusing on the trustworthiness of AI solutions through regulatory processes and obligations and technical audits or assessments against Control Objectives defined in the official

[5] These include systems using subliminal techniques, ones which distort the behaviour of persons on the basis of their age, disability or a specific social or economic situation, use of biometric data to categorise individuals, inferring personal information such as race, political opinions, etc., some uses of real-time remote biometric identification systems in publicly accessible spaces and AI systems to infer emotions of a natural person in the areas of workplace.

[6] It is worth highlighting that this is a *regulatory*, not *technological* sandbox i.e. it is meant to address and mitigate regulatory risks through the residency period in the sandbox, rather than being a technical solution to limit the interaction of a system with its environment.

guidelines covering different facets, including: (i) auditing of design and development processes; (ii) appropriate measures in place to address risks such as cybersecurity and the handling of personal data; (iii) design components intended to support assessment of live systems, including the *forensic node* and *system harness* (more on these later); (iv) measures mitigating data and bias risk including ones potentially arising post deployment; (v) functional correctness; and (vi) regulatory compliance.

Assessment for functional correctness (which is particularly interesting in the context of this paper) is addressed through requirements for: (i) the provision of a well-defined blueprint documenting the system in question; (ii) an English language description for consumer protection reasons (which is considered to prevail over what the system actually does). Both approaches use informal requirements, against which audits and assessments are performed, but pragmatically, from a regulatory perspective, it is crucial that obligations and requirements are not over-onerous, and, given that even from a theoretical perspective, there will always be a *documented-specification* vs. *actual-requirements* gap, the question is where to place the bar of specification writing to balance reasonable obligations against increased-trustworthiness returns [10].

A more thorough overview of the approach adopted by Malta can be found in [6], but we will hereby highlight three aspects which are very relevant to the rest of this paper:

System harness: The guidelines make it a requirement that an AI system should include an *ITA Harness*[7] [21]. The harness must surround the underlying system (see Fig. 3), thus providing a safety net (i) enabling the monitoring environmental interaction to ensure compliance with the documented expected behaviour; (ii) enabling identification of anomalies it detects e.g. out-of-bounds inputs and outputs. Note that the harness encompasses, not only the core system (the dotted box and the bottom box), but also the data collection and processing engines (the top two boxes), as well as the training process (if applicable). This enables its use, not only to enable assessment of the system, but also to collect information and detect anomalies elsewhere e.g. detecting potential bias during the data collection phase. It is worth adding that the Control Objectives require the technical auditors to review that the implementation of the harness is as specified in the regulatory guidelines i.e. collects all relevant information and events faithfully.

Forensic node: The Forensic Node is intended to be a data repository to store all relevant events and information during the runtime of the AI-ITA in real-time, and in a secure tamper-proof manner. In conjunction with the system harness, this enables offline assessment of the system's operating effectiveness of the controls during an audit and would support legal compliance investigations. Although the official guidelines architectural diagram (Fig. 3) do

[7] *Innovative Technology Arrangement* (ITA) is the general term for the digital systems generally covered by the legislation, of which AI systems applied to critical areas pertain.

