The Importance of Assuring Algorithm-based Verification Agents

Odd Ivar Haugen[1,*]
[1] DNV GL, Group Technology & Research, Trondheim, Norway

ABSTRACT

Safety verification is about creating trust and building confidence that a system is safe and conforms to the specified requirements. The term confidence means the assurance level which is built by generating objective evidence through activities such as testing. In the maritime industry, classification societies (a.k.a. *class*) are instrumental in the assurance process of maritime safety-critical systems. These systems become more and more software-intensive, enabling a high degree of complexity and even autonomy. Automatization in the verification effort emerges as system complexity increases and cost-pressure rises. An automatic condition-based survey scheme, utilizing data from sensors and algorithms is seen as more efficient and effective than the traditional calendar-based survey scheme performed by trained class surveyors (people) today. In the assurance of self-learning adaptive systems such as autonomous navigation systems, possibly based upon Machine Learning (ML), online safety monitors may become instrumental in creating relevant safety evidence. These monitors may also be based on ML and may be adaptive, resulting in one adaptive ML-algorithm verifying another adaptive ML-based target system. Class surveyors are test engineers who are *verification agents* and generate evidence about the system safety level. The verification algorithms, such as a Condition Monitoring system should also be categorized as a *verification agent*; an Algorithm-based Verification Agent (AVA). Moreover, class surveyors represent an independent Verification Organization. Independence in the verification effort increases the assurance level because the level of evidence objectiveness increases. If the AVA is developed by the target system developer, it decreases the evidence objectiveness and affects the agency of humans in the verification. This paper argues that AVAs must be assured at a level reflecting their agency within the verification effort, and the target system criticality. The same cognitive and societal biases infecting the target system may also affect the AVA if it is developed by the same organization as the developer, possibly masking critical defects, and making the generated evidence less trustworthy.

Keywords: Objectivity; Algorithm-based Verification Agent; Verification of Algorithms; Emic Verification; Etic Verification.

1. INTRODUCTION

Independent verification is an important enhancement of in-house verification in creating stakeholder (and societal) trust in safety-critical systems. In the maritime industry, some parts of the independent verification effort are conducted by classification societies (a.k.a. class). Class publish class rules, imposing safety requirements on the technology, development process, building, and operation of maritime vessels (DNV GL, 2018a; DNV GL, 2018b; DNV GL, 2018c). The verification process, like any other verification process, describes tasks, activities, artefacts, documentation, and roles. The industry must comply with these rules to obtain the certificate necessary to be allowed to operate within the jurisdiction of the national maritime authorities (Ship Safety and Security Act,

* Corresponding author: +4791715040, odd.ivar.haugen@dnvgl.com

2007). Furthermore, class acts as an independent Verification Organization (VO) in certain activities of the verification process (DNV GL, 2018a), and upon certain system artefacts. In other activities (and other artefacts), class acts as an assessor, and/or a test witness.

Historically, class has focused on the physical (safety-related) properties of a vessel, such as hull strength, separate watertight compartments, reliability of the propellers etc. Although physical properties of the vessel are still imperative for the safety of maritime vessels, safety relies increasingly on software-intensive control systems of growing complexity, e.g. (DNV GL, 2018d). The latest step in this complexity stairway is autonomous vessels using control-algorithms based on Machine Learning (ML) (DNV GL, 2018e; Vartdal, Skjong, & St.Clair, 2018).

Whether systems are based on Artificial Intelligence (AI) technology, or there is a more traditional highly complex software-intensive system, we expect that algorithms will play an increasingly important role in the verification effort of such systems. These algorithms generate safety evidence; they possess a certain role; they perform certain tasks and activities; they assume a certain responsibility, and are therefore termed *Algorithm-based Verification Agents* (AVA). Introducing AVAs, perhaps based on ML, into the verification effort of complex systems, challenges the roles and processes outlined in the class rules (i.e. verification process).

This paper discusses aspects threatening the objectiveness and trustworthiness of AVA-generated safety evidence, and more specifically, the **evidence instance trustworthiness** (see section 2 below) in a situation where at least part of the verification effort is performed by algorithms, using maritime classification as a background theme. Other industries, like aviation, comprehensive and dedicated guidance on tool qualification processes exists (RTCA, 2011). Although comprehensive, this guidance does not discuss *agency* within the verification effort as a key concern with respect to the trustworthiness of the generated evidence, which is the focal point in this paper.

2. EVIDENCE PROPERTIES AND OBJECTIVENESS IN THE GENERATION OF EVIDENCE

Objectivity is central in the definition of “Verification”: "*The process of providing objective evidence...*" (Institute of Electrical and Electronics Engineers [IEEE], 2016, p. 26). Other standards have a similar definition e.g. (International Electrotechnical Commission [IEC], 2014, p. 28). Systems are examined (i.e. reviewed, analysed, tested) by class and others to generate evidence that confirms compliance with the class rules. The body of evidence must possess some specific properties to be useful in demonstrating such compliance at an adequate level of confidence. "*The overall assurance that is achieved is ultimately determined by the evidence that is put forward to support the argument*" (Hawkins & Kelly, 2011, p. 1).

In maritime classification, Safety Cases (Ministry of Defence, 2007, p. 9) are not used explicitly; however, elements of a safety case are implicit in the certification process. Here, the top "claim" is that the system is safe and complies with the class rules. The objective of the verification process (specified by class) is to generate adequate evidence proving conformance. Furthermore, (Hawkins & Kelly, 2011, pp. 2-4) list some important evidence properties that are instrumental if the evidence is to be able to create confidence in the target system:

- **Evidence type capability** - the capability of a certain type of evidence (e.g. from testing, analysis, review, or perhaps in-use) in proving system safety
- **Evidence instance capability** - the quality or relevance of a particular instance of evidence
- **Evidence instance trustworthiness**

Evidence trustworthiness is determined by the process and tools used in the generation of evidence instances (Hawkins & Kelly, 2011). One aspect of an instance of evidence being found trustworthy is that it is *objective*. It should be recognized that trust may be generated by other means than objectivity in the body of evidence, e.g. through authority. It will be shown that the latter approach to create trust will also come into play when algorithms create evidence.

