Multi-Agent Model of Centrally Coordinated Compliance Inspections (ICARUS)[1]

Slaven Smojver[2]

Compliance inspection is a detailed examination procedure for determining compliance of a particular person or organisation with the given set of provisions (in regulations, standards, business rules, etc.). The optimal selection of persons or organisations and provisions for inspection is one of the key compliance inspection challenges.

This paper describes a multi-agent compliance inspection model (ICARUS) applicable to environments where an inspection agency, via centrally coordinated inspections, examines compliance of organizations which must comply with multiple provisions. The contents of the paper are adapted from the author's doctoral dissertation and translated in English.

The developed model was implemented in a computer simulation in the NetLogo environment. The paper describers, in details, calibration, verification, validation and sensitivity analysis of the model. The developed model and simulation can be applied to a number of inspection problems. The simulation generated data was used to test the hypotheses and confirm that the compliance inspection strategy in which inspection effort is proportionate to the costs of compliance with provisions achieves significantly fewer non-compliances in the system than random and cyclical inspection strategies. The hypothesis that a smaller number of non-compliances is achieved by applying the inspector's (Stackelberg) leadership strategy to conduct inspections was rejected.

The results of the research indicate possible improvements in the conduct of compliance inspections, which could lead to a reduction in the overall non-compliance and is applicable to various areas such as financial sector, environmental protection, occupational health and safety, etc.

Keywords: Inspection, compliance, violation, multi-agent, agent-based model, ABM, MABM, simulation, resources, ICARUS, NetLogo.

JEL Classification: C63, C72, C73, D81, D83, K42, K23, K32

---

[1] This paper contains research results from the doctoral dissertation of the author. Parts of the dissertation are adapted and translated from Croatian to English. The research was not previously published and the aim of this paper is to make research results available to a wider audience. The complete dissertation is available here: https://repozitorij.foi.unizg.hr/islandora/object/foi:3558 (only in Croatian). The dissertation was completed under mentorship of prof. dr.sc. Neven Vrček.

[2] https://orcid.org/0000-0001-6669-0188

Contents

1. Introduction

Codification of desirable social norms but also of undesirable patterns of behaviour through laws and other types of regulations is one of the preconditions for the existence of human civilization. **Regulations** define the permissible and impermissible behaviour of individuals and organizations, including companies, non-profit organizations, etc. In addition to defining permissible and impermissible behaviour, regulations – in principle – prescribe penalties or sanctions for individuals and organizations that do not adhere to the set limits. The main goal of penalties (punishment) is to sanction unwanted behaviour, but the indirect – and perhaps more important – goal is to discourage or deter the entire population from unwanted behaviour.

The general reason for the adoption of regulations related to organizations is the failure of markets and society to independently (i.e. without external influence) achieve socially desirable results. In doing so, regulations can prevent or limit negative consequences (monopolies and natural monopolies, excessive earnings, unwanted externalities, inadequate information, restriction of competition, moral hazard) or encourage desirable results (continuity and availability of necessary services, provision of public goods and services, allocation of limited resources, rationalization and coordination, and planning) [1].

The application of regulations usually increases costs for a regulated entity, both direct (for example, implementation cost) and indirect (for example, reduction of earnings due to restrictions on service provision, greater competition, increase in production costs, reduction of information asymmetry, etc.). Therefore, regulated entities often try to avoid compliance with regulations (all regulations, or only those whose implementation is expensive), but they also want to avoid the penalty for non-compliance. If the mechanisms that will detect and punish non-compliance with regulations are not established or the penalty (material and social) is less than the cost of compliance, entities will decide to violate the regulations and the desired results will not be achieved. This is particularly important in the context of regulations that apply to enterprises. It is commonly considered that the violation of regulations by enterprises is rational and situationally opportunistic; enterprises violate regulations intentionally and make decisions based on weighing expected costs and expected economic gains [2, p. 36].

Regulations violation by organizations is not trivial and can cause significant negative consequences for the society. For example, violations of regulations in the field of so-called "White-collar crime" is on the rise and already in 2006 in the United States (USA) caused 18 times higher costs than traditional forms of property crime [3, p. 550]. Analyses of the financial crisis of 2007-2009, which had great negative consequences for the whole world, indicate [4] that among the fundamental reasons for the crisis is "moral hazard", i.e. the tendency of financial institutions' management structures to take unreasonably high risks under the assumption that they will not have to bear the “price” of their actions (i.e., unpunished violation).

The assessment whether an **entity** (organization or individual) is compliant with regulations often is not trivial. Namely, although in some environments and situations compliance can be checked easily and at negligible cost (for example, by checking relevant publicly available information), specialized knowledge and focused efforts are often required to precisely determine whether an entity is compliant with rules or regulations.

This process of in-depth analysis and verification, which may also mean an exhaustive official verification, is called an **inspection** [5, p. 608]. **Compliance inspection** is the process of detailed verification with the aim of determining whether a particular person or organization complies with a given set of provisions (contained in regulations, standards, policies, etc.). In addition to the direct consequence of detection of non-compliance (e.g. punishment), knowledge of the existence and possibility of inspection acts on the supervised entities as a deterrent mechanism [6]. The empirical research shows that even in environments with generally low non-compliance, the knowledge of obligees that they certainly will not be subject to inspection leads to a high level of non-compliance [7]. On the other hand, there is empirical evidence that the introduction of punishment or the possibility of punishment leads to a reduction of non-compliance [8].

Inspections require resources. Therefore, from the set of all possible combinations of persons or organizations and provisions, only a small subset is usually selected for compliance inspection (**inspection sample**) [1]. The method of determining the inspection sample depends on the capacity to conduct inspections, information available to the inspector, possible consequences of violations, cost of inspections, etc. To optimize limited resources, inspections are often centrally coordinated by inspection agencies. **Inspection agencies** are organizations that have been tasked with verifying compliance with a regulation or a group of regulations. Such organizations may be limited in their activities in various ways: by their ability to perform only reactive inspections, restricted budget and number of employees, inability to conduct objective and fair inspections, and inadequate range of possible penalties [2, p. 97].

Given their importance for society and the need to achieve maximum efficiency, compliance inspections have been the subject of scientific research in various scientific disciplines and have been considered through different paradigms. Furthermore, given the legal and ethical constraints associated with conducting applicable experiments outside of laboratory conditions, compliance inspections are often analysed using different models.

Existing models for inspection sample selection and inspection conduct have different approaches, but frequent limitations are overly idealized conditions (one extreme) or a very narrow scope (other extreme) [9]. The conducted research generally [9] ignores the information that could, in accordance with the so-called economic model of crime [10], be useful in selecting an inspection sample and strategy, such as the different costs (resource requirements) of fulfilling different provisions (i.e., the rules contained in regulations). Likewise, existing models do not model situations in which centrally coordinated compliance inspections select combinations of entities and rules that will be inspected.

This paper analyses the problem of optimal selection of organizations and provisions for inspection in an environment in which numerous entities (organizations) must adhere to numerous provisions (rules). The compliance of these organizations with the given provisions is validated via centrally coordinated compliance inspections. The described environment is typical of many real-life situations such as the inspection of compliance in the banking sector, compliance with environmental regulations, compliance with the occupational safety regulations, etc.

2. Research objectives, questions and hypotheses

The research had the following goals:

C.1 Develop a multi-agent model of centrally-coordinated compliance inspection in a system with multiple supervised organizations, each of which must comply with multiple provisions, based on insights from theoretical research and empirical data.

C.2 Through the simulation of the developed model, investigate whether and under what conditions is it possible to reduce the overall level of non-compliance in the system, by applying different strategies for selection of provisions that will be inspected.

Based on the set research goals, the following research questions were constructed:

I.1 Can a multi-agent model of centrally-coordinated compliance inspection in a system with multiple supervised organizations, each of which must comply with several provisions, developed on the basis of theoretical research and empirical data, faithfully reproduce patterns identified via analysis of secondary data on compliance inspections with environmental, banking and occupational safety regulations?

I.2 Will the developed multi-agent model of compliance inspection show that inspection strategies which rely on the assumption of different frequency of violations of different provisions, and based on the economic model of crime, are more effective than frequently used inspection strategies?

I.3 Will the knowledge of the supervised entities about the application of the inspection strategy, in which the inspection effort is proportionate to the assessed costs of compliance with different provisions affect the effectiveness of the implementation of that strategy?

Model validation presents the greatest challenge associated with the application of multi-agent models[3]. Therefore, in this paper, great attention is paid to the appropriate verification and validation of the model, since it is crucial to investigate whether the model can adequately reproduce the patterns identified in the empirical data on compliance inspections. The developed model of compliance inspection, although not restrained to only one specific area, is limited to the areas of inspection of compliance with environmental, banking and work safety regulations. These areas were selected because of certain similarities in organization and conduct of compliance inspections, as well as because of the availability of empirical data on compliance inspections.

The second research question further elaborates the second goal of this paper, namely to examine whether inspection strategies that take into account (or assume) that the costs of achieving and maintaining compliance are different for different rules are more successful than other, frequently used inspection strategies.

The last research question examines whether the entity's knowledge that an inspection agency conducts inspections based on the assumption that the costs of achieving and maintaining compliance are different for different rules will affect the effectiveness of that strategy. This knowledge also influences the decisions of the entities on violation or compliance with individual rules, which will ultimately likely have an impact on the effectiveness of the overall inspection strategy.

[3] The basis for this statement is explained in depth in the doctoral dissertation of the author.

The following hypotheses were derived from the research questions:

H.1 By selecting the provisions for compliance inspection relative to the costs of compliance with those provisions, the simulation results in a lower level of noncompliance, compared to the use of random selection.

H.2 By selecting the provisions for compliance inspection relative to the costs of compliance with those provisions, the simulation results in a lower level of noncompliance, compared to the use of cyclic selection.

H.3 By applying the inspector's (Stackelberg) leadership to inspections in which the selection of provisions for compliance inspection is performed relatively to the costs of compliance with those provisions, the simulation results in a lower level of noncompliance, with respect to the inspection without the inspector's leadership.

Hypotheses arise from the research questions and allow objective verification. Hypotheses were tested on simulation data. Hypotheses 1 and 2 arise from the first research question. Random and cyclical selection strategies have been identified as inspection strategies that are frequently used in practice. The cyclical selection strategy entails the inspection of the entire population (all relevant entities) and part or all of the rules in a given period (cycle). The random selection strategy randomly selects the entities to be inspected and in these entities conducts an inspection of the compliance with all the rules or with a randomly selected (sub)set of rules. Knowledge by an entity that the inspection agency uses a particular inspection strategy can be formally incorporated into the model by applying the concept of Stackelberg leadership.

3. Literature review

The chapter provides an overview of the agent-based models (ABM) applied to inspections, with an emphasis on compliance inspection. In addition, the chapter provides an overview of the conclusions of empirical research on compliance inspection that are relevant to this research.[4]

3.1. Agent-based models of inspection

(Multi)agent models (MABM) of inspection often encompass assumptions and models from the game theory, the rational choice theory (RCT) and assumption of limited rationality of agents.