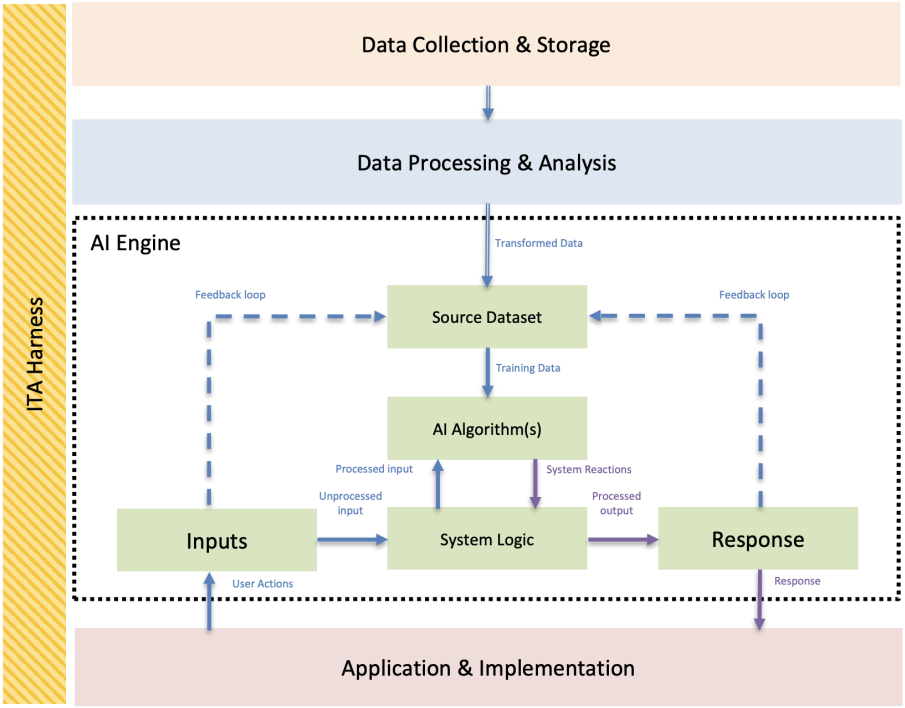

Fig. 3. ITA Harness around an AI system (taken from [21]).

not show the forensic node, it is mentioned that it would typically feed off the system harness or be an integral part of it.

Data-focused requirements: The guidelines place requirements on the documentation (as part of the system blueprint and the terms-of-service, and hence taken to be part of the specification to be audited) to report risks and mitigation measures on addressing bias, anonymisation and explainability [19]. This links in with the harness (which is also monitoring the data collection) to enable active assessment and reporting of such risks.

5 Runtime Verification and AI: A Proposal

In order to address the challenges in achieving trustworthy AI in a regulated setting, we are looking into combining different types and levels of monitors and runtime verifiers to deal with different risks arising from the deployment of such AI solutions. We will be using examples from the UAV system outlined in Section 3 in order to illustrate how different types of requirements and obligations will be mapped onto the proposed monitoring framework. We identify four roles of runtime verification in such a setting, as discussed in the sections below.

5.1 Monitoring Data Flow

Typical systems which verification has been applied to in the past rely deeply on the control flow of the system under scrutiny. They piggy-back on the abstraction notions designed and applied by the engineers, thus allowing a compositional approach to specification and verification. In contrast, much of the logic inherent in a solution reliant on AI and machine learning techniques is typically the result of a data-driven process, leaving unstructured logic not easily amenable to this compositional approach.

We have identified various types of such properties to monitor in order to address this:

Statistical properties of training data The nature of collected training and test data is that it will contain less representative instances and even errors. However, one would expect a degree of statistical evenness of the training data over a number of metrics and balance in the representation of different classes in the dataset. Such properties we include in our specifications include the likes of *'Wind force values received during training will never exceed 50 knots'* (which identify hard-limits on the training data), and *'Once enough training data has been received, monitor whether wind values in new training data lies within 2 deviations of the previously received values'* (which focuses on the distribution of the data).

Data bias A second set of properties can monitor for known and documented undesirable data biases in the litter (and objects) being detected by the system. The process doubles up as a means of identifying, reporting and reviewing for such biases. Misclassification of litter or objects might send litter pickers to incorrect locations or incorrectly direct them to collect false positives, such as a rock being classified as a paper carton.

Verdict confidence Finally, drawing inspiration from related works reviewed above, post-deployment data monitors can also be used to ensure that the real-world scenarios being encountered are within the scope of the training class of data originally used. For instance, we envisage adding properties regarding the size, location and orientation of litter items identified at runtime to indicate how well represented these were in the original training set. Similarly, one can monitor to ensure that low-confidence predictions are not considered, and are omitted from the main system control.