In the frame of this paper, a Verification Organization (VO) generates evidence; an *assessor* or *test witness* does not. To qualify as a VO, the organization must specify the testing procedures;

must test the products/system; must use its own tools; and must follow up non-conformities (IEEE, 2016, p. 198). A party/organization witnessing a verification activity, or assessing verification artefacts, does not qualify as VO.

3. THE POSITION AND VALUE OF INDEPENDENCE IN THE VERIFICATION EFFORT

Before discussing the position of class as a VO in the (safety) certification of marine vessels within a framework of algorithm-based verification, some aspects of independence, and their relationship to objectiveness will be elaborated. Verification is about creating trust, and building confidence that a system conforms to the requirements (i.e. class rules). Trust and confidence are built by generating **objective evidence** through activities like testing. Of course, trust and confidence are also built by the knowledge that the supplier has been proven capable of producing the same, or similar systems in the past, or in other ways substantiate its organizational maturity.

When the system is put into operation, the stakeholders need assurance that the system performs as anticipated. The level of stakeholder confidence (i.e. the assurance level) must increase as dependence upon the system increases. Stakeholders' dependence upon a safety-critical system (e.g. an autonomous navigation system), must be supported by a reasonably high level of confidence that the system is safe during operations. There are several activities and properties in the life-cycle of a system that create trust and confidence. Apart from the organizational maturity of the supplier/developer, the rigour and intensity of the verification effort is imperative. Other aspects, such as transparency into why the supplier is deemed mature or, transparency into how the verification process was conducted, and who did it, as well as transparency into the functionality of the product itself, are important in the creation of trust.

The question about *who* (human or other) performed the verification in the context of evidence objectivity quickly becomes a question about independence in the verification effort. Independence in the verification effort can be attributed as the attachment, and the dependency of the VO, and the tools used, with respect to the developer of the system. Dependency and attachment can take different forms, such as organizational, financial, and technological (IEEE, 2016, p. 198). Even cultural, educational and ethical/moral attachments can influence independence.

Independence in the verification effort is one aspect that increases the assurance level, because the level of objectiveness in the generated evidence increases, and thereby increases trust and confidence in the system. Other aspects in the verification effort that support objectivity include, for example, transparency, utilizing sound scientific methods, using acknowledged tools, and domain knowledge of the verification practitioner. Such aspects also increase the probability of adequate evidence instance quality, the elaboration of which is outside the scope of this paper. The question though that the current paper attempts to address is "How will independence (and evidence objectiveness), and through that evidence trustworthiness, be affected if tasks are performed, and decisions are made in the verification effort by (ML) algorithms?"

4. THE ASSURANCE OF SAFETY-CRITICAL SYSTEMS IN THE CONTEXT OF MARITIME CLASSIFICATION OF SHIPS IN OPERATION

As part of assuring the safety of maritime systems, verification activities (e.g. surveys) are performed when the vessel is in operation. An example of such requirements is the survey requirements of one specific maritime safety-critical system; the Dynamic Positioning system (DP). The purpose of the DP is to maintain the position of a vessel using only thrusters and propellers. According to the International Maritime Organization (IMO, 2017):

> *an annual survey should be carried out within three months before or after each anniversary date of the Dynamic Positioning Verification Acceptance Document 1. The annual survey should ensure that the DP system has been maintained in accordance with applicable parts of the Guidelines and is in good working order*. (p. 12)

And according to the same guideline: "*a periodical testing at intervals not exceeding five years to ensure full compliance with the applicable parts of the Guidelines*" (p. 12). These requirements are reflected in the class rules published by the classification societies.

To-date, the requirements are *calendar-based*, as opposed to *condition-based*. Class sends surveyors around the world to perform the above activities on-board every vessel that holds a certificate. Moreover, the vessel is drydocked as part of conducting the five-year survey. There are several issues with this practice that may be discussed, but for the purpose of this paper, two issues will be discussed: cost, and the capability of generated evidence type. Although it is the *trustworthiness* of the evidence that comprises the focal point of this paper, the evidence type capability is a motivating factor for a change.

Sending a surveyor on-board, and possibly taking the vessel out of service once a year imposes costs that the vessel owners would like to avoid. The five-year survey is much more comprehensive and requires drydocking the vessel so that vital equipment can be dismantled and manually inspected. It is clear that this activity is expensive, and the vessel owners could save money if such an activity could be done at longer intervals, or preferably, only when required. Moreover, there is always a risk of damage, or even destroying the equipment when dismantling and re-fitting it.

The way the activities above are conducted, reflects what was considered as most important for safety in earlier times; the physical condition of components (in isolation). While still important, in modern vessels the physical condition of components is just one aspect that is important for safety. The interaction *between* components such as between the system software (i.e. control system), and the physical components (actuators, sensors etc.), is also crucial for safety (Leveson, 2011). Moreover, when utilizing Artificial Intelligence (AI), such as autonomous navigation, decision-making is removed from the crew and handed over to software.

Alterations, such as software updates, is hard to verify for a local class surveyor. Other types of skills and tools are needed to "inspect" a complex software system, compared to the physical condition of a component, like the wear and tear on a propeller shaft. In fact, the whole idea of a calendar-based survey scheme for software-based systems may be questionable. Software does not wear out in the same way as the shaft of a main propeller. Of course, software may become obsolete as a result of changes in the system, or the working environment which it is part of. If autonomous navigation systems are to be based on machine learning, such as Online Learning Neural Networks (OLNN), it is obvious that an alternative method must be found to generate timely evidence with adequate capabilities.

5. EXAMPLES OF FUNCTIONS THAT CHALLENGE TRADITIONAL VERIFICATION METHODOLOGY

The examples below describe two systems that are very different in their role regarding the target system. The first example, an autonomous navigation system, is *part* of the target system. The second example is an AVA, tightly *coupled to* the target system. Both may rely upon complex algorithms, possibly ML-based, which challenge the traditional assurance framework laid out by the classification societies.