Rauhut and co-authors discussed the application of multi-agent simulation to the problem of inspection in several articles [11][12][13]. Multi-agent simulation is used because the considered analytical model is too complex to solve [11]. Rauhut and Jud [13] present a multi-agent model of inspection in which learning is modelled as a fictitious play, where agents in the simpler variant take into account only the last move, and in the more complex variant all the moves made so far. In doing so, the historical experiences of the agent are discounted over time, i.e. more recent experiences have a greater impact on the behaviour of the agent. The model, however, is based on the assumption that the results of all inspections conducted so far are known to all agents. Rauhut and Juncker [11] in a multi-agent inspection model consider a learning strategy based on the experience of agents. Agents in described models are limited in rational decision-making, with limited rationality modelled in two ways: as "bounded learning" – agents err in learning but perform flawlessly in decision-making, or as "bounded decision-making" – agents err in calculation when making decisions. The results of the experiments conducted by Rauhut and Juncker support the "bounded decision-making" model.

Tax inspections are often analysed via multi-agent models and simulation [14][15][16][17][18]. These models have different assumptions, complexity, validation methods and were developed in different environments. However, they all start from simple models which are then adapted with increasingly complex assumptions and patterns of behaviour. The key contribution of these models is enabling analysis of impact of various parameters on the increase or decrease of tax fraud. The models are mainly based on available, relatively detailed empirical data on tax returns and tax fraud. The basic model of agent behaviour is based on the rational choice theory (RCT) [7].

Antunes and co-authors [19] present 4 models of tax compliance inspection, starting with the simplest rational choice model (standard theory), and progressively add complexity by including agent individuality (and different risk appetite), adaptability of strategy depending on inspection history and sociability of agents. Bloomquist [17][20] describes several multi-agent tax inspection models, using Markov's processes to model taxpayer's decision-making. These models rely on the rational choice theory, but also on the assumption that a significant number of people are "pathologically honest" (25% in the model), who are always in line with regulations [17]. Bloomquist validates the model on an extensive set of U.S. tax compliance and inspection data.

Asselt and co-authors [21] applied multi-agent simulation to the analysis of compliance of agricultural producers in the Netherlands with the relevant regulations. In the model, agents (farmers) take into account their own assessment of the risks and benefits of breaches or compliance, but are also affected by the environment, with agents differing in their attitude towards risk-taking. Verwaart and Valeeva [22] analyse possible communication strategies aimed at increasing food producers ’compliance with food safety

[4] Extensive description of the theoretical background of this research, including the application of rational choice theory, game theory and agent-based modelling as well as the overview of the limitations of such approaches are described in the doctoral dissertation of the author and are not influenced in this paper.

regulations using multi-agent simulation. McPhee-Knowles [23] applies multi-agent simulation to food safety inspection and considers the optimal number of inspectors, given the applied inspection strategy.

Multi-agent modelling is also applied to simulation of compliance inspection with environmental regulations. Liu and Ye [24] model different patterns of behaviour of Chinese companies in achieving compliance with environmental regulations.

Malleson independently [25] and with co-authors [26] applied multi-agent simulation to modelling criminal activity and decision-making of criminals, and the author of this paper applied multi-agent simulation to compliance inspections of banks and on-site banking supervision [27][28].

Most models mentioned in this chapter are based (in design, calibration or validation) on secondary empirical data relevant to the narrow scope of the model and generally do not consider environments in which the inspected entity must comply with multiple rules (or types of rules) and in which inspector may select a sample of rules for inspection.

3.2. Empirical research on compliance inspections

Publicly available empirical data on compliance inspections is relatively extensive. The primary source of data are publications by inspection agencies and other regulatory and supervisory bodies on their approach to inspections and, above all, on the results of the inspections. However, published data rarely describes in depth inspectors' methodologies and it is usually impossible to compare them directly due to differences in scope, legal framework, inspection practices, cultural differences and the like. In addition, inspection agencies and supervisory bodies often deliberately do not publish part of the information. Namely, the data that could reveal the methods of inspectors, enable the supervised entities to avoid inspections and consequently reduce the efficiency of inspections.

The **selection method** of the entities to be inspected as well as the selection of the inspection area depends on a number of criteria, including the size of the organization, business complexity, complexity of the inspection area, implemented controls, etc. [29]. Furthermore, the method for selecting entities and areas to be inspected usually relies on the cycle and/or randomness of inspections.

The **inspection cycle** determines the period of time during which all relevant entities or all entities in a sub-group will be inspected, at least once. Many inspection agencies have an externally or internally prescribed inspection cycle. E.g., the guidelines on the compliance inspection with the U.S. air and water quality regulations recommend that inspections of large pollutants (companies) be conducted every two years and inspections of medium-sized pollutants every 5 years [30]. Supervisors of credit institutions in the European Union have to cover all credit institutions over a period of time, with the length of the cycle depending on the systemic importance of the credit institution [31]. Systemically important institutions have to be supervised every year, and small institutions without systemic importance need to be supervised at least once every 3 years. Inspections of compliance with the occupational safety regulations in the United States conducted by the OSHA (Occupational Safety and Health Administration) should assess each regulated entity at least once every two years [32]. The existence of an inspection cycle is sometimes indirectly visible – for example, entities that have been recently examined for compliance with the US environmental regulations are less likely to be subject to inspection again [33]. Quantitative data on bank inspections (conducted as a part of bank supervision) show large differences, from country to country, indicating different practices. Delis and Staikouras [34] compared data on banking supervision from 17 countries during 9 years and found large variations in the average number of supervisions per institution (from 0.14 to 9.86). These data suggest that there are also large differences in the inspection cycle from country to country.

The empirical data related to environmental regulations compliance, as well as studies of the tax inspections show that a certain level of **random** inspections is necessary to maximize compliance [30]. Randomness ensures that the inspections could cover the entire domain, which is necessary to discourage violations [17]. Random inspections are sometimes the dominant, or even the only, way to select entities for inspection – for example, the IRS (Internal Revenue Service) in the United States has conducted only random inspections of tax compliance for 25 years [7]. The theoretical basis for conducting inspections by a random selection strategy can also be found in the inspection game[5] [35, p. 10].

The **intensity and methods of inspections** may vary from area to area and even from case to case. For example, related to the compliance inspections with the environmental regulations, low-intensity inspections may be based exclusively on visual inspection, medium-intensity inspections may include inspection of the functioning, maintenance and reporting by the inspected entity, while high-intensity inspections include extensive sampling and testing by inspectors [36]. In general, formal and structured inspections result in higher levels of compliance [37].

However, inspections are not the only supervisory mechanisms. Complementary – mostly signalling – mechanisms may exist which in turn may affect the overall level of compliance. For example, an inspection agency can obtain information on personal income tax from both the employee and the employer [7]. In such circumstances, it is possible to achieve a high level of overall compliance even with a very low level of inspection. For example, the IRS in the U.S. audits only 2% of households, but estimates that 91.7% of the personal income taxes are accurately reported [19]. On the other hand, the estimated level of tax evasion in the areas which are very difficult to inspect is 90% [7]. Bloomquist [17] states that tax evasion is 1% in circumstances where the inspector has a lot of secondary information on compliance, 8-11% in circumstances when such information is "enough" and up to 56% when such information is scarce or non-existent. Supervision of banks' operations also relies on extensive data collection and analysis, including analysis of the banks' financial statements, analysis of external auditors' reports [38] and internal control mechanisms [39], as well as the inspections by banking supervisors [40].

The mechanisms for selecting the entities that will be inspected may differ significantly even within the same country. Shimshack [36] shows that in the same period in the US state of North Carolina, compliance with air quality regulations was inspected in as many as 95% of regulated companies, while in the New York state only 10% of regulated companies were inspected. The same study cites similar ranges for inspection of compliance with water quality regulations.

When selecting **area or provisions** that will be subject to **inspection**, supervisory agencies can be guided by various data.

May and Winter [41] state that the literature on repressive mechanisms and measures is relatively consistent in concluding that oversight needs to focus on the categories of regulations with which a higher level of violation has historically been found. Furthermore, the authors conclude that, assuming limited resources for supervision, monitoring or inspection, a focus on minor offenses will reduce the overall level of compliance.

Heyes and Rickman [42] conclude from empirical data that the US EPA (United States Environmental Protection Agency) tolerates higher levels of non-compliance in some areas, with the presumed intention of achieving higher levels of compliance in other areas or higher overall levels of compliance. That is, greater tolerance of violations in certain areas is a strategic choice of the US EPA, not a coincidence or collusion with violators.

Slemrod [7] states that the risk culture of an organization is related to the level of (non) compliance, since the level of non-compliance with tax regulations is higher in organizations where management is rewarded with higher bonuses (i.e. where risk-taking is rewarded). Therefore, a possible recommendation could be to focus inspections on such organizations.

The selection of the areas of inspection in banks, within the banking supervision process, takes into account several factors, including the size of the bank, the complexity of the area, information on the implemented controls, etc. [29].

The **overall level of compliance** is difficult to assess, given: the interests of the regulated entities; usually a limited dataset (sample) that is assessed; and difficult determination of a representative sample, given the different characteristics of the regulated entities. For example, Magat and Viscusi [43] state that the average compliance of the paper industry in the United States with environmental regulations is 75%. The

---

5 The inspection game is application of the game theory to the inspection problem. A review of the inspection game literature can be found in [35][102][103][99].

official data estimate compliance with US environmental regulations at 86%, but also warn that these data are probably overly optimistic [120]. Gray and Shimshack [30] state that the overall level of compliance with the air quality regulations is 38% and the overall level of compliance with the water quality regulations is 75%.

In 2001, the total uncollected tax in the United States was estimated at 16.3% of the total tax collected [7]. However, non-compliance with tax regulations varies significantly, depending on the area of application, from 8% [19] to 90% [7]. The level of compliance also varies from country to country. Slemrod [7] states that the estimated total uncollected tax in New Zealand is only 5.1%, while in Peru it is as high as 68.2%.

Delis and Staikouras [34] compared banking supervision data in 17 countries in 9 years and showed large variations in the average number of penalties per supervised institution (from 0.02 to 4.06), which suggests possible large differences in the overall level of compliance or in the supervisory approach.

The analyzed empirical research in the vast majority of cases supports the conclusion that conducting **compliance inspections increases** regulatory **compliance**.

Gray and Shimshack, in a meta-study [30] that included data on compliance inspections with air and water quality regulations, and occupational safety regulations in the United States, concluded that there is a clear link between compliance inspections and subsequent lower levels of violations. The meta-study shows that after the compliance inspection, organizations had a 10% higher compliance with air quality regulations, and a 20% lower level of water pollution.

Corman and Mocan [44] based on the data on crime in the New York between 1970 and 1996 conclude that there is a link between police resources and crime levels. The increase in the number of police officers, as well as their greater focus on more serious crimes, is associated with a decrease in the total number of criminal offenses.

Empirical studies related to compliance with the environmental protection regulations show, relatively consistently, that inspections have a positive impact on compliance, i.e. that inspections, sanctions, as well as a greater threat of inspections or sanctions affect the behaviour of organizations and their further (greater) compliance [36][120]. A higher level of tax inspection also results in a higher level of compliance, with a general deterrence effect[6] [45].

Higher levels of inspection of compliance with tax regulations reduce the level of non-compliance – increasing the level of inspection by one percentage point reduces tax evasion by 0.5% [45]. However, taxpayers' perceptions of the likelihood of an inspection of compliance with tax regulations have a much greater impact on compliance than the actual likelihood of an inspection [45]. Bloomquist [17] states that the perceived probability of a tax inspection is as much as nine times higher than the actual one (9% compared to 1%).