The main challenges we see runtime verification techniques face here are those of efficient, data- and statistic-intensive monitoring. Techniques to perform parts of such monitoring synchronously, but delegating some parts to be performed asynchronously or offline is crucial. Still, this is not a trivial task, since even offline monitoring has its own challenge of potentially large data storage requirements in these cases.

5.2 Specification Adherence and Functional Correctness

The second class of properties is one for which runtime verification has been widely used: functional correctness. The key difference is that we are missing

the internal control logic flow within the model, information that is frequently used in expressing code-aware properties in traditional systems. For example, in traditional use cases such as [3], we had included properties such as *'As long as the system is executing the method* `verifyTransactionOrigin`*, the value of the transaction may not be changed.'* In use cases such as the one being considered in this paper, such control delineation is not always available, and one would have to depend more heavily on events visible outside the black box offered by the model. Properties which depend on such events include *'Once the drone starts accelerating in a direction and as long as it has been doing so for more than 5 seconds, it will not reorient the camera direction,'* and *'Once the battery is lower than 10%, no acceleration away from the base will be applied for more than 5s.'*

One type of functional behaviour properties are ones addressing the proper functioning of the underlying AI model. Monitoring for effectiveness of the outcome against information gathered at runtime (or potentially later e.g. whether an identified object was, in fact, litter) can prove to provide an effective means of feedback into the system.

5.3 Safety Protection

Following the idea of the shield in related work, there are instances where AI output can pose a safety risk. While this can be considered as part of functional correctness, it could be useful to treat separately when the stakes are high. Furthermore, a strict enforcement approach could be taken with safety concerns, leaving 'softer' monitor reactions focused on punishment and reward to more flexible functional requirements. In our case study, there is a substantial list of safety concerns that need to be considered. Here are a few examples: (i) Damaging drone and/or third party property: the monitor should ensure that if the drone is about to hit an obstacle, it diverts its course on time. Furthermore, if the battery power goes below a certain threshold, the monitor should ensure that the drone returns to base in time to avoid falling down. Similarly, if the drone loses contact with its control point, the monitor should guide it back to safety. (ii) Danger to human litter pickers: Initially, we see the system will be used to guide human litter pickers to locations where litter is detected. A set of properties can monitor these recommendations before they are presented to humans on mobile devices to ensure they are not guided to unsafe areas such as the edges of cliffs.

5.4 Regulatory Compliance

We finally move to the fourth role of runtime verification in these settings-regulatory compliance. We separate compliance into two parts: *operational* and *technological* compliance:

Operational compliance: As with all systems operating in a domain where real-world effects are regulated, compliance is a crucial element of the underlying system. The two facets of (i) building a solution to comply; vs (ii)

ensuring that the system complies, is the implementation vs. specification separation-of-concerns issue that runtime verification plays an important role in. In addition, in many cases, providing evidence of compliance is part of the compliance requirements themselves e.g. one may have to report financial transactions beyond a certain threshold to show that they were all appropriately handled. This separation is even more crucial in the case of regulation (when compared to requirements documents) since regulations frequently change independently of the underlying system.

We have investigated the use of runtime verification for regulatory compliance in the context of payment services [3, 2], building risk-based solutions to monitor and report for compliance requirements. We envisage a similar approach here, with regulation pertaining to drone operations, privacy, personal data, model prediction etc. For instance, one can add rules governing flight over third-party property due to privacy concerns.

Technological compliance: The second aspect is that of compliance not due to the real-world impact of the system, but rather compliance with regulation arising from the nature of the technology being used. Since AI is being regulated as a technology itself, these regulations have to be adhered to directly. We see the use of runtime monitoring and verification ideal here, especially when one finds requirements such as the forensic node and ITA harness discussed in Section 4.