Example 1 (Vessel control system): An autonomous ferry or bulk carrier operating in a coastal area needs to classify possible obstacles that may interfere with the current course and speed (e.g. making the distinction between a canoe and a log). Even after correctly classifying an obstacle as a log, the system must decide whether to run over it or alter the course to avoid a collision. Of course, running over a canoe is not an option. The canoe/log must be correctly classified in all sea states and light/weather conditions. A classification such as this may be implemented by an Artificial Neural Network (ANN) such as a Learning Vector Quantization (LVQ).

Example 2 (System self-verification): Calendar-based inspection is not sufficient with respect to the above example, or any other complex safety-critical vessel control system. Industry stakeholders want to shift from calendar-based inspections and maintenance to online Continuous Monitoring (CM), enabling a more efficient Condition-Based Maintenance (CBM). The condition of

the system is inferred by analysing data (i.e. using algorithms) from sensors placed within, and on different on-board equipment. A *sensor* may be physical, measuring a physical property or condition of a component, however, it may also be a software-implemented *observer* within the vessel control system that monitors timing, event sequences, and other internal states. This means that a condition may be anything between rust development on a steel plate to a sequential shift between internal states in the control system. If the sensor is capable of directly measuring the physical property of a component, and the CM system only raises a warning if the measurement exceeds predefined limits, the CM may not be very complex. However, if the CM system must infer an emergent property (like the safety level in complex systems) based on multiple (seemingly unrelated) measurements, a predictive neural network may be used. It is obvious that verifying the correctness of the latter will be very difficult, or impossible to perform during an inspection tour done by a class surveyor. This system replaces to some degree the evidence earlier generated by the surveyor, which means that the CM system takes the role of a *digital surveyor*. The generated *evidence type capability* is improved, but what about its *objectivity*?

While some of the traditional verification tasks will still be performed by humans, other tasks may have to be performed by algorithms. Human expertise, their independence from the development organization, tools, and their applied methodology can easily be scrutinized, and thereby the trustworthiness of the generated evidence can be assessed. On the other hand, in case of an algorithm-based verification of the target systems, where the verification agent is a black-box, how can we assess the trustworthiness of the generated evidence? In Example 1: that the safety level is maintained in every possible situation and over time, and in Example 2: that unsafe system conditions are reliably detected.

6. EXAMPLES OF ALGORITHM-BASED VERIFICATION AGENTS

The following examples illustrate that an algorithm can have different roles with different purposes in the verification effort. The functionality described is not new, such tools have existed for decades. However, the degree that we have to rely upon them (due to the complexity in the target system), their (autonomous) decision-making capacity, and their intractability, motivates increased focus on them. Earlier, they were just a “tool” with a limited set of functions, fully controlled and understood by the verification practitioner. Now, they are moving away from being merely a *tool* towards being an *agent*.

Before describing the examples, we must clarify the term *test data*. Test data means two different things in the following two examples. In Example 1, test data means data put aside when training an ML-algorithm to estimate its performance when faced with unseen data. In software testing terminology, however, test data is defined as: "*data created or selected to satisfy the input requirements for executing one or more test cases...*" (International Organization for Standardization [ISO], 2013, p. 7). This definition is used in Example 2.

6.1. Example 1: Generating Artificial Test Data through Automated Test Data Trajectory Generation Algorithm (ATTG)

One obstacle in utilizing Artificial Neural Networks (ANN) or other types of ML-based algorithms in safety-critical control systems is the lack of sufficient sets of data for adequate test coverage. Much of the available data is used in the training of the Neural Network. The rest is supposed to be used for testing the algorithm. Data is often scarce and lacks sufficient representation of the feature space to adequately train the ANN. A way of improving the situation and estimating the generalization performance is to use cross-validation. In this case however, all available data is used both in the training, and in the testing, leaving no *independent* test data. The lack of representativeness (epistemic constraint) in the training and test data set with respect to rare dangerous events is a very important safety factor to consider.

To improve this limitation in the original data set, Automated Test Data Trajectory Generation Algorithm (ATTG) (Taylor, et al., 2006) may be used to generate artificial statistically related data

points covering an increased feature space used as test data of the ANN. Such testing is termed *knowledge testing* (Taylor, et al., 2006). ATTG design requires a number of human decisions that are susceptible to subjectivism and biases. These decisions include selecting: a distance measurement, a clustering technique and desired number of clusters, and a representative component from each cluster (Taylor, et al., 2006).

6.2. Example 2: Generating Test Data Using Genetic Algorithms (GA)

This particular example is related to the one presented above in the way that both approaches can generate input to the system, and set up the operational environmental conditions. ATTG can be used in generating a continuous input space suitable for testing an ML-based algorithm, while Genetic Algorithms (GA) can be used to generate discrete test data that both explores the input space, and satisfies the input requirements of the system.

In a simulator-based test environment, when testing a Dynamic Position (DP) system mentioned earlier, typical test data are the speed and direction of the wind and ocean current. When the DP is enhanced with autonomous obstacle-detection and navigation decision functionality, test data describing obstacles (ship, canoe/log etc.), and environmental properties important for successful detection and collision avoidance become a major issue. Test data such as the shape of an obstacle, its size, speed and course, and environmental properties such as light (e.g. sunshine, dark, dim) and weather conditions (e.g. rain, snow, fog) must be included into the test data describing a test case (see above for the definition of test data in software testing).

GA can be used to explore the input space and at the same time obey real life restrictions. A canoe will most probably have a speed less than the average speed of the current world record for 200 m, which is about 11 knots, at least over a considerable time period. However, what if there is more than one person in the canoe? What would a “realistic” maximum speed be then? Furthermore, 100 m long Platform Supply Vessel (PSV) will most probably need some time to come to a full stop from an initial speed of a few knots. The GA *input population* and the following *generations* must obey these physical limitations to generate realistic test cases. However, it is important, as in all forms of SW testing, to challenge the system by testing the system response to inputs at the edge of the legal input space. The response to equipment faults, perhaps in combination with other inputs at the edge of the legal input space, is also imperative to test.