Studies of inspections of compliance with environmental regulations also emphasize that the threat of inspection, i.e. the perceived higher probability of inspection, increases compliance [120]. However, detection errors reduce the deterrent effect [36].

The data on the direction of the effect of the inspection of compliance with occupational safety regulations are unambiguous, but the data on the size of the effect are not completely consistent. Analyses of empirical data show that inspections in that area do not have as great an impact on compliance as inspections of compliance with environmental regulations [45]. Scholz and Gray [46] show that increasing the level of occupational safety inspection by 10% leads to 1% fewer injuries at work. Gray and Mendeloff [47] conclude that the deterrent effect of inspections falls over time, from 19% to 1%. A controlled experiment conducted by Levine et al [48] shows that inspections had the following impact (compared to the control group): 9% fewer work-related injuries and 26% lower costs as a result of these injuries.

The impact of inspections on compliance is not necessarily linear. Data on compliance inspections with environmental regulations [32][30], tax inspections [17, p. 25], occupational safety [47] and banking supervision [29] show a declining impact of inspections on compliance. That is, after the first inspection there is a significant drop in non-compliance, and each subsequent inspection has less and less impact on non-compliance.

Ko and co-authors [32] also conclude that extending the period without inspection increases the number of non-compliances.

---

[6] General deterrence considers the impact that law enforcement action on one entity had on other relevant entities.

Empirical research indicates that **increasing the likelihood and/or quantity of punishment** has the effect of **increasing regulatory compliance**.

Grogger [49] shows, based on a sample of fifteen thousand convicts in the US state of California, that inevitability or a higher perceived probability of punishment by a (potential) offender has a significant deterrent effect on the criminal offense. According to the same study, the severity of punishment also has a deterrent effect, but less than the perceived probability of punishment. Data on banking supervision show the existence of a linear negative relationship between penalties and the risks to which banks are exposed [160].

Empirical data on inspections of compliance with occupational safety regulations show that inspections that do not penalize non-compliance have little or no deterrent effect [47]. Different types of punishment have different deterrents. Descending by their effect are: imprisonment, high fine, conviction, court proceedings, identification of responsible persons as violators [50]. Additionally, empirical data shows that significant fines have a high deterrent effect, while informal fines without a financial effect on the offender do not affect compliance [45]. On the other hand, the effectiveness of financial penalties as a penalty mechanism may be questionable due to the difficulties in enforcing them, i.e. the low level of collection [51].

In line with the expectations of the economic model of crime[7], in environments with a low probability of materialization of punishment, the deterrent effect of punishment is low [37]. In such cases, the cost of compliance can have a significantly greater impact on the overall level of compliance than the fear of punishment [37].

On the other hand, underestimating the probability of punishment leads to a higher level of non-compliance – Slemrod [7] shows that organizations that promote a culture of risk-taking, which is evident from, for example, higher bonuses, have a higher level of non-compliance with regulations.

Analysed literature suggests that **conducting inspections has a deterrent effect on violators**. General deterrence can have a significant impact – for example, according to Shimshack [36], 10-40% of organizations state that they perform changes related to compliance when they receive information that another organization has been penalized for non-compliance. General deterrence is also visible in laboratory experiments of tax evasion inspection, but a significant percentage of pro-social behaviour that is visible in laboratories is not visible in reality [17, p. 24].

In accordance with the assumptions of the inspection game, the threat of punishment must be credible. Therefore, a better understanding of the inspection relationship and the understanding that the actual probabilities of inspection and punishment are less than initially perceived, leads to a reduction in the deterrent effect of punishment [37].

Administrative penalties can reduce recidivism in companies (i.e., specific deterrence[8] is at work) and the perception of certainty of violation detection is higher if the company has already been in contact with the inspector [2, p. 97]. Executives' attitudes about the negative ethical and moral dimensions associated with non-compliance can have a significant deterrent effect. Executives who believe they will experience business success as a result of regulatory violations are the most likely violators [2, p. 151]. Thus Simpson [2, p. 152] concludes that the threat of punishment is necessary to control the behaviour of a part of the executive and as a reminder to other executives. Simpson [2, p. 42] also warns that the threat of formal sanctions has little impact on populations that are committed to breaking the law.

Analysed research indicates that there is a **negative correlation** between the **price (cost) of achieving compliance and the overall level of compliance**. Empirical data on compliance inspections in the paper industry indicate that entities which frequently violate the rules have a higher cost of compliance and are therefore less likely to respond to inspectors' activities [30]. That is, repressive measures will not have the desired effect. Capital investment needed to achieve compliance can also have a significant effect [43].

---

[7] In the Becker's economic model of crime [104], the (potential) offender (i.e., agent) attempts to maximize the value of its utility function.

[8] Specific deterrence considers the impact that law enforcement action had (only) on the penalized entity.

May and Winter [41] in their empirical research on compliance with environmental regulations in Denmark identify costs as the main reason for non-compliance. Furthermore, the cost of achieving compliance may have a greater impact on (non) compliance than the fear of inspection and punishment [37]. High compliance costs can also be a sign of a company's technological backwardness, and in such cases, violations committed by a particular organization can be a good indicator of future violations [33].

The data and conclusions presented in this chapter indicate the existence of some regularities in the impact of inspections, penalties, compliance costs and some other factors on the overall level of compliance, which are present in different countries, areas and industries. However, it is also evident that there are large differences in the inspection practices in the world, depending on the country, area of economic activity, characteristics of the regulated entities and inspection agencies, etc.

4. Description of the model and the simulation

The chapter contains a description of the developed multi-agent model of compliance inspection. In the model, the inspection agency, via centrally coordinated inspections, assesses compliance of a number of entities with a number of rules. The developed model is named ICARUS (Inspecting Compliance to mAny RUleS).

The chapter includes a conceptual description of the model and a description of the implementation of the model in a computer simulation. The description of the model and computer simulation is described in line with the ODD+D protocol [52][53].

ICARUS is based on earlier versions of the compliance inspection model published in [28] and [9].

4.1. Model overview

4.1.1. Purpose

The model displays a centrally-coordinated inspection of compliance in a system with multiple supervised organizations, with each organization having to comply with multiple (regulatory) provisions. The model faithfully reproduces the patterns identified in the secondary data on compliance inspections with environmental, banking and occupational safety regulations. In addition, the model and implemented computer simulation enable assessment whether inspection strategies that rely on the assumption of different frequency of violations of different provisions are more effective than frequently used inspection strategies, i.e. the model allows confirmation or rejection of the stated hypotheses. Finally, the model and the developed computer simulation enable the analysis of other inspection problems and contribute to the improvement of the implementation of compliance inspections in practice.

The main motivation for the model development was the fact that the inspection models developed so far focus almost entirely on environments in which entities must comply with a single rule. These models do not provide a credible depiction of highly regulated areas such as financial services sector or compliance with the environmental regulations. An additional motivation for the model development is the ability to test hypotheses that conducting inspections that rely on the assumption that compliance with some regulatory requirements is more expensive than compliance with other requirements may reduce the total number of violations in the system. That is, differences in the costs of compliance could be used as a signalling mechanism available to inspectors (although not with complete accuracy), based on which the accuracy of the inspections could be increased and – ultimately – the overall non-compliance could be reduced.

The motivation for using agent-based modelling on the inspection problem is the flexibility of such approach, the characteristics of agent-based modelling that correspond to the problem, but also legal and ethical limits related to conducting experiments and limitations of applying only analytical approach (such as game theory) to the problem.

4.1.2. Agents, states and processes

The model contains 3 types of agents: inspected entities, inspection agency and inspectors/inspections (hereinafter: inspections). The formal description of the prototype of the model presented in this chapter was published, in a limited form, in [9].

Let $\mathcal{E} = \{1, \dots, n\}$ be a set of n entities (agents or organizations), $n \in \mathbb{N}$, where each entity in $\mathcal{E}$ is obliged to comply with all the social norm (rules in regulations) contained in $\mathcal{O} = \{1, \dots, m\}$, $m \in \mathbb{N}$. Each entity from $\mathcal{E}$ in each discrete time interval t from the set $\mathcal{T} = \{1, \dots, \tau\}$, $\tau \in \mathbb{N}$ decides whether to comply with or violate each of the rules contained in $\mathcal{O}$.

The inspection agency $\mathcal{J}$ has the task of supervising the compliance of the entities with all the rules (provisions). At every moment $t \in \mathcal{T}$, $\mathcal{J}$ decides whether to inspect each of the possible combinations (pairs) of entities and rules. When an entity i decides whether to abide by or violate the rule j at t, it does not know whether $\mathcal{J}$ will inspect $\{i, j\}$ at t. Analogously, $\mathcal{J}$ does not know the current state of compliance when deciding which combination of entities and rules inspections will cover at t. $\mathcal{J}$ assesses compliance via inspections. Inspections are agents that can verify the compliance of any entity in $\mathcal{E}$ with one or more of rules from $\mathcal{O}$, directed by $\mathcal{J}$. At every moment $t \in \mathcal{T}$, one inspection can assess compliance of only one entity. After inspection of $\{i, j\}$ at t, it assesses whether the entity i was compliant with the rule j at the time interval t. The inspection immediately informs the agency $\mathcal{J}$ on the state of compliance of $\{i, j\}$ at t. Figure 4.1 displays the game-tree of this interaction (game) – that is, all possible outcomes – at the entity-rule level.

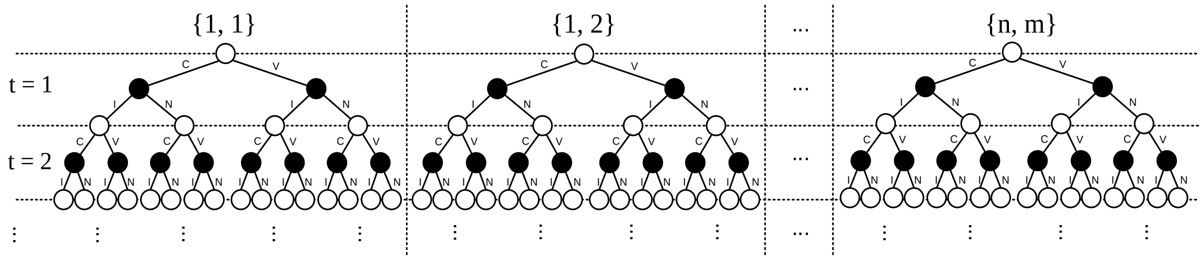

Figure 4.1 – Game tree (adapted according to [9, p. 284]**, fig. 2)**

If an inspection detects a violation of a rule, it will penalize the inspected entity with the prescribed penalty. Entities are aware that they will be penalized if the inspection identifies a violation of a rule. The entities remember the history of their decisions and the results of the inspections to which they have been subjected and will, on the basis of that knowledge and their individual characteristics, make decisions on compliance or violation of the rules. The interests of the entities and the inspection agency are directly opposed, in line with the postulates of the inspection game. Entities independently decide on compliance or violation, while the inspection agency implements a given inspection strategy. Inspections are non-independent agents that consistently carry out the directions of the inspection agency. Figure 4.2 displays the class diagram of the described model.