5.5 Architecture

Following the overview of the various kinds of properties outlined in the previous subsections, we delve into more practical design decisions.

Alert/Action/Enforce: The dilemma of how intrusive a monitor should be is a palpable one in the area of runtime verification, particularly in the context of real-world systems. When it comes to monitoring of statistical properties of training data, we are not envisaging a runtime enforcement approach (in that data not satisfying these constraints is not used), but rather an opportunity to tag and quantify the frequency of instances. Although one can go further and have different levels of risk associated with the properties and have different appropriate actions in place. We have used this approach in the past on risk-based monitoring of financial transaction systems [2] where, depending on the severity of the violation caught (e.g. how late a refund occurs), a transaction is either simply reported (low-risk), tagged for compliance review (medium-risk) or stopped altogether (high-risk). Similarly, here, one can have such properties indicating a high risk of undesirable data, in which case it can be removed altogether e.g. *'Wind force values received during training will never exceed 100 knots'* may indicate data which is clearly wrong.

Other properties concerning safety and legislation might require a more intrusive approach, possibly enforcing properties to some extent, e.g. to avoid potentially serious repercussions such as fines or damages incurred due to

drone behaviour. This resonates with the idea of the shield introduced in the related work section and the harness proposed in the technology guidelines adopted by Malta[8].

Online/Asynchronous/Offline monitoring: A closely related issue is deciding whether to run the monitor in an online or offline fashion. Taking the latter option is preferred from the point of view of the computational and memory overheads that might be introduced. Properties concerning statistical analysis of data could be particularly prone to this issue. However, as explained above, properties which will not be enforced, can be performed asynchronously (or completely offline) from the training process. The other properties which require instant action and/or enforcement (e.g. concerning the drone's battery level) do not afford this luxury and the overhead concern would need particular attention.

Pre/Post deployment monitoring: An important choice in runtime verification of AI systems is what to monitor during training and what to monitor beyond deployment. Given enough resources, training-time properties could be useful to monitor post-deployment, particularly if live data is also used for training (e.g. by obtaining feedback from a human authority). In our use case, this will be applicable to the post-deployment litter identification model by monitoring information from identified and retrieved items to ensure that the use is within the constraints identified before. Moreover, should litter be misclassified, the imagery and the desired outcome could be used to update the model.

Black/White-box approach: While as reviewed in the related works, some data monitoring approaches could benefit from a white-box approach (e.g. statistical properties of training data), other monitoring concerns (e.g. checking that the drone doesn't reach geographically-prohibited zones) might best be kept completely separate as explained in the legal compliance section above. Such monitors do not differ much from what is traditionally performed in runtime verification solutions where the monitoring and verification engine are not interwoven with the underlying system, but kept architecturally apart.

The proposed monitoring and verification architecture is shown in Fig. 4. It is worth noting that the harness and forensic node are monitoring components (in the traditional sense) themselves.

6 Conclusions

AI-based solutions are being used to automate more complex and critical processes, requiring, on the one hand, more autonomy but increasing safety and

[8] Although related, they are not equivalent: the notion of a shield is a component of the harness because the former is mostly concerned with enforcing safety properties while the latter is also interested in other aspects such as collecting and analysing data more generally.