A *fitness function* governs parent selection, and the GA *evolves* through that parent selection and through *mutation*. Observers within the autonomous detection and decision functions act as input to the fitness function. These observers must be chosen to reflect how close the system is to making a wrong detection/decision as a response to a particular test case. The International Regulations for Preventing Collisions at Sea (COLREG) (IMO, 1977) sets requirements about how to navigate to avoid collisions. In case two vessels are on a collision course, based on the situation (course, speed etc.) COLREG grants one vessel “stand-on”, while the other must “give way” (IMO, 1977). In the context of GA-generated test data, a point of interest could be the *pivoting point* of “give way" as opposed "stand-on" that must be reflected in the GA *fitness function*. Test data can be automatically generated to investigate or challenge the autonomous navigation system around this “singularity”. As can be seen from the above discussion, GA design requires domain knowledge, and a number of human decisions susceptible to subjectivism and biases.

6.3. Example 3: Online (safety) Monitors

As mentioned above, in the quest for more flexible and efficient verification in operation, the maritime industry would like to develop a verification regime based on sensor and algorithm-intensive systems, i.e. a CM system, which may be seen as a form of online safety monitor. Moreover, if the safety-critical control system is a form of Online Learning Neural Network (OLNN), online monitors may even be required to ensure that the control systems do not become unsafe as a result of the continuous learning during operations (Taylor, et al., 2006).

6.4. Conclusion from the Examples

Product verification generates *primary evidence* by observing the response from the target system to some input, or investigating/reviewing a model representing the target system. *Process verification* generates *circumstantial evidence* by inspecting artefacts related to the process and standards used in the development process of the target system. Both types of verification generate evidence that ultimately indicates the degree of confidence stakeholders can place in a system. Drawing conclusions from *circumstantial* evidence requires inference. Questions have been posed regarding the degree to which such evidence is capable of providing proof towards a conclusion about the *product* (i.e. the needed strength of the inference).

A sound development process increases the probability of a good quality system. However, product verification becomes increasingly important (and difficult) facing complex software-intensive systems, possibly utilizing AI. Moreover, in AI systems, the *"...model's reasoning can therefore be considered independent of the software implementation*" (Douthwaite & Kelly, 2018, p. 96), or in other words, ML-based and adaptive systems incorporate additional components of knowledge that are not represented in the software code. This means that other aspects in the development process than are traditionally considered, become increasingly important.

A new category of evidence related to product verification is evidence generated from the verification of verification *tools* (ATTG/GA) and online monitors, such as CM systems. An online monitor may be seen as a form of an algorithm-based verification practitioner or agent; an ***Algorithm-based Verification Agent***, or *AVA*. The tools (e.g. monitors) then generate primary evidence (i.e. product verification). This category of evidence may be termed as *indirect primary evidence.* The questions about process verification can also be put to this new category of evidence: To what degree does evidence generated from the verification of tools and monitoring systems (i.e. indirect primary evidence) support a safety claim made about the system? Although the definite answer to this question will not be given here, we argue the *importance* of assuring these algorithms, and of being aware of their *position* within the verification effort.

7. HUMAN INFLUENCE OVER DECISIONS MADE BY AN ALGORITHM-BASED VERIFICATION AGENT (AVA)

Algorithms are mainly developed by humans: "...*the human influence in algorithms are many: criteria choices, optimization functions, training data, and the semantics of categories, to name just a few.*" (Diakopoulos N. , 2016, p. 23). Algorithms make prioritizations/ranking, and associations, they classify and filter. These tasks are governed by a number of design decisions made by developers, such as the criteria used for ranking, taxonomy for classification, rules for filtering, as well as the degree of correlation between items (which sometimes are confused with causation). If the Algorithm-based Verification Agent (AVA) is ML-based, selecting, tagging, filtering training and test data, as well as decisions made about the necessary size of the data set, are all made by humans.

Irrespective of whether the knowledge is incorporated into the code or is based upon training data, an online monitoring system can act like a stethoscope/microscope depicting a relatively objective "state of affairs", or it may be more like a pair of toy binoculars where enjoyable pictures are built into the toy, so you see "bright skies" whatever you are looking at. Or, according (Sandvig, Hamilton, Karahalios, & Langbort, 2014):

> *the president of American, Robert L. Crandall, boldly declared that biasing SABRE's search results to the advantage of his own company was in fact his primary aim. He testified that "the preferential display of our flights, and the corresponding increase in our market share, is the competitive raison d'etre for having created the [SABRE] system in the first place" (Petzinger, 1996). We might call this perspective "Crandall's complaint:" Why would you build and operate an expensive algorithm if you can't bias it in your favour?* (p. 2)

It is clear that an AVA which can possibly be ML-based carries biases inherited from the developer. If the AVA and the target system are developed by the same organization, such biases may mask critical flaws within the target system. This is similar to when humans verify their own work - only in a subtler way.

8. BIASES SWAYING DECISIONS MADE BY AN AVA

Biases may emerge as a result of human cognitive flaws (psychological effects and groupthink), and/or as a result of societal pressure (e.g. financial, managerial, organizational etc.). For the purpose of this paper, these two types are respectively called *cognitive bias* and *societal bias*. There are interactions between these two types, such as groupthink that emerges from being in the same organization and thereby sharing a common set of values. The American-Israeli psychologists Daniel Kahneman and Amos Tversky were instrumental in the research and description of cognitive biases, such as confirmation bias (Daniel Kahneman, n.d.). While cognitive biases are inherent in being human (there is no way you can erase these biases from your brain) (Kahneman, 2011) , societal biases may be more apparent and thereby more readily constrained.

Both types of biases, in any form, are the enemy of *objectivity*. Although the sense of objectivity is indistinct, the word carries some properties that most people can agree upon, like impartialness and value-neutralness. Some may think of objectivity as a synonym for truth, which is not the understanding of objectivity in this paper. This means that even a highly objective body of evidence does not reveal the ultimate truth about the target system.