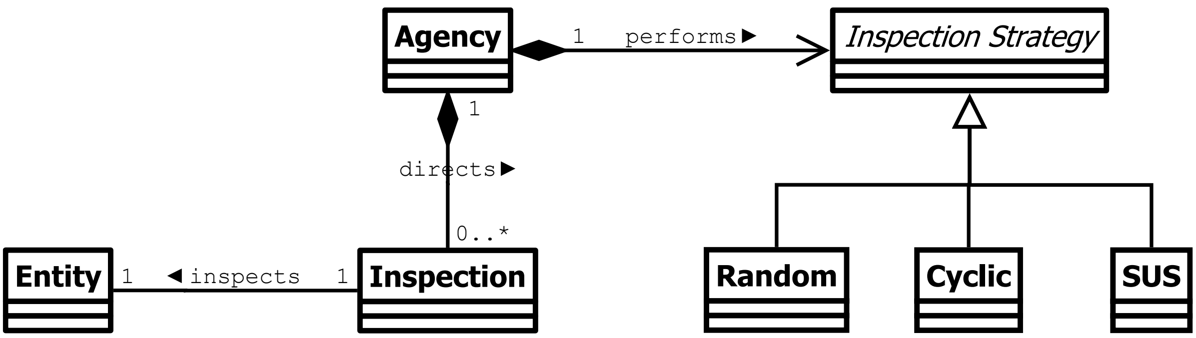

Figure 4.2 – ICARUS class diagram

The inspection agency implements one of 3 possible groups of inspection strategies. **Random** strategy, in which the entities and the rules that will be included in the inspection are chosen randomly. **Cyclic** strategy, wherein in a given cycle (i.e. with a defined frequency) inspections cover all the entities and a part or all of the rules. **Stochastic universal sampling** strategy (SUS), in which the provisions that will be included in the inspection are selected based on the estimated costs of compliance (i.e. the probability of inspection of each provision is proportional to the estimated costs of compliance with that provision). The inspection agency implements the selected strategy consistently, throughout a single run of the simulation.

An inspection can at t inspect only one entity and compliance with one or more rules (in that entity). At each t, the inspection agency initiates a number of inspections, depending on the chosen strategy and the inspection capacity. The inspection capacity of the agency is constant throughout the simulation and determines in how many entities and with how many individual provisions compliance can be checked in each discrete time interval t. The sequence of activities of entities matches the **fictitious play**[9] algorithm [54].

The model is characterized by a number of parameters and state variables. Some of them vary from entity to entity, from inspection to inspection, from state to state. Furthermore, the setup of the model, the simulation and its results depend on a number of input parameters that partly deterministically and partly stochastically affect the state of the system.

4.2. Model design concepts

4.2.1. Theoretical and empirical background

Entity's decision-making and action mechanisms are based on the rational choice theory (RCT) and game theory. Inspection agency strategies are based on the simplified patterns identified in empirical research and briefly described in the chapter 3.2.

The model can be described as a repeated, simultaneous, non-cooperative game of pure competition. Agents have imperfect, incomplete and asymmetric information. Entities in each "move" (discrete time) choose one pure strategy for each rule (provision), depending on their own subjective assessment of the likelihood of inspection. The inspection agency implements the selected inspection strategy.

4.2.2. Features and behaviour of entities

Entities are characterized by their uniqueness, heterogeneity, explicit goals, autonomy, locality, flexibility, and limited rationality. Additionally, they have the following characteristics:

1. Entities are rational, but their **rationality** is **bounded**[10], due to the limited scope of data on which they make decisions.
2. Entities are heterogeneous, given their propensity to take risks and given costs of compliance (resources required to achieve compliance).
3. Entities learn in line with the fictitious play model, that is, they draw conclusions about the future based on their own history.
4. Entities have a bias towards the present, that is, they attach more value to more recent experiences.
5. Entities do not communicate with each other.

The entity i is characterized by a vector of its **compliance costs,** i.e. resources needed to fulfill each rule in $\mathcal{O}$:

$$\boldsymbol{c_i} = (c_{i1}, \dots, c_{im}), \{c_{ij} \mid c_{ij} \in \mathbb{R} \wedge c_{ij} > 0\}, \forall i \in \mathcal{E}, \forall j \in \mathcal{O} \quad (1)$$

The costs of compliance may vary from entity to entity, from rule to rule. Differences in the compliance costs reflect possible differences in the complexity of the rules as well as in the characteristics of the entities that must fulfil them. For example, rules may require only the formalization of certain procedures, major process changes, or the establishment of entirely new processes that require significant changes to existing procedures and significant investments. On the other hand, organizations (entities) can differ from each

[9] Fictitious play is a simple model that obviously does not represent a completely real human learning [54], but because of its simplicity and suitability is often used as a model of learning agents and is used in more complex multi-agent models [54][92][105][106]. In a fictitious play, the agent learns from history and adjusts its belief about the strategies of other agent(s) in accordance with what it has learned. That is, the agent "believes" that the opponent is playing a mixed strategy and adjusts its estimate of the probability of the opponent's future actions based on the empirical distributions of its previous actions [54]. The fictional game is sensitive to the agent's initial assumptions [54] and is used to model learning in inspection models [13].

[10] Limited or bounded rationality assumes that people are not perfectly rational in decision-making, given the possession of incomplete information, cognitive limitations, and the final time in which they have to make decisions [10].

other in size, complexity, internal organization, business model, and the like. All of the above can also affect differences in the costs of achieving and maintaining compliance.

The entities also differ according to their propensity to take risks, i.e. according to their **risk appetite**. The propensity for risk-taking affects the entity's assessment of the likelihood of an inspection and is given by:

$$r_i \in \mathbb{R}, \forall i \in \mathcal{E} \tag{2}$$

The risk appetite is a stable individual characteristic[11] (in each run of a set simulation) that results from the personal preferences of the entities, and may vary from entity to entity.

The entities memorize history of their actions and the inspections to which they were subjected. At the moment t, entity i knows the state of its compliance with all rules and the results of all inspections to which it has been exposed in the last l time intervals (i.e. l determines entity's recollection span or memory). The information on the entitiy i's actions and results that are known to it are contained in its **memory matrix**:

$$\boldsymbol{\chi_i}(t) = \begin{bmatrix} h_{i1}(t-1) & \cdots & h_{im}(t-1) \\ \vdots & \ddots & \vdots \\ h_{i1}(t-l) & \cdots & h_{im}(t-l) \end{bmatrix}, \tag{3}$$
$$\{t, l \mid t, l \in \mathcal{T} \wedge 1 \leq l \leq t\};\ \forall i \in \mathcal{E}$$

The possible values of the elements of the momory matrix depend on the following:

$$h_{ij}(t-u) = \begin{cases} -2, & \text{if there was an inspection and it determined violation of } j \text{ at } t-u \\ -1, & \text{if there was not an inspection of } j \text{ at } t-u\text{, and the entity was noncompliant} \\ 1, & \text{if there was not an inspection of } j \text{ at } t-u\text{, and the entity was compliant} \\ 2, & \text{if there was an inspection of } j \text{ at } t-u\text{, and the entity was compliant} \end{cases} \tag{4}$$

Entities make (boundedly) rational decisions about whether to comply or violate each rule in $\mathcal{O}$, comparing the cost of compliance and the perception of the expected value of the penalty. The perception of the expected value of a penalty for an entity i for violating a rule j is given by the product of the prescribed penalty for violating that rule k_j and the entity's assessment of the **perceived likelihood of inspection** $p_i(t)$.

Penalties can, in general, be specific to each provision and are represented by a vector:

$$\boldsymbol{k} = (k_1, \dots, k_m), \{k_j \mid k_j \in \mathbb{Z} \wedge k_j > 0\}, \forall j \in \mathcal{O} \tag{5}$$

In the developed model, the **penalty for violation** is the same for all the provisions, which is inherent to many regulations, i.e.:

$$k = k_1 = k_2 = k_3 = \ \dots = k_m \tag{6}$$

$p_i(t)$ is the inspection probability valuation function given by:

$$p_i(t) = \begin{cases} \frac{p_{Ci}}{r_i}, & \text{for } \ t = 1 \\ f(r_i, \boldsymbol{\chi}_i(t), \kappa), & \text{for } \ t > 1 \end{cases}, \quad \forall i \in \mathcal{E}, \forall t \in \mathcal{T} \tag{7}$$

Since there is no inspection history in the first step of the simulation, the entity's initial perception (belief) of the likelihood of inspection (at $t = 1$) is set based on the input parameters, with p_{Ci} being the entity i's initial perception of the likelihood of inspection related to the inspection agency's capacity to conduct inspections. In addition, the initial perception of the likelihood of inspection is inversely proportional to the entity's risk appetite (r_i).

[11] In the model, entities are primarily organizations. Therefore, risk appetite is considered to be a consequence of the corporate culture of the organization. In some circumstances, risk appetite may not be a stable characteristic.

In the further steps, perception of the entity i's probability of inspection is influenced by its risk appetite (r_i), the history known to it $\boldsymbol{\chi}_i(t)$ and **the time discount index** κ, i.e. the measure of sensitivity to delay, which is modelled by a hyperbolic function[12]. History influences the entity's perception of the likelihood of an inspection at the time t ($p_i'(t)$) and the assessment of the "danger" of the inspection ($\mathrm{E}_i(t)$). $\mathrm{E}_i(t)$ reflects the **accuracy of inspections in detecting noncompliance** and is based on the entity's knowledge of the historical accuracy of inspections. Namely, since the inspection may encompass one or more rules from $\mathcal{O}$, the entity cares not only about the perception of the probability that it will be inspected at a given time interval, but also about the danger (risk) that the inspection will encompass one or more rules that the entity violates.

Accordingly, the perception of the probability of inspection in steps (time intervals) $t > 1$, is:

$$p_i(t \mid t > 1) = p_i'(t) \cdot \mathrm{E}_i(t) \cdot \frac{1}{r_i} \tag{8}$$

where $p_i'(t)$ is perception of the entity i about the probability that it will be the subject of an inspection at the time t[13]:

$$p'_i(t) = \frac{\sum_{v=1}^{t-l}\left(\frac{\sum_{j=1}^{m}\left[h_{ij}(v) = |2|\right]}{m(1+v\kappa)}\right)}{\sum_{z=1}^{t-l}\frac{1}{1+z\kappa}}, \forall i \in \mathcal{E}, \forall t, l \in \mathcal{T} \tag{9}$$

The entities do not know which inspection strategy the agency is implementing, but based on the history of inspections and in accordance with the fictitious play algorithm, they learn about inspections and try to predict the future move of the inspection agency. Therefore $p_i'(t)$ is calculated taking into account the recorded history, with historical experiences being discounted according to κ. That is, the impact of historical experiences on the current estimate depends on the index κ and experiences from the past have the same ($\kappa = 0$) or smaller influence ($\kappa > 0$) than more recent experiences.