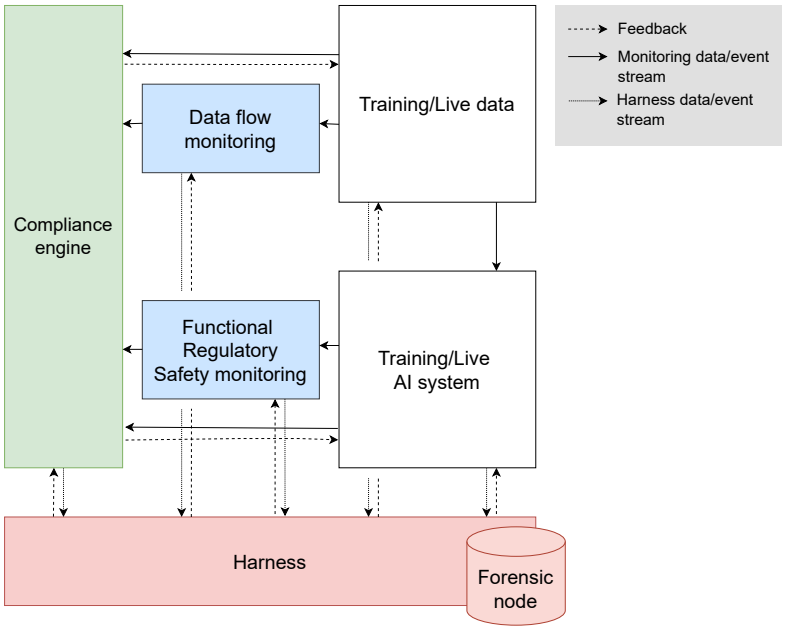

Fig. 4. The monitoring, verification and compliance layers.

legal stakes on the other. In this context, runtime verification which has a legacy of supporting improved system reliability post deployment, is increasingly being used to monitor AI systems.

In the regulatory scene, emerging legislation and guidelines are putting higher expectations on AI systems, such as the need for a harness and forensic node in the Maltese context when the risks are considered high. These developments have led us to explore further how all the pieces can fit together for a pragmatic runtime verification solution.

The AI-assisted litter detection and collection system being developed in Malta is an example of the various elements that need consideration before deploying such a system in real-life. In particular, we have organised the various concerns under four headings: data, functional, safety, and legal. These elements have been combined in a pragmatic architecture which also brings on board the Maltese regulatory guidelines on ITAs to include a harness and a forensic node.

We hope that through the comprehensive architecture proposed in this paper, the identified risks of complex and autonomous AI systems can be managed more effectively, opening up more possibilities, not least for an efficient way of cleaning rural Malta.

References

1. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 2669–2678. AAAI Press (2018). https://doi.org/10.1609/AAAI.V32I1.11797, https://doi.org/10.1609/aaai.v32i1.11797
2. Azzopardi, S., Colombo, C., Ebejer, J., Mallia, E., Pace, G.J.: Runtime verification using VALOUR. In: Reger, G., Havelund, K. (eds.) RV-CuBES 2017. An International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, September 15, 2017, Seattle, WA, USA. Kalpa Publications in Computing, vol. 3, pp. 10–18. EasyChair (2017). https://doi.org/10.29007/BWD4, https://doi.org/10.29007/bwd4
3. Azzopardi, S., Colombo, C., Pace, G.J.: A controlled natural language for financial services compliance checking. In: Davis, B., Keet, C.M., Wyner, A. (eds.) Controlled Natural Language - Proceedings of the Sixth International Workshop, CNL 2018, Maynooth, Co. Kildare, Ireland, August 27-28, 2018. Frontiers in Artificial Intelligence and Applications, vol. 304, pp. 11–20. IOS Press (2018). https://doi.org/10.3233/978-1-61499-904-1-11, https://doi.org/10.3233/978-1-61499-904-1-11
4. Cheng, C., Nührenberg, G., Yasuoka, H.: Runtime monitoring neuron activation patterns. In: Teich, J., Fummi, F. (eds.) Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019. pp. 300–303. IEEE (2019). https://doi.org/10.23919/DATE.2019.8714971, https://doi.org/10.23919/DATE.2019.8714971
5. Ellul, J., Galea, J., Ganado, M., McCarthy, S., Pace, G.J.: Regulating blockchain, dlt and smart contracts: a technology regulator's perspective. ERA Forum **21**, 209–220 (2020). https://doi.org/https://doi.org/10.1007/s12027-020-00617-7
6. Ellul, J., Pace, G.J., McCarthy, S., Sammut, T., Brockdorff, J., Scerri, M.: Regulating artificial intelligence: a technology regulator's perspective. In: Maranhão, J., Wyner, A.Z. (eds.) ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21-25, 2021. pp. 190–194. ACM (2021). https://doi.org/10.1145/3462757.3466093, https://doi.org/10.1145/3462757.3466093
7. European Union: Regulation (EU) 2019/881 of the European Parliament and of the Council of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification and repealing Regulation (EU) (2019)
8. European Union: Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts (2024)
9. García, A.A., Cambronero, M., Colombo, C., Llana, L., Pace, G.J.: Runtime verification of contracts with themulus. In: de Boer, F.S., Cerone, A. (eds.) Software Engineering and Formal Methods - 18th International Conference, SEFM 2020, Amsterdam, The Netherlands, September 14-18, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12310, pp. 231–246. Springer (2020). https://doi.org/10.1007/978-3-030-58768-0_13, https://doi.org/10.1007/978-3-030-58768-0_13