Impartialness and value-neutralness must not be understood as of *having no interest*, only that they indicate viewpoints that do not favour any parties that have invested time, money, and prestige into the realization of the target system. An independent verification organization does play a part, and does have values that inflict upon their decisions in the verification process. In the case of a classification society, the purpose is to advocate and secure the safety goals given by the national maritime authorities. Biases carry shared values, loyalty (to the organization), financial and managerial pressure, and limitations in (common) tools used in the development and in the verification. There is no reason to believe that developing an AVA is free from these biases. According to (Kitchin, 2017):

> *And while the creators of these algorithms might argue that they 'replace, displace, or reduce the role of biased or self-serving intermediaries' and remove subjectivity from decision-making, computation often deepens and accelerates processes of sorting, classifying and differentially treating, and reifying traditional pathologies, rather than reforming them.* (p. 19)

The challenge is that the algorithms are more intractable when it comes to understanding their decisions, making the biases that we know are present and may result in swayed decisions, more difficult to identify for an assessor validating the quality of the generated evidence. *"The opacity of technically complex algorithms operating at scale makes them difficult to scrutinize, leading to a lack of clarity for the public in terms of how they exercise their power and influence*" (Diakopoulos N. , 2015, p. 398). A framework for analysing biases in computer systems is outlined by (Friedman & Nissenbaum, 1996). Perhaps such a framework can be used when analysing biases in AVAs as well? Biases can be characterized based on their source: "Pre-existing Bias", "Technical Bias", and "Emergent Bias" (Friedman & Nissenbaum, 1996).

- **Pre-existing Biases**: Include cognitive, and societal biases mentioned earlier. Biases generated from being a member of the same developer organization, sharing values and goals: "*They can also reflect the personal biases of individuals who have significant input into the design of the system*". (p. 333)
- **Technical Biases**: Can originate from common tools used in the development of the safety-critical system that is to be verified, and the tools used in the development of the

algorithm-based verification agent. Moreover, common (generic) algorithms used in both systems may introduce such biases.

- **Emergent Biases**: Biases arising from the use of the system. These will not be discussed further in this paper.

Product and systems, such as an AVA, will be biased in one direction or another by the developer organization, swaying the decisions made by the AVA. It may therefore show an opaque and skewed picture of the "state of affairs" in the target system. At the same time, AVAs are complex systems in their own right, making it difficult for human verification practitioners to scrutinize them. A human verification agent may no longer be capable of performing vital parts in the process of generating adequate primary evidence from certain types of systems, such as autonomous navigation systems. AVAs, such as GA-generated test data, may therefore become required in the generating of certain types of evidence with sufficient level of intensity and rigour. However, the above discussion shows that if humans are not vigilant, biases cause a decrease in the level of objectivity and trustworthiness in the evidence generated by the AVA.

9. POSSIBLE PITFALLS AND RISKS IN CREATING A TRUSTWORTHY BODY OF EVIDENCE (THE ILLUSION OF TRUST)

Objectivity in the processes serves the function of promoting trust in the outcome of that process (i.e. the products). In the case of a verification process, the product is a body of evidence that is supposed to support a safety claim/expectation about the system. As mentioned above, objectivity is not a rigidly defined property. However, another working description could be: "...*a set of norms that obliges persons or group of persons to apply impersonal modes of reason in the course of their inquiries or deliberations*" (Axtell, 2016, pp. Chapter "Introduction: A Valuable but Contested Concept", 2. paragraph). Objectivity is a property of a process, not a product. Let us assume that an objective process leads to objective outcomes, or products: "*To call the result (product) of inquiry objective is on a social level to endorse those products as trustworthy due to characteristics of the process by which they were produced*." (Axtell, 2016, pp. Chapter "Introduction: A Valuable but Contested Concept", 2. paragraph). This must not be confused with the *quality* of the outcome. As we all know, ensuring that the process of developing a product holds up to some standard (i.e. process verification), does not ensure adequate quality of the outcome of that process. To ensure adequate quality of the product, we must perform product verification which means generating primary evidence.

From the above discussion, we can conclude that an objective process of generating evidence results in the evidence being trustworthy. The key is therefore that the process by which an AVA produces evidence, must be objective. Objectivity is a normative concept and places requirements on our thinking. It requires us to distinguish facts from opinions, and avoid the previously mentioned biases. The "thinking" (reasoning and decisions) of the AVA must be objective, in order to create trustworthy evidence. Objectivity is unfortunately not the only way of creating trust. Authority can do the same: "Trust me, I'm a doctor...", or "Trust me, I'm an expert from a class society with some 150 years of successful history...". Creating trust through authority is like saying: "You should trust me; therefore, you should trust my proposition without evidence". Authority influences our critical thinking in that we have an inherited *Authority Bias*: "*People at positions with formal authority are often expected to make better decisions and fewer mistakes, and therefore their opinions and contributions are given higher weight*" (Hinnosaar & Hinnosaar, 2012, p. 1), which may lead to *obedience*. It is hard to say whether we yield to a proposition out of trust or obedience, the effect remains: lack of demand for scrutinizing the body of evidence.

Trust may also, quite easily as it seems, be created by constructing a coherent story. This story may not be objective, or even true, the only criterion for its ability to create trust and confidence is that it is coherent. "*Subjective confidence in a judgement is not a reasoned evaluation of the probability that this judgement is correct. Confidence is a feeling, which reflects the coherence of the information and the cognitive ease of processing it*" (Kahneman, 2011, p. 212). In this case,

propositions (stories) target what Daniel Kahneman calls the brain's "System 1" (the "system" that has a strong urge to jump to conclusions without a shred of evidence supporting the conclusion), and the cognitive bias of "The Illusion of Validity" (Kahneman, 2011).

A coherent story may be in the form of a *Safety Case* (Ministry of Defence, 2007, p. 9). Without ensuring adequate objectivity in the body of evidence supporting the top claim of a Safety Case, it is just another *coherent story* created to build subjective confidence. Another example is by stating that a safety standard like (IEC, 2010) has been followed in the development process of the safety system. This standard is regarded by some industries as THE safety standard, creating an aura of authority around it, and at the same time it helps building a coherent story about the development process. On a side note, the author has not seen evidence proving whether the reliability of safety systems built according to (IEC, 2010)is a correlation or in fact causation. The same way as both a doctor and a coherent story can create an illusion of trust, which is merely a notion of subjective confidence or obedience, so can an AVA. Algorithms can create non-evidence-based trust through authority (Lustig & Nardi, 2015), and/or being part of a coherent story.