$\mathrm{E}_i(t)$ acts as a corrective factor for $p_i'(t)$ and is calculated as:

$$\mathrm{E}_i(t) = \sum_{u=1}^{t-l}\left(\frac{m\sum_{j=1}^{m}\left[h_{ij}(u) = -2\right]}{\sum_{j=1}^{m}\left[h_{ij}(u) = |2|\right] \cdot \sum_{j=1}^{m}\left[h_{ij}(u) < 0\right]}\right), \forall i \in \mathcal{E}, \forall t, l \in \mathcal{T} \tag{10}$$

That is, $p_i'(t)$ reflects only the probability that the entity will be subject to inspection, but not the probability that it will be caught in violation. Let's imagine a situation in which an entity is non-compliant with the same, but only one rule out of 10 rules, and the inspection is conducted in each step of the simulation and always includes 9 rules with which the entity is compliant, $p_i'(t)$ would be $\frac{9}{10}$. However, the entity would never be caught in violation. Therefore, based on the history of inspections, the entities assess how accurate the inspections are in detecting non-compliance and compare that data with their own non-compliance record. For example, if the entity i was inspected at the time u, and the inspection encompassed 4 rules, with $m = 10$, the inspection determines 2 violations, and the entity violated a total of 3 rules, then $\mathrm{E}_i(u) = \frac{10 \cdot 2}{4 \cdot 3} = 1{,}67$. That is, the entity will, due to the fact that it violated 33% of the rules, and the inspection found non-compliance in 50% of the rules it supervised, increase the assessment of the perceived accuracy of the inspection. Only offenses in which entities were caught are included in the numerator. It the entity i violated

[12] Time discounting is the tendency (or bias) to value events less (or to give events less "weight"), the more distant they are from the present. In the example of the fictitious play, the application of the time discounting may mean that information about recent events has a greater impact on a player's belief in the opponent's presumed strategy than older information and, consequently, on his decisions. Time discounting is often [107] modelled by a hyperbolic function.

[13] The expressions (9) and (10) utilize the Iverson notation [108].

the rule j at the time t and the inspection did not recognize that or erroneously concluded that the entity was compliant with j, such an event would be considered as inspected compliance.

After calculating the perceived probability of inspection, entities compare the product of that probability and the punishment with the cost of compliance and based on the optimization of the **expected utility function** of the entity i at t (11), make a decision on compliance or violation of every rule.

$$\pi_i(t) = \sum_{j=1}^{m} arg\ min\left\{c_{ij}, p_i(t) \cdot k_j\right\}, \forall i \in \mathcal{E}, \forall j \in \mathcal{O}, \forall t \in \mathcal{T} \tag{11}$$

That is, an entity makes a choice between 2 options for each rule in $\mathcal{O}$ at the time t: to violate the rule or to comply with it, depending on which choice will have a lower expected cost. In the general case, the expected utility function maximizes the benefit (utility). In this model, compliance has unavoidable costs, and non-compliance has expected costs. The entities try to minimize the cost at every step of the simulation.

If the product of the amount of the prescribed penalty (k_j) for a rule j and the perceived probability of inspection assessed by the entity i ($p_i(t)$) is greater than the cost of compliance with that rule at the moment t, the entity will comply with the rule. Otherwise, the entity will choose to violate the rule. Thus, the compliance of the entity i with all the rules from $\mathcal{O}$ at the time t is given by the following vector:

$$\boldsymbol{o_i}(t) = (o_{i1}(t), \dots, o_{im}(t)), o_{ij} \in \{-1,1\}, \forall i \in \mathcal{E}, \forall j \in \mathcal{O}, \forall t \in \ \mathcal{T} \tag{12}$$

The possible values of the compliance vector depend on the following:

$$o_{ij}(t) = \begin{cases} -1,\ if\ c_{ij} > p_{ij}(t)\ \cdot k_j \text{ (violation)} \\ 1,\ if\ c_{ij} < p_{ij}(t)\ \cdot k_j \text{ (compliance)} \\ \sim U\{-1,1\},\ if\ c_{ij} = p_{ij}(t)\ \cdot k_j \text{ (random choice)} \end{cases} \tag{13}$$

If, at the time $t-1$ entity i's compliance with the rule j was not inspected, the following will apply: $h_{ij}(t-1) = o_{ij}(t-1)$.

4.2.3. Features and behaviour of the inspection agency and inspections

The inspection agency's assessment of the costs of compliance with each provision is contained in the vector $\boldsymbol{d}$:

$$\boldsymbol{d} = (d_1, \dots, d_m), \left\{d_j \mid d_j \in \mathbb{R}\ \wedge d_j > 0\right\}, \forall j \in \mathcal{O} \tag{14}$$

The inspection agency determines which entities and rules will be inspected at the time t, depending on the chosen inspection strategy and the inspection capacity. The **inspection capacity** $I_C, \{I_C \mid I_C \in \mathbb{N}\ \wedge 0 \leq I_C \leq mn\}$, determines how many entity-rule combinations $\{i,j\}$, $i \in \mathcal{E}, j \in \mathcal{O}$ inspections can cover, in a discrete time interval t. Thus, the selection of entities and rules for inspection is a function by which $I(t) \subseteq \mathcal{E} \times \mathcal{O}, t \in \mathcal{T}$ is determined. $I(t)$ is influenced by the chosen inspection strategy.

By applying the **random** strategy, the inspection agency selects entities and rules completely randomly, with each entity-rule combination having the same probability of being inspected:

$$p^{inspection} = \frac{I_C}{m \cdot n} \tag{15}$$

That is, the choice of entity-rule combinations that will be inspected can be viewed as a random variable with a discrete uniform distribution over $T, \{T \mid T \in \mathbb{N} \wedge 1 \leq T \leq mn\}$. The selection process is repeated until I_C entity-rule combinations are selected for inspection at t.

By applying the **random entity** strategy, the inspection agency completely randomly selects entities that will be inspected. The inspection then encompasses all the rules in selected entities. The probability of selecting an individual entity in the inspection sample is given by:

$$p^{inspection} = \frac{\frac{I_C}{m}}{n} = \frac{I_C}{m \cdot n} \tag{16}$$

Selection of entities that will be inspected can be viewed as a random variable with the uniform distribution over V, $\{V \mid V \in \mathbb{N} \wedge 1 < V < n\}$. This selection process is repeated $\frac{I_C}{m}$ times.

By applying the **cyclic** strategy, inspection agency sequentially selects entities for inspection, until all entities are inspected, which marks the completion of the cycle. After that, a new cycle begins. The cycle may include one or more rules from O. If it is specified that one inspection cycle will encompass g rules, $g \cap O$, then one inspection cycle will last T steps (time intervals):

$$T = \frac{n \cdot g}{I_C} \tag{17}$$

If the inspection agency still has unused capacity to conduct inspection within one step (time interval), and after the completion of the cycle (i.e. if T is not an integer), a new inspection cycle will begin in the same step. After all the rules in all the entities have been inspected, further rules are sequentially inspected in the next cycle. For example, if the entities must comply with 5 rules, and in one inspection cycle 3 rules are included, then in the first inspection cycle compliance with rules 1, 2 and 3 will be assessed, in the second inspection cycle compliance with the rules 4, 5 and 1, in the third cycle compliance with the rules 2, 3 and 4, in the fourth cycle compliance with the rules 5, 1 and 2, etc. The entities are selected sequentially in the cycle, but the order of their selection and the rules that are encompassed by the inspection in that cycle are not known to the entities.

Inspection cycles have different durations in practice and may differ for different types of entities. For example, in the area of inspection of compliance with environmental regulations, each supervised entity may have to be inspected once every two years [30], once every three years [41] or once every five years [30]. In the area of occupational safety regulation inspection, entities may have to be inspected once every two years [32]. In the banking supervision area, banks may have to be supervised at least once every year, every other year or every third year [31].

By applying the **weighted random selection** strategy, the inspection agency randomly selects the entities and provisions that will be inspected, but the probabilities of inspecting individual provisions (rules) vary, depending on the relevant value of the vector $\boldsymbol{d}$. That is, a rule for which the inspection agency believes to entail higher compliance costs will be proportionally more often inspected: $p_j^{inspection} \propto d_j$. Accordingly, the probability that the rule j will be covered by the inspection is given by:

$$p_j^{inspection} = \frac{I_C}{n} \cdot \frac{d_j}{\sum_{v=1}^{m} d_v} \tag{18}$$

The random weighted selection strategy is implemented via **stochastic universal sampling** (**SUS**). Stochastic universal sampling is a proportionate selection algorithm which is commonly used in the field of genetic algorithms to select units for re-combination. The algorithm was proposed by Baker [55], as an improvement on the classic "roulette wheel", since SUS removes bias inherent to the "roulette wheel". Additionally, SUS requires generation of only one random number per step. SUS has been applied as a strategy of random weighted selection in ICARUS due to its efficiency and the fact that, from the point of view of statistical sampling theory, it corresponds to systemic sampling [56, p. 32].

Figure 4.3 shows how the SUS algorithm works and how it is applied in ICARUS. Selection of entities and provisions for inspection can be visually presented as a tape that sequentially lists all the rules and all the entities. The length of the part of the tape that displays a particular rule depends on the relevant value of the vector $\boldsymbol{d}$. That is, the length of the part of the tape that relates to the rule j is proportional to the amount d_j. During the initialization of the model, a "window" is calculated. It indicates the "distance" between two inspections and depends on the number of entities, the number of rules, the capacity of the inspection agency and assumptions of the inspection agency about the costs of compliance of each provision:

$$w = \frac{n}{I_C} \cdot \sum_{j=1}^{m} d_j \tag{19}$$

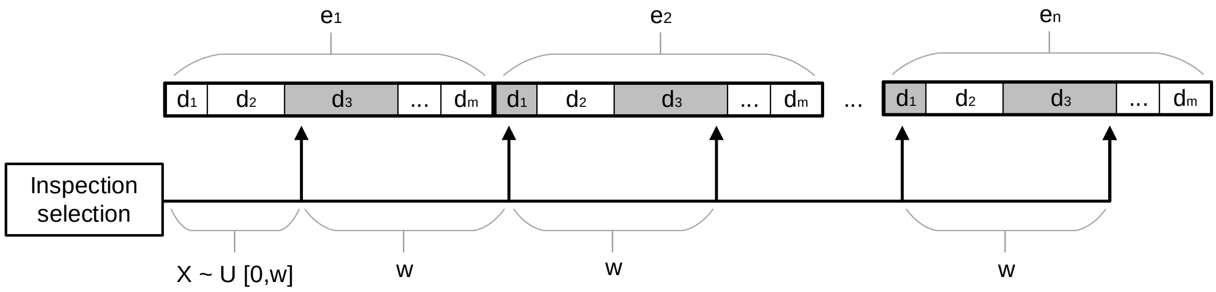

Figure 4.3 – Graphical representation of the stochastic universal sampling (SUS) algorithm

The inspection window allows division of the tape into I_C segments. At the inspection step (time) t, the agency, based on the value of a random variable $X(t)$ with a uniform distribution over $[0, w]$, selects inspection sample that, relative to the tape in the Figure 4.3, can be expressed as:

$$I(t) = \{X(t), X(t) + w, X(t) + 2w, \ldots, X(t) + (I_C - 1)w\} \tag{20}$$

Only pure inspection strategies are implemented in the ICARUS. A strategy cannot be changed during the run of the simulation.

Based on the chosen inspection strategy, the inspection agency coordinates inspections that verify compliance of the selected entities with the chosen rules. It is not reasonable to assume that inspections are infallible, that is, that they always establish compliance or violation with complete accuracy. However, since the inspected entity has a reason to complain (and present evidence) when unjustly accused of being in violation, it is reasonable to expect that inspection errors will almost always be one-sided, i.e. errors will be manifested as misidentified compliance rather than as an erroneously determined noncompliance [32].