10. Gordon J. Pace, Joshua Ellul, I.R., Schneider, G.: When is good enough good enough? on software assurances. ERA Forum **23**, 337–360 (2023). https://doi.org/https://doi.org/10.1007/s12027-022-00728-3
11. Havelund, K.: What does AI have to do with rv? - (extended abstract). In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change - 5th International Symposium, ISoLA 2012, Heraklion, Crete, Greece, October 15-18, 2012, Proceedings, Part I. Lecture Notes in Computer Science, vol. 7609, pp. 292–294. Springer (2012). https://doi.org/10.1007/978-3-642-34026-0_22, https://doi.org/10.1007/978-3-642-34026-0_22
12. He, Y., Schumann, J., Yu, H.: Toward runtime assurance of complex systems with ai components. In: PHM Society European Conference. vol. 7, pp. 166–174. PHM Society (2022). https://doi.org/10.36001/phme.2022.v7i1.3361, https://doi.org/10.36001/phme.2022.v7i1.3361
13. Henzinger, T.A., Lukina, A., Schilling, C.: Outside the box: Abstraction-based monitoring of neural networks. In: Giacomo, G.D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020). Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2433–2440. IOS Press (2020). https://doi.org/10.3233/FAIA200375, https://doi.org/10.3233/FAIA200375
14. Könighofer, B., Bloem, R., Ehlers, R., Pek, C.: Correct-by-construction runtime enforcement in AI - A survey. In: Raskin, J., Chatterjee, K., Doyen, L., Majumdar, R. (eds.) Principles of Systems Design - Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 13660, pp. 650–663. Springer (2022). https://doi.org/10.1007/978-3-031-22337-2_31, https://doi.org/10.1007/978-3-031-22337-2_31
15. Lu, S., Lysecky, R.: Time and sequence integrated runtime anomaly detection for embedded systems. ACM Trans. Embed. Comput. Syst. **17**(2), 38:1–38:27 (2018). https://doi.org/10.1145/3122785, https://doi.org/10.1145/3122785
16. Lukina, A., Schilling, C., Henzinger, T.A.: Into the unknown: Active monitoring of neural networks. In: Feng, L., Fisman, D. (eds.) Runtime Verification. Lecture Notes in Computer Science, vol. 12974, pp. 42–61. Springer International Publishing (2021)
17. Mallozzi, P., Castellano, E., Pelliccione, P., Schneider, G., Tei, K.: A runtime monitoring framework to enforce invariants on reinforcement learning agents exploring complex environments. In: Proceedings of the 2nd International Workshop on Robotics Software Engineering, RoSE@ICSE 2019, Montreal, QC, Canada, May 27, 2019. pp. 5–12. IEEE / ACM (2019). https://doi.org/10.1109/ROSE.2019.00011, https://doi.org/10.1109/RoSE.2019.00011
18. of Malta, G.: Innovative Technology Arrangements and Services Act (2018)
19. Malta Digital Innovation Authority: AI ITA Blueprint Guidelines (October 2019), https://www.mdia.gov.mt/wp-content/uploads/2022/11/AI-ITA-Blueprint-Guidelines-03OCT19.pdf
20. Malta Digital Innovation Authority: AI ITA Guidelines (October 2019), https://www.mdia.gov.mt/wp-content/uploads/2022/11/AI-ITA-Guidelines-03OCT19.pdf
21. Malta Digital Innovation Authority: AI ITA Nomenclature (October 2019), https://www.mdia.gov.mt/wp-content/uploads/2022/11/AI-ITA-Nomenclature-03OCT19.pdf