If an AVA becomes a "divine oracle", resisting any attempts to independently scrutinize its precondition for its (autonomous) decisions and outputs, then built-in biases (i.e. cognitive/societal, pre-existing/technical) may go undetected and only create subjective confidence in the way described above. It is therefore imperative to minimize, or at least create some balance, in the biases in the development of AVAs so that the process by which the agent generates evidence becomes as objective as possible if used in the verification of safety-critical systems.

10. AFFECTING THE OBJECTIVENESS IN THE WORKINGS OF AN AVA

We can divide AVAs into two categories: *Etic* (not to be confused with Ethic) and *Emic*. These two terms are borrowed from the social and behavioural science, and in the context of AVAs, they depict different viewpoints related to the verification process. *Etic*: investigating the behaviour of people or system from an outsider's viewpoint using universal reference points. This is opposed to *Emic*: an investigation and explanation by the viewpoint of the people *within* a culture using their own perspective and concepts; an insider's viewpoint. In the context of AVA:

- *Etic*-AVA: generates evidence based upon a **different viewpoint** from the developer of the target system.
- *Emic*-AVA: generates evidence based upon the **same viewpoint** as the developer of the target system.

The terms *Emic* and *Etic* can also describe the detachment and consequently, the alternative viewpoints in the verification effort when performed by the target system developer, as opposed to an independent Verification Organization (VO). *Emic verification* is conducted by the same organization developed the target system, and *Etic verification* is conducted by an independent VO. The following subsections elaborate on these two categories of AVAs. The first section discusses the case when the AVA is developed and controlled by a different organization than the developer of that target system. The second section promotes *Etic verification* of *Emic*-AVAs.

10.1. Constructing an *Etic* AVA

The most apparent way of avoiding or minimizing shared biases in the workings of an AVA is that it is developed and controlled by an independent Verification Organization, making the agent what might be thought of as *etic* (see above). The agent can be seen as just another verification tool, like an independent system simulator used in a simulator-based test environment. The *Etic*-AVA generates (or directly contributes in generating) primary evidence about the system and the independent organization developer may be thought of as the VO through exerting direct control over the agent.

For online verification agents (e.g. Online Learning Neural Networks monitors), this solution may be challenging because there is a risk of the agent interfering with the safety-critical system. The responsibilities must be clear and uncertainty about a third-party agent interfering with the control system may be unacceptable. Moreover, the distinction between the control system and the *etic* AVA may become vague when the agent is incorporated into the running system, undermining the responsibility of the system supplier. An online *Etic*-AVA may challenge the understanding of the role such an agent possesses. How close is such an agent to possess parts of the role of a human operator? One responsibility of a human operator is to react upon, and possibly take control, when the system is no longer capable of handling the situation. To do this, the human operator must monitor indicators that enable him/her to recognize such situations. Can such a third-party *Etic*-AVA negatively influence the rigour of which the built-in robustness is incorporated? The answer to such a question is outside the scope of this paper, but the specific concern should be mentioned.

Another problem with an *Etic*-AVA is the need for accessing internal states or proprietary communication channels that the vendor considers a business secret. However, as the class societies act as both regulators through their class rules, and independent VOs through the classification process, they can place requirements upon the systems that are to be certified according to those rules, so that the system is able to interface such an agent. The interface between the system under test and the *Etic*-AVA may be standardized allowing any third-party Etic-AVA to be utilized. In any case, as indicated above, the role of an online verification agent should be well-defined.

10.2. Verifying an *Emic* Algorithm-based Verification Agent (*Emic*-AVA)

For different reasons, for some of which were outlined above, an Etic-AVA may not be feasible. An agent developed by the target system developer organization, may be termed an *Emic*-AVA (*emic* - see explanation above). Just as with *emic verification* (i.e. verifying your own work using your own conceptual framework), an *Emic*-AVA may also be infected with the same preconceptions and biases as the target system, decreasing the objectivity and thereby the trustworthiness of the generated evidence. The *Emic*-AVA generates primary evidence, and the organization developing the target (safety-critical) system, therefore also becomes the VO, making any independent party that may be involved in the verification effort, an assessor (i.e. not creating evidence, but merely assessing evidence properties).

When using an *Emic*-AVA to generate primary evidence from a safety-critical system, such an agent should be subject to *etic verification,* to minimize, or at least balance, the shared biases between the AVA and the target system. This will increase the objectiveness and thereby the trustworthiness of the generated evidence beyond merely subjective confidence. An AVA is in its own right a complex software-intensive system with no "rights or wrongs", posing challenges in the verification as with any other complex system. One cannot guarantee the agent's correctness. Estimating quantitative reliability of the agent is as difficult as it is with humans. However, a methodology for verifying an AVA should address the categories of the biases described earlier. A report on four different frameworks for verifying and auditing algorithms, and AI-based systems is presented in the Appendix. This may serve as a starting point for an *etic* verification of *Emic*-AVAs.

11. CONCLUSION

In the maritime industry, classification societies play a key role in ensuring the safe operation of vessels through their class rules and the enforcement of these. Historically, the verification performed by the classification societies has focused on the physical (i.e. mechanical) condition of components that are important for vessel integrity, such as hull strength and other hydrodynamic and mechanical properties. Introducing more and more complex software-intensive systems, the risk picture changes, and therefore the verification effort should shift towards the assurance of these systems. However, these systems are becoming so complex (e.g. autonomous navigation systems), that the traditional calendar-based surveys in operation may not generate the body of evidence

needed to ensure adequate confidence and trust. These kinds of systems may, for different reasons, require an algorithm-based verification agent to generate a body of evidence with adequate capability and quality. The question, however, is how trustworthy the evidence generated by such an agent is. This paper has argued that in order for an algorithm-based verification agent to generate a trustworthy body of evidence, this agent must be developed by another party than the (safety-critical) system developer, forming an *Etic* Algorithm-based Verification Agent (*Etic*-AVA). An alternative is that if the system developer also develops the verification agent, forming an *Emic* Algorithm-based Verification Agent (*Emic*-AVA), the latter must be assured by an independent party, such as the classification society, to remove/minimize/balance shared biases (between the AVA and the target system) for the generated body of evidence to become as objective as possible and thereby sufficient trustworthy.