Therefore, inspections may result in a type I error (false positive), but not in a type II error (false negative). The **inspection accuracy rate** in determining the violations in the model is defined by γ. The type I errors in the model are implemented via the random variable $X_{ij}(t)$. When inspecting the rule j in the entity i at t, if i did not violate that rule j, the inspection assesses that i is compliant with the rule j. However, if i did violate the rule j, then depending on the value of a random variable $X_{ij}(t)$ and its value in relation to γ the inspection determines violation or compliance. Therefore, the state of compliance of the entity i with the provision j at the time t determined by the inspection can be presented as:

$$u_{ij}(t) = \begin{cases} -2, & (violation)\ o_{ij}(t) = -1 \ \wedge \ X_{ij}(t) < \gamma, X_{ij}(t) \sim U(0,1) \\ 2, & (compliance)\ o_{ij}(t) = -1 \ \wedge \ X_{ij}(t) \geq \gamma, X_{ij}(t) \sim U(0,1) \\ 2, & (compliance)\ o_{ij}(t) = 1 \end{cases} \tag{21}$$

4.2.4. Additional features of the model

This chapter presents information that, in line with the ODD+D protocol[14], should be listed in the description of the model design, and is not contained in the previous chapters.

Social norms and collectives. When deciding whether to violate or comply, the entities take into account only their (internal) perception of the likelihood of inspection, punishment and compliance costs and do not consider social norms or cultural values. ICARUS models entities as completely independent agents. The decisions and activities of an entity are not influenced by another entity (not even a neighbouring one). They are based only on the internal parameters and values of the state variables of that entity. Entities' internal parameters take on values based on the input parameters, stochastic processes, and the interaction

[14] Historically, there was no generally accepted protocol for describing multi-agent models [69]. Grimm and co-authors [53] published the ODD protocol (Overview, Design concepts, and Details) in 2005 with the intention of standardizing the description of individual-based and agent-based models and facilitating the understanding and replication of the model. The ODD protocol was initially aimed at ecological models [52], but the popularity of the protocol led to further upgrades, such as the ODD+D protocol [52], which attempts to advance the description of models that involve decision-makers.

of entities and inspections. The entities do not coordinate their activities (among themselves nor with other agents), and the inspection agency coordinates the activities of inspections (the order in which each inspection will inspect compliance, including the determination of the entity and the rules to be inspected).

Learning and decision making. Learning is modelled as fictitious play. Entities learn individually, i.e. each entity learns exclusively on the basis of the history of its decisions and experienced inspections. Entities are boundedly rational because they have imperfect and incomplete information on the inspection agency's strategy, as they learn and predict future actions solely on the basis of their own experience. The entities have an explicit goal – to optimize the value of the expected utility function $\pi_i(t)$ that is defined with (16) – and their decisions are aimed exclusively at achieving that goal. In each step of the simulation, the entities decide whether to violate or to comply with the rules in the current step, and not on the actions in the further steps. The inspection agency and inspections are not self-directed and only implement the inspection strategy that was defined via the input parameters.

Perception of the environment and interactions. The spatial location of the entity has no influence on the model and the simulation results, however, the interaction of the entity and the inspection in the simulation is visually displayed by moving the inspection avatar to the entity avatar, i.e. all interactions are local. These movements are driven by the chosen inspection strategy.

The parameters and state variables of the implemented model are documented in detail in the rest of this chapter. Inspections are the only form of interaction between agents in the model and affect the values of a number of state variables. The inspection process encompasses the following activities (see Figure 4.4):

1. Agency → inspection : initiating an inspection (activity 2.4).
2. Inspection → agency : inquiry on the inspection target (activity 2.5).
3. Agency → inspections : set the inspection target (completion of activity 2.5).
4. Inspection → entity : the state variable request (event 2.6).
5. Entity → inspection : information on the state of variables (completion of activity 2.6).
6. Inspection → agency : result of the inspection (completion of activity 2.4).

All of these interactions are direct, and the selection of specific agents that will interact depends on the chosen inspection strategy and the stochastic processes described later in this chapter.

All agents know the values of all of their own state variables and a part of the global parameters and state variables. The exchange of information on the values of agent's parameters and state variables takes place only through inspection and communication of the inspection agency with inspections. Entities can gain new knowledge about the environment only through interactions with inspections and through the analysis of history of their own actions. Since inspections can erroneously determine compliance (even though the entity was non-compliant with the inspected rule), depending on the set accuracy of inspections (γ) and the value of the random variable, inspection results may introduce incorrect information into the system.

Inspections are costless for the inspection agency, i.e. the number and scope of inspections depends exclusively on the capacity of the inspection agency and the chosen inspection strategy.

Time, which is included in the model as a discrete variable, has a significant impact on behaviour of the entities. The entity i's decisions depend on the history that is known to it, contained in the matrix $\boldsymbol{\chi_i}(t)$, and on the time discounting index κ, which determines how the impact of previous events on entity's decisions weakens with time. The impact of time on the behaviour of the inspection agency depends on the chosen inspection strategy. Time does not affect random inspection strategies and random weighted strategies, but does affect the cyclical strategies. Namely, when cyclical strategies are applied, the choice of entities and rules that will be inspected in the next step is contingent on the previous steps of the inspection cycle.

The computer simulation of the model has two parts: initialization, that is performed only once (at the beginning of the simulation) and a step, which can be performed more than once.

Heterogeneity and randomness. Random variables have a significant impact on the model and are used extensively, to better model the uniqueness, heterogeneity and bounded rationality of entities as well as errors in compliance inspections. Random variables affect the value of the following variables and interactions:

1. The order of selection of entities and inspections in all phases and activities of the model is random and depends on the values of a random variable with a discrete uniform distribution.

2. The inspection agency's knowledge of the costs of compliance ($\boldsymbol{d}$) depends on the values of a random variable with a discrete uniform or exponential distribution, unless explicitly specified through the input parameters of the model. The value of the vector $\boldsymbol{d}$ is set when the model is initialized.
3. The costs of compliance ($\boldsymbol{c_i}$) vary from entity to entity and differ from the inspection agency's assessment of the costs of compliance ($\boldsymbol{d}$). They depend on the values of a random variable with a discrete uniform or exponential distribution. The value of the vector $\boldsymbol{c_i}$ is set when the model is initialized.
4. The risk appetite (r_i) varies from entity to entity, depending on the values of a random variable with a discrete uniform distribution. Value of the parameter r_i is set when the model is initialized.
5. The following activities (choices) related to inspection strategies depend on a random variable with a discrete uniform distribution:
    a. Entity selection and rules that will be inspected in the random selection strategy.
    b. Entity selection in the random entity selection strategy.
    c. Entity selection and rules that will be inspected in the weighted random selection strategy.
    d. The order of the entity and rule selection in the cyclical strategy.

    These choices occur in every step of the simulation.
6. Determining compliance (but not violation) during an inspection depends on the values of a random variable with a discrete uniform distribution. Compliance/violation is determined at each step of the simulation, at each inspection.

Accordingly, the entities are heterogeneous in their parameters and state variables and differ from each other in terms of the costs of compliance ($\boldsymbol{c_i}$) and risk appetites (r_i). On the other hand, entities are homogeneous in terms of the decision-making mechanism (all entities have the same goal – optimizing the value of the expected utility function $\pi_i(t)$), the learning algorithm, as well as the scope of data used in learning and decision-making. Inspections are homogeneous.

The uniform distribution is often used to generate heterogeneity and randomness in agent-based models, especially when empirical data on parameter distribution characteristics are not available [17][23][12][57][58]. The exponential distribution was also applied, given the empirically observed regularity that a small percentage of locations, victims, offenders, law enforcers, or other units in any distribution of crime or injustice cause the most harm [59].

4.3. Computer simulation implementation

This chapter describes the implementation of the model in the NetLogo [60] environment.

4.3.1. Processes in the implemented model

Figure 4.4 displays the sequence of activities in the model and the implemented computer simulation via a UML sequence diagram.

The user starts the simulation by initiating the `setup()` procedure (activity 1.1), in which the initially set exogenous (input) simulation parameters are used in creating and preparing the environment and setting the internal parameters and initial values of the state variables. In activity 1.2, the simulation creates the entities and sets their internal parameters and variables, and then creates the inspection agency (activity 1.3) and sets its internal parameters and variables. The inspection agency then (activity 1.4) creates inspections, thus completing the `setup()` procedure. After that, the user in activity 2.1. (the `go()` procedure) initiates one or more simulation steps.

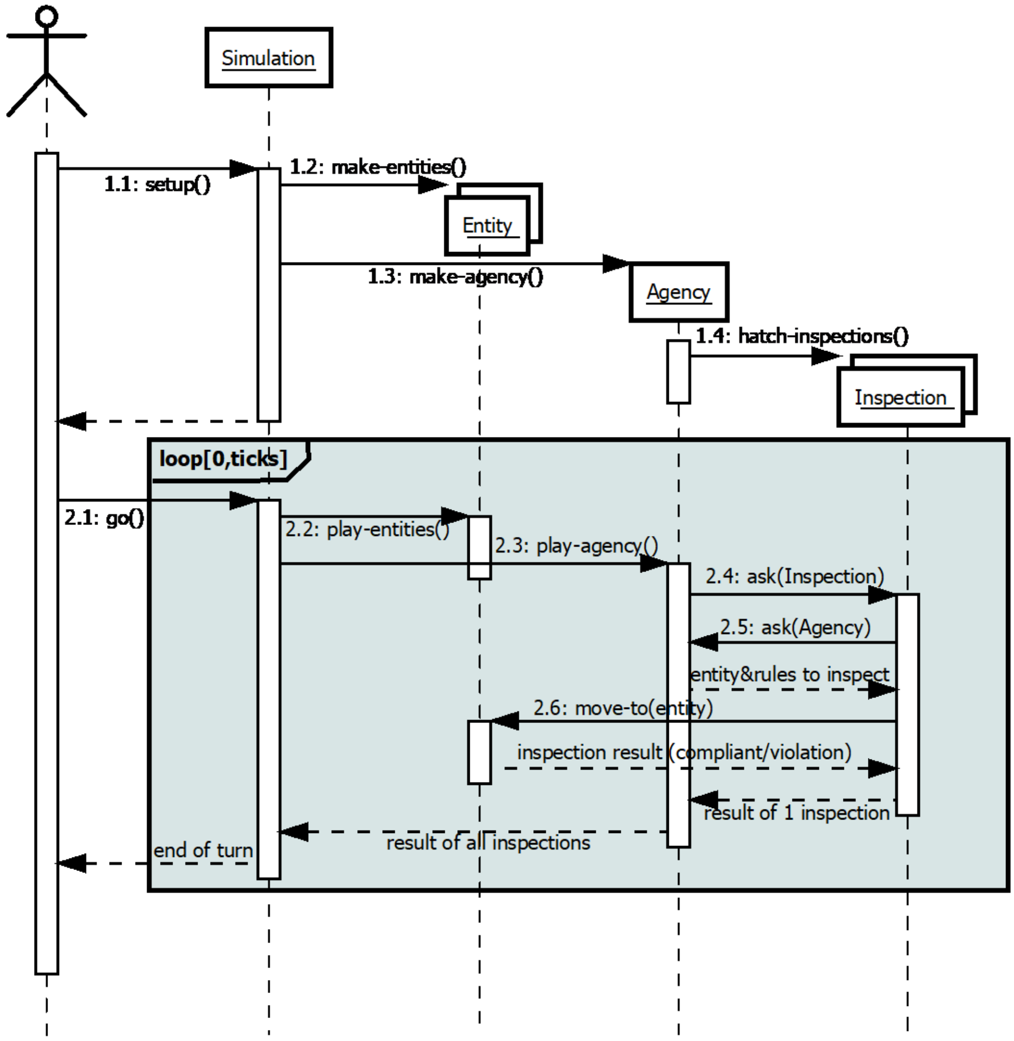

Figure 4.4 – UML sequence diagram of the ICARUS

Every simulation step t starts with the activity 2.2 in which each entity from $\mathcal{E}$, depending on the initial parameters and the recorded history of its actions and inspections it has experienced, decides whether to violate or to comply with each rule from $\mathcal{O}$ (in this simulation step). The procedure `go()` also initiates the activity 2.3 in which the inspection agency, depending on the inspection strategy defined via the input parameters of the simulation, determines which entities and which rules will be inspected in the simulation step t. The inspection agency then initiates communication with all the inspections (activity 2.4). Each inspection asks the agency for instructions on which entity and rules it should inspect in that step (activity 2.5). In one step, one inspection can inspect only one entity. After receiving the instructions, each inspection moves spatially to the location of the given entity (activity 2.6), inspects the given rules (i.e. examines the values of relevant variables of the entity at the given location) and informs the inspection agency about the result of the inspection. After the inspection agency receives the results of all inspections, it forwards them to the simulation environment and the results are recorded and displayed graphically. This completes the simulation step and the simulation can continue with the next step or it can stop.