22. Malta Digital Innovation Authority: AI System Auditor Control Objectives (October 2019), https://www.mdia.gov.mt/wp-content/uploads/2022/11/AI-ITA-SA-Control-Objectives-03OCT19.pdf
23. Malta Digital Innovation Authority: Technology Assurance Sandbox v2.0 Programme Guidelines (June 2020), https://www.mdia.gov.mt/wp-content/uploads/2022/11/MDIA-Technology-Assurance-Sandbox-TAS-Programme-Guidelines.pdf
24. Neufeld, E.A., Bartocci, E., Ciabattoni, A., Governatori, G.: Enforcing ethical goals over reinforcement-learning policies. Ethics Inf. Technol. **24**(4), 43 (2022). https://doi.org/10.1007/S10676-022-09665-8, https://doi.org/10.1007/s10676-022-09665-8
25. Pace, G.J., Ravn, A.P. (eds.): Proceedings Sixth Workshop on Formal Languages and Analysis of Contract-Oriented Software, FLACOS 2012, Bertinoro, Italy, 19 September 2012, EPTCS, vol. 94 (2012). https://doi.org/10.4204/EPTCS.94, https://doi.org/10.4204/EPTCS.94
26. Pace, G.J., Sánchez, C., Schneider, G.: Reliable smart contracts. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation: Applications - 9th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2020, Rhodes, Greece, October 20-30, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12478, pp. 3–8. Springer (2020). https://doi.org/10.1007/978-3-030-61467-6_1, https://doi.org/10.1007/978-3-030-61467-6_1
27. Pardo, R., Colombo, C., Pace, G.J., Schneider, G.: An automata-based approach to evolving privacy policies for social networks. In: Falcone, Y., Sánchez, C. (eds.) Runtime Verification - 16th International Conference, RV 2016, Madrid, Spain, September 23-30, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10012, pp. 285–301. Springer (2016). https://doi.org/10.1007/978-3-319-46982-9_18, https://doi.org/10.1007/978-3-319-46982-9_18
28. Pisani, D., Seychell, D., Schembri, M.: Detecting litter from aerial imagery using the SODA dataset. In: 2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON) (2024)
29. Rahman, Q.M., Corke, P., Dayoub, F.: Run-time monitoring of machine learning for robotic perception: A survey of emerging trends. IEEE Access **9**, 20067–20075 (2021). https://doi.org/10.1109/ACCESS.2021.3055015, https://doi.org/10.1109/ACCESS.2021.3055015
30. Schembri, M., Seychell, D.: Small object detection in highly variable backgrounds. In: 2019 IEEE 11th International Symposium on Image and Signal Processing and Analysis (ISPA). pp. 32–37 (2019)
31. Sha, L.: Using simplicity to control complexity. IEEE Softw. **18**(4), 20–28 (2001). https://doi.org/10.1109/MS.2001.936213, https://doi.org/10.1109/MS.2001.936213
32. Turing, A.M.: Checking a large routine. In: Report of a Conference on High Speed Automatic Calculating Machines. pp. 67–69 (June 1949)
33. Turing, A.M.: Computing machinery and intelligence. MIND: A Quarterly Review of Psychology and Philosophy **LIX**(236), 67–69 (October 1950)