ACKNOWLEDGEMENT

I would like to thank Professor Torbjørn Skramstad who is a colleague at DNV GL for his expert advice and support during the review process of this paper. The technical precision, as well as grammar improved considerately as a result of his input.

REFERENCES

Axtell, G. (2016). *Objectivity (Key concepts in philosophy) [Kindle].* Cambridge, UK: Polity Press. Retrieved from http://www.amazon.com

Daniel Kahneman. (n.d.). Retrieved 11 29, 2018, from Wikipedia: https://en.wikipedia.org/w/index.php?title=Daniel_Kahneman&gettingStartedReturn=true

Diakopoulos, N. (2014). *Algorithmic Accountability Reporting: On the investigation of black-boxes*. Retrieved from Tow Center for Digital Journalism: https://towcenter.org/research/algorithmic-accountability-on-the-investigation-of-black-boxes-2/

Diakopoulos, N. (2015). Algorithmic Accountability. *Digital Journalism, 3*(3), 398-415. doi:10.1080/21670811.2014.976411

Diakopoulos, N. (2016). Accountability in Algorithmic Decision-making. *acmqueue, 13*(9).

DNV GL. (2018a, July). RULES FOR CLASSIFICATION, Ships, Part 1 General regulations, Chapter 1 General regulations. Retrieved from https://rules.dnvgl.com/servicedocuments/dnvgl/#!/home

DNV GL. (2018b, July). RULES FOR CLASSIFICATION, Ships, Part 1 General regulations, Chapter 1 Class notations. Retrieved from https://rules.dnvgl.com/servicedocuments/dnvgl/#!/home

DNV GL. (2018c, July). RULES FOR CLASSIFICATION, Ships, Part 1 General regulations, Chapter 3 Documentation and certification requirements, general. Retrieved from https://rules.dnvgl.com/servicedocuments/dnvgl/#!/home

DNV GL. (2018d, July). RULES FOR CLASSIFICATION, Ships, Part 6 Additional class notations, Chapter 3 Navigation, manoeuvring and position keeping. Retrieved from https://rules.dnvgl.com/servicedocuments/dnvgl/#!/home

DNV GL. (2018e, September). CLASS GUIDELINE, Autonomous and remotely operated ships (DNVGL-CG-0264). Retrieved from https://rules.dnvgl.com/servicedocuments/dnvgl/#!/home

Douthwaite, M., & Kelly, T. (2018). Safety-Critical Software and Safety-Critical Artificial Intelligence: Integrating New Practices and New Safety Concerns for AI systems. *Proceedings of the Twenty-Sixth Safety-Critical Systems Symposium* (pp. 95-110). York, UK: Safety-Critical Systems Club.

Friedman, B., & Nissenbaum, H. (1996, July). Biases in computer systems. *ACM Transactions on Information Systems, 14*(3), 330-347.

Hawkins, R., & Kelly, T. (2011). A Structured Approach to Selecting and Justifying Software Evidence. *5th IET International Conference on System Safety* (pp. 1-6). Manchester, UK: IET. doi:10.1049/cp.2010.0825

Hinnosaar, M., & Hinnosaar, T. (2012, August 31). *Authority Bias.* Retrieved from Academia: https://www.academia.edu/2108445/Authority_Bias

Institute of Electrical and Electronics Engineers [IEEE]. (2016). Standard for System and Software Verification and Validation (IEEE Standard 1012). Retrieved from https://www.standard.no/en/

International Electrotechnical Commission [IEC]. (2010). Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements (IEC 61508-3). Retrieved from https://www.standard.no/en/

International Electrotechnical Commission [IEC]. (2014). Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations (IEC 61508-4). Retrieved from https://www.standard.no/en/

International Maritime Organization (IMO). (2017). GUIDELINES FOR VESSELS AND UNITS WITH DYNAMIC POSITIONING (DP) SYSTEMS. *MSC.1/Circ.1580*. Retrieved from https://vp.imo.org/

International Maritime Organization [IMO]. (1977). CONVENTION ON THE INTERNATIONAL REGULATIONS FOR PREVENTING COLLISIONS AT SEA. Rule 15 & 16 & 17. Retrieved from https://vp.imo.org

International Organization for Standardization [ISO]. (2013). Software and systems engineering Software testing - Part 1: Concepts and definitions (ISO/IEC/IEEE 29119-1). Retrieved from https://www.standard.no/en/

Kahneman, D. (2011). *Thinking Fast and slow.* London: Penguin Group.

Kitchin, R. (2017). Thinking Critically about and research algorithms. *Information, Communication & Society, 20*(1), 14-29.

Leveson, N. (2011). *Engineering a Safer World, Systems Thinking Applied to Safety.* Cambridge, Massachusetts: MIT Press.

Lustig , C., & Nardi, B. (2015). Algorithmic Authority: The Case of Bitcoin. *48th Hawaii International Conference on System Sciences* (pp. 743-752). Kauai, HI, USA: IEEE. doi:10.1109/HICSS.2015.95

Ministry of Defence. (2007). Safety Management Requirements for Defence Systems (Defence Standard 00-56 Issue 4, Part 1). Retrieved from https://www.skybrary.aero/bookshelf/books/344.pdf

Pasquale, F. (2014). The emperor's new codes: Reputation and search algorithms in the finance sector. *Draft for discussion at the NYU 'Governing Algorithms' conference.*

RTCA. (2011, December 13). Software Tool Qualification Considerations (RTCA DO-330). RTCA Inc. Retrieved from https://www.rtca.org/

Sandvig, C., Hamilton, K., Karahalios, K., & Langbort, C. (2014). Auditing Algorithms, Research Methods for detecting Discrimination on Internet Platforms. *Data and Discrimination: Converting Critical Concerns into Productive Inquiry, a preconference at the 64th Annual Meeting of the International Communication Association.* Seattle. Retrieved from http://www-personal.umich.edu/~csandvig/research/

Ship Safety and Security Act. (2007, February 16). *Act of 16 February 2007 No. 9 relating to ship safety and security, Chapter 7 Supervision, § 41 Supervisory authority*. Retrieved from Lovdata: https://www.sdir.no/contentassets/a7a1a5cc4998405286e99c6fbccc5c8a/ship-safety-and-security-act.pdf?t=1543320287463

Taylor, B. J., Darrah, M. A., Pullum, L. L., Ammar, K., Smith, J. T., Skias, S. T., . . . Fuller, E. J. (2006). *Methods and Procedures for the Verification and Validation of Artificial Neural Networks.* (B. J. Taylor, Ed.) New York: Springer.