A brief description of the functionality of all the procedures in the simulation is given below. Procedures that are started manually, through the user interface are:

1. `setup()` : set up simulation environment.
2. `go()` : perform one or more simulation steps.
3. `profiles()` : detection of parts of the code that represent a "bottleneck" at the execution time; includes one run of the `setup()` procedure and 100 repetitions of the `go()` procedure .
4. `validate()` : performing the specific validation of the model (these functions are described in detail in the latter chapters); includes one run of the `setup()` procedure and a given number of repetitions of the `go()` procedure .

Procedures that control actions of the entities, inspection agency and inspections:

1. `make-entity()` : setting the initial parameters of all entities. Run from the `setup()` procedure.
2. `make-agency()` : setting the initial parameters of the inspection agency. Run from the `setup()` procedure .
3. `play-entity()` : entity activities in one simulation step. Run from the `go()` procedure.
4. `play-agency()` : activities of the inspection agency and all the inspections in one step of the simulation. Run from the `go()` procedure .

Auxiliary procedures used in the simulation, which have output values:

1. `occurrences (Integer, List) : Integer` : returns the number of occurrences of a given number in a given list.
2. `abs-occurrences (Integer, List) : Integer` : returns the number of occurrences of numbers in the given list with an absolute value equal to the given value.
3. `greater (Integer, List) : Integer` : returns the number of occurrences of numbers with a value greater than the given value.
4. `smaller (Integer, List) : Integer` : returns the number of occurrences of numbers with a value less than the given value.
5. `replace-elem (Integer, Integer, Integer, List) : Integer` : replace (in a two-dimensional list) a given element with a new element at a given location.

Auxiliary procedures used in the simulation:

1. `debug-prnt (List)` : print debug data.
2. `clean-up ()` : prepares the visual appearance of the user interface.
3. `out-prnt (List)` : record the output values of the simulation to a file.

The Figure 4.5 displays the UML class diagram of the implemented model. Since NetLogo is not an object-oriented programming language, this UML diagram provides only an approximation of the implemented model, and is aimed at facilitating understanding. The described procedures are shown in the class diagram as methods (or operations), and the parameters and state variables are shown as attributes of the class diagram. The *Simulation* class is not an actual class but represents a global context – the global context is often shown as a separate class in UML class diagrams that describe agent-based models [61][62].

The inspection agency, entities and inspections are declared in the NetLogo as turtle-type agents. Each agent can use a series of primitives that are specific to its type, which includes predefined state variables as well as specific procedures. These primitives are not listed in the class diagram.

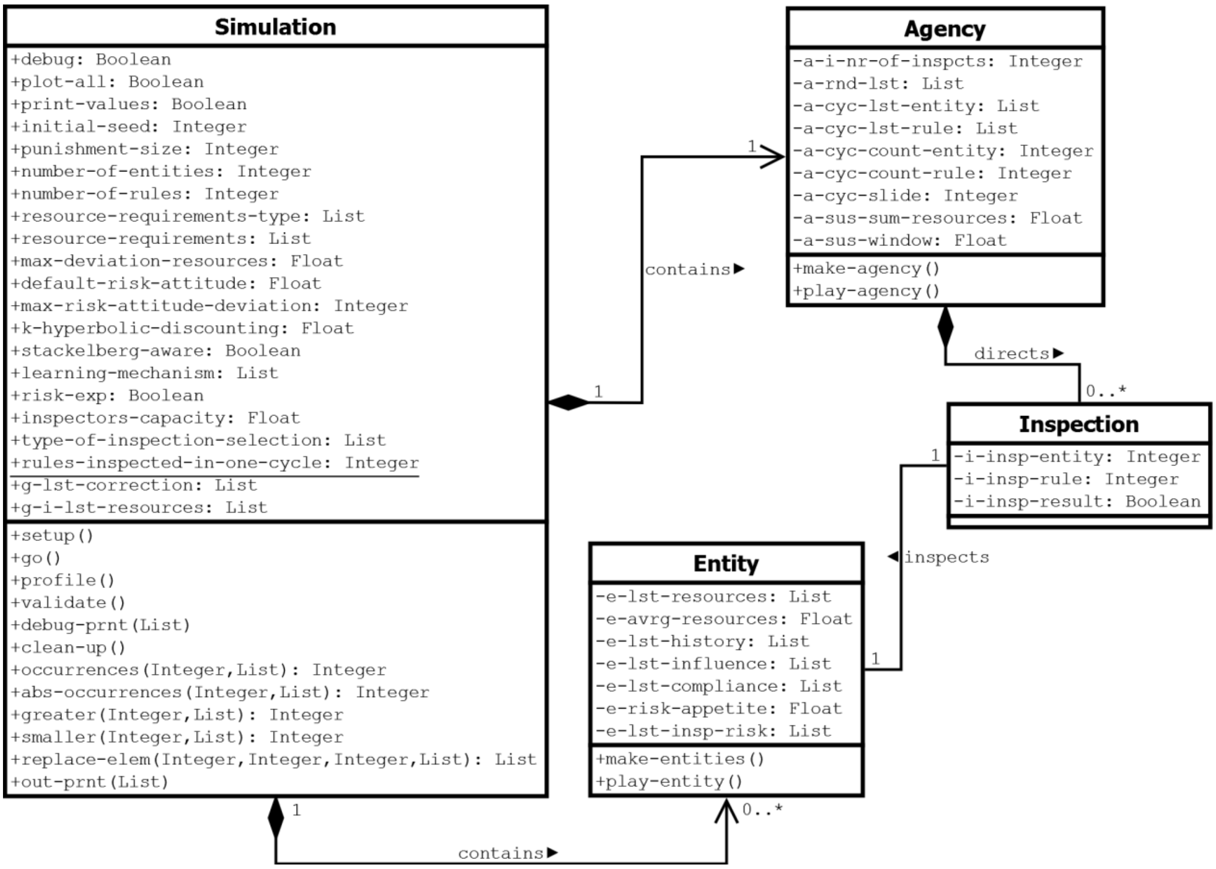

Figure 4.5 – UML class diagram of the implemented model

4.3.2. Input parameters and state variables

Table 4.1 shows the link between the names of the variables in the implemented model and the variables listed and described in the previous subchapters.

Table 4.1 – Links between the names of the variables

Global	
Variable	**NetLogo variable**
k	punishment-size
n	number-of-entities
m	number-of-rules
$\boldsymbol{d}$	g-i-lst-resources
κ	k-hyperbolic-discounting
I_C	inspectors-capacity
γ	inspection-accuracy
g	rules-inspected-in-one-cycle
l	g-memory-turns

Entities	
Variable	**NetLogo variable**
$\boldsymbol{c_i}$	e-lst-resources
$\boldsymbol{f}(\boldsymbol{c_i})$	e-avrg-resources
$\boldsymbol{\chi_i}$	e-lst-history
$\boldsymbol{f}(\boldsymbol{\chi_i}, \kappa)$	e-lst-influence
E_i	e-lst-insp-risk
$\boldsymbol{o_i}$	e-lst-compliance
r_i	e-risk-appetite

Inspection agency	
Variable	**NetLogo variable**
$\boldsymbol{f}(\boldsymbol{d}, w)$	a-i-nr-of-inspcts
w	a-SUS-window

A brief description of the variables that are not already described in the previous subchapters is as follows.

Parameters whose value can be entered and changed via the user interface in NetLogo are implemented globally and are listed in the class diagram (Figure 4.5) in the context of the “class” *Simulation*. The variables in the *Simulation* "class" that do not have the prefix "g-" indicate the input parameters of the simulation, or parameters that can be set via the user interface.

The computer implementation contains several variables of the *Boolean* type that can be set via the user interface and which control the way the simulation progresses, including: `debug` (whether to print *debug* data), `plot-all` (whether the simulation results are plotted on graphs on the user interface) and `print-values` (whether the simulation results will be printed to the output file). The `initial-seed` parameter allows setting the seed that will be used to generate (quasi) random values. If the value of the parameter is 0, the seed will be generated (quasi) randomly at the start of the simulation. In two simulations with the identical input parameters and identical initial seed, all the simulation steps will have the same results. This enables experiment replication.

The parameters `resource-requirements`, `resource-requirements-type` and `resource-requirements-param` are connected to setting the value of the vector $\boldsymbol{d}$, i.e. with the knowledge of the inspection agency about the compliance costs of each rule (provision). The `resource-requirements-type` parameter allows the user to select one of the following options: *Input from line*, *Uniform distribution*, *Exponential distribution*, and *Validation*. If the *Input from line* option is selected, the user must define the inspection agency's knowledge about the compliance costs of each rule manually via the `resource-requirements` parameter . If the user selects *Uniform distribution*, the resource requirements for each provision j will be $d_j = X \sim U[0,$ `resource-requirements-param` $], \forall j \in \mathcal{O}$. If the user selects *Exponential distribution*, then compliance costs of each provision will be specified as the value of a random variable with an exponential distribution whose arithmetic mean is the value of the `resource-requirements-param` variable. By selecting the *Validation* option, compliance costs are defined through the `validate()` validation procedure. The knowledge of the inspection agency on the compliance costs is ultimately recorded in the global variable `gi-lst-resources` .

The parameters `default-risk-attitude`, `max-risk-attitude-deviation` and `risk-exp` are connected to the risk appetite of the entity (r_i, i.e. `e-risk-appetite`). If the `risk-exp` variable is activated (*TRUE*), then the risk appetite of each entity will be the value of a random variable with an exponential distribution with the arithmetic mean specified by the `default-risk-attitude` parameter. If the `risk-exp` variable is not activated (*FALSE*), then the risk appetite of each entity will be determined as a random variable with a uniform distribution in the range with the mean defined by the variable `default-risk-attitude`, and the upper and lower limits defined by the variable `max-risk-attitude-deviation` (as a percent of deviation from the `default-risk-attitude` value). The risk appetite of each entity is created dynamically, at the beginning of the simulation in the `make-entity` procedure. The risk attitude of 1 means that the entity is perfectly rational in making decisions. A risk attitude greater than 1 denotes that the entity is risk-taker, and a value less than 1 denotes that the entity is risk-averse.

The `learning-mechanism` variable allows the user to choose one of 2 possible learning methods: *Fictitious play* or *Reinforcement learning*. All entities in the model learn via the same (selected) method.

The `type-of-inspection-selection variable` allows the user to select one of 4 possible inspection sample selection methods: *Random*, *Random entity*, *Cycle* and *Stochastic universal sampling*. The inspection agency applies the selected method to guide entity inspections throughout the simulation.

If the `stackelberg-aware` variable is activated (*TRUE*), all entities know that the inspection agency applies the *Stohastic universal sampling* inspection strategy and weigh the perceived probability of inspection of each provision with (their own) costs of compliance for the provision in question.

The global variable `g-lst-correction` contains corrective factors for discounting historical data calculated based on the value of the variable `k-hyperbolic-discounting`.

The entities are characterized by several state variables whose values are determined based on the input parameters and the simulation runtime.

The `max-deviation-resources` parameter determines how much will each entity's costs of compliance deviate from the costs of compliance that the inspection agency believes to be accurate, and which are stored in the `gi-lst-resources` variable . That is, the cost of compliance of the entity i with the provision j will be $c_{ij} = X \sim U[d_j$ - `max-deviation-resources`, d_j + `max-deviation-resources`$], \forall i \in \mathcal{E}, \forall j \in \mathcal{O}$.

The value of the variable `e-avrg-resources` is the arithmetic mean of the values of the elements of vector $\boldsymbol{c_i}$ (`e-lst-resources`), and the variable `e-lst-influence` determines the influence of the previous simulation steps on entity decisions.

The inspection agency is characterized by several state variables. The choice of the subset of variables that will be used depends on the applied inspection strategy, i.e. on the value of the `type-of-inspection-selection` variable. Figure 4.6 displays a conceptual UML class diagram, as it would be implemented if NetLogo were an object-oriented programming language. The aim of the diagram is to clarify the purpose of individual variables.

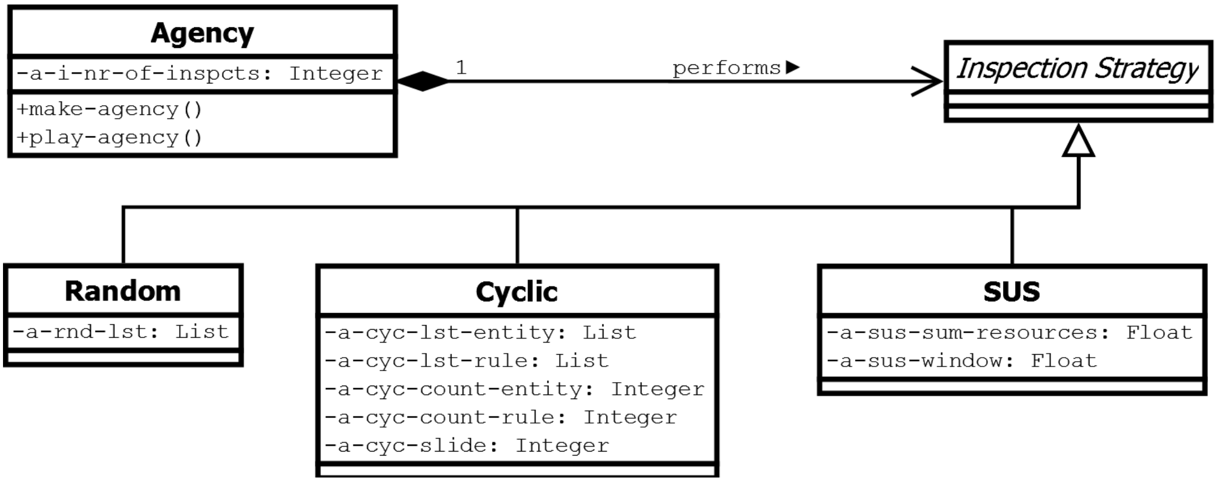

Figure 4.6 – Conceptual UML class diagram of the inspection agency

Inspections utilize 3 state variables: `i-insp-entity`, `i-insp-rule` and `i-insp-result`. Variables `i-insp-entity` and `i-insp-rule` specify which entity and which rule will be inspected, and `i-insp-result` returns the results of inspection.

The value of the variable `ai-nr-of-inspcts` depends on the inspection capacity (I_C, i.e. `inspectors-capacity`) and determines the number of inspections that can be performed in one step of the simulation. Additionally, when applying the *Stohastic universal sampling* strategy, the value of `a-i-nr-of-inspcts` depends also on the inspection window (w or `a-SUS-window`).

The variable `a-rnd-lst` is used if one of the random inspection strategies is selected and contains the list of entities that will be inspected in the following step (the selection by the inspection agency).

Variables with the *a-cyc* prefix are used if the *Cycle* inspection strategy is selected. The `a-cyc-lst-entity` variable contains a list of entities in the order in which they will be inspected, and the `a-cyc-lst-rule` contains the order in which the provisions will be inspected. The variable `a-cyc-count-entity` records the position (in the list `a-cyc-lst-entity`) of the entity that was last inspected, and the variable `a-cyc-count-rule` records the "position" of the rule that was last inspected (in the list `a-cyc-lst-rule`). The variable `a-cyc-slide` is used to record a subset of the rules that will be inspected at each step of the cycle. The variable `a-sus-sum-resources` contains the sum of all elements of the vector $\boldsymbol{d}$ (`gi-lst-resources`).

Some input parameters replace or depend on the values of other input parameters. Figure 4.7 displays the **input parameters interdependence matrix** of the ICARUS model simulation, which is based on the *Design Structure Matrix* (DSM) [63].

All input parameters are set via the user interface. The initial values of the parameters are an integral part of the .nlogo file.

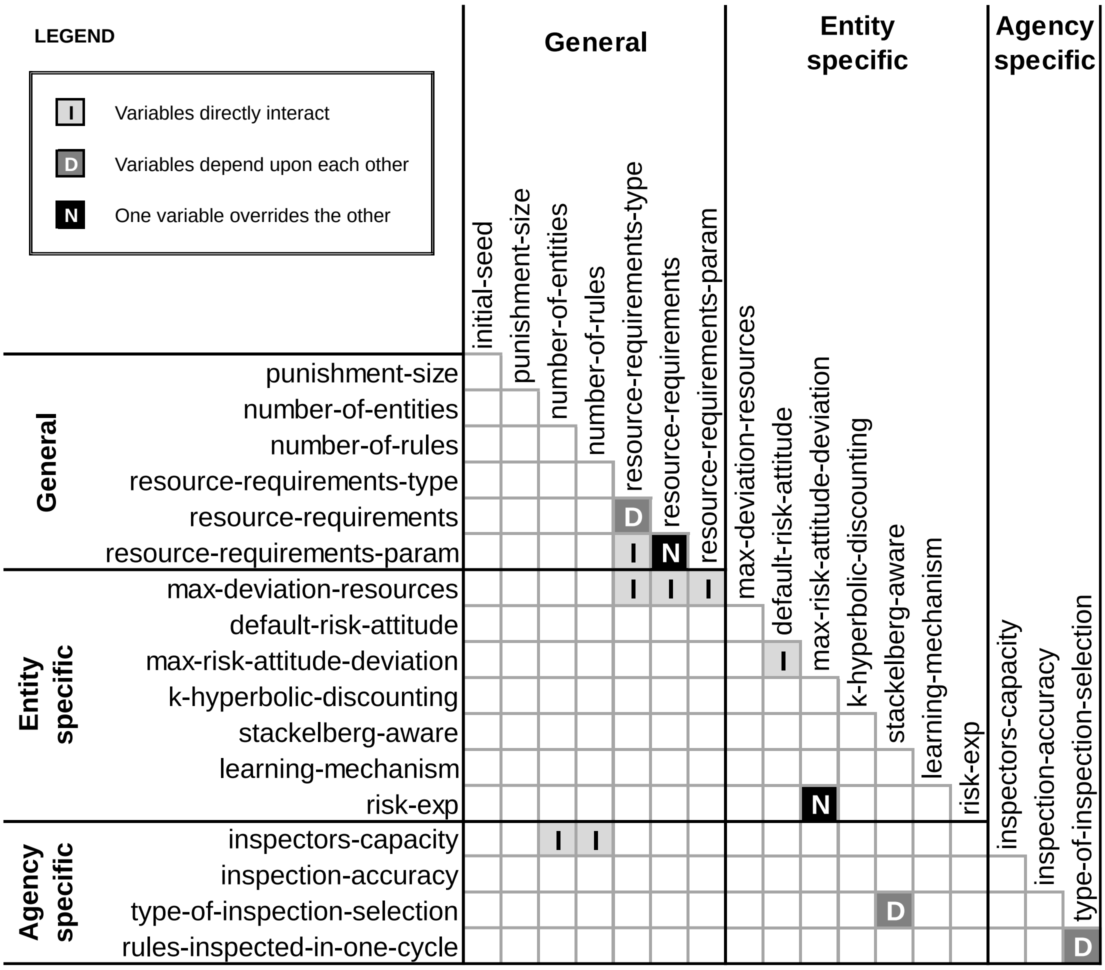

Figure 4.7 – Input parameters interdependence matrix of the ICARUS model

At the beginning of the simulation, the "world" is empty and is populated by agents through the `setup` procedure, based on the values of the input parameters.

4.3.3. User interface and the output values

The user interface was implemented with the goal of facilitating ease of use and understanding of the simulation results. Figure 4.8 shows the user interface.

The interface is visually divided into 3 parts: the input parameters of the simulation are entered on the left side, the central part of the interface graphically displays the results of the current simulation step, and the quantitative and descriptive statistical data about the simulation are on the right side.

The configuration (input) data is entered primarily via visual discrete bars – sliders (for the majority of input parameters), via menus with predefined options, and via fields for entering numeric values.

The central part of the interface graphically displays the progress of the simulation, with the entities, the inspection agency and the inspections being represented by different icons. At the beginning of the simulation (`setup`) the defined number of entities (`number-of-entites`) are randomly spatially positioned, wherein each entity has a unique numerical label. In each simulation step (`go`), inspection icons are positioned (spatially) to the relevant (selected) entity and inspection is performed. The colour of the inspection (inspector) indicates whether the inspection in question found compliance (green) or violation (red), and the shade of colour of every entity indicates how many rules that entity violates (darker red indicates that the entity violates more rules). The sequence number of the current simulation step is visible on the interface.

On the right side of the interface, the simulation results are displayed. In the lower central part are numerical indicators:

1. `rp-violations`: the total (accumulated) number of rule violations during the simulation; includes violations identified by the inspection, violations not recognized by the inspection, and violations not subjected to inspection.
2. `rp-compliance`: the total (accumulated) compliance with the rules during the simulation; includes compliance confirmed by an inspection as well as those not covered by inspection.
3. `rp-trn-