Vartdal, B. J., Skjong, R., & St.Clair, A. L. (2018). *Remote-Controlled and autonomous ships in the maritime industry.* Høvik, Norway: DNV GL.

APPENDIX: REPORT ON VERIFYING AND AUDITING ALGORITHMS

This appendix reports on four different frameworks for verifying and auditing algorithms, and AI-based systems. The four approaches describe viewpoints, activities, and key questions that may reveal built-in biases, and serve as a starting point for *etic* verification (possibly performed by a classification society) of *Emic* Algorithm-based Verification Agents (*Emic*-AVAs). The items listed in each framework below are not complete, but they offer a basic understanding of the respective approaches. More concrete methods for the independent verification and validation of Artificial Neural Networks used in safety-critical applications are found in (Taylor, et al., 2006).

Framework 1
(Diakopoulos N. , 2016) suggests "...*five broad categories of information that we might consider disclosing: human involvement, data, the model, inferencing, and algorithmic presence*.":

- **Human involvement**: "*Might contain goal, purpose, and intent of the algorithm*"
- **Data**: "*Its accuracy, completeness, and uncertainty, as well as its timeliness, assumptions and other limitations. How was it defined, collected, transformed, vetted, and edited (either automatically or by human hands)?*"
- **The model**: "*What the model actually uses as input. What features or variables are used in the algorithm? Often those features are weighted: What are those weights? We want to know the rationale for weightings and the design process for considering alternative models or model comparisons. What are the assumptions (statistical or otherwise) behind the model, and where did those assumptions arise?*"
- **Inferencing**: "*What is the margin of error? What is the accuracy rate, and how many false positives versus false negatives are there? What kinds of steps are taken to remediate known errors? Are errors a result of human involvement, data inputs, or the algorithm itself?*"
- **Algorithmic presence**: "...*disclose if and when an algorithm is being employed at all"*

Framework 2
(Douthwaite & Kelly, 2018) suggest six different viewpoints "...*to provide a framework within which the full range of considerations associated with AI-intensive systems can be captured and analysed.*":

- **Model viewpoint**: "*Model structure, parameterisation and resulting model behaviours/properties, and all associated modelling assumptions, decisions and development activities.*"
- **Data viepoint**: "*All data acquisition, processing and storage concerns. This includes knowledge engineering and expert elicitation activities (if relevant), and the quality and integrity of any resultant data artefacts*."
- **Computational viewpoint**: "*The properties of all algorithms used for learning and reasoning tasks within the system, their selection process, and the associated assumptions and design decisions*"
- **Operational viewpoint**: "*The evolution and maintenance of the system after deployment*"
- **Technology viewpoint**: "*The necessity, properties, constraints and assumptions of modelling frameworks used in the system*"
- **Implementation viewpoint**: "*All 'conventional' software and hardware engineering concerns, including 'normal' function allocation, requirements and associated verification and validation activities.*"

Framework 3
(Diakopoulos N. , 2014) lists some key questions that may serve as a "...*basis for beginning to investigate an algorithm*"

- *What is the basis for a prioritization decision?*
- *What are the criteria built into a ranking, classification, or association...?*
- *What are the limits to measuring and operationalizing the criteria used by the algorithm?*
- *What are the limits of an algorithm and when is it known to break down or fail?*
- *What are thresholds used in classification decisions?*
- *What kind of uncertainty is there in the classifier?*
- *What are the potential biases of the training data used in a classifying algorithm?*
- *How has the algorithm evolved with that data?*
- *What types of parameters or data were used to initiate the algorithm?*
- *How are the semantics and similarity functions defined in an association algorithm?*
- *Are there some pieces of information that are differentially over-emphasized or excluded by the algorithm?*

Framework 4

(Kitchin, 2017) lists "...*six methodological approaches for researching algorithms that I believe present the most promise for shedding light on the nature and workings of algorithms, their embedding in socio-technical systems, their effects and power, and dealing with and overcoming the difficulties of gaining access to source code*."

- **Examining pseudo-code/source code**: "*Perhaps the most obvious way to try and understand an algorithm is to examine its pseudo-code (how a task or puzzle is translated into a model or recipe) and/or its construction in source code.", "...it requires that the researcher is both an expert in the domain to which the algorithm refers and possesses sufficient skill and knowledge as a programmer...*"
- **Reflexively producing code**: "*...rather than studying an algorithm created by others, a researcher reflects on and critically interrogates their own experiences of translating and formulating an algorithm.*"
- **Reverse engineering**: "*reverse engineering is the process of articulating the specifications of a system through a rigorous examination drawing on domain knowledge, observation, and deduction to unearth a model of how that system works*."
- **Interviewing designers or conducting an ethnography of a coding team**: "*Interviewing designers and coders, or conducting an ethnography of a coding team, provides a means of uncovering the story behind the production of an algorithm and to interrogate its purpose and assumptions*."
- **Unpacking the full socio-technical assemblage of algorithms**: "*...algorithms are not formulated or do not work in isolation, but form part of a technological stack that includes infrastructure/hardware, code platforms, data and interfaces, and are framed and conditions by forms of knowledge, legalities, governmentalities, institutions, marketplaces, finance and so on. A wider understanding of algorithms then requires their full socio-technical assemblage to be examined, including an analysis of the reasons for subjecting the system to the logic of computation in the first place*".
- **Examining how algorithms do work in the world**: "*Given that algorithms do active work in the world it is important not only to focus on the construction of algorithms, and their production within a wider assemblage, but also to examine how they are deployed within different domains to perform a multitude of tasks." "...algorithms perform in context – in collaboration with data, technologies, people, etc. under varying conditions – and therefore their effects unfold in contingent and relational ways, producing localised and situated outcomes.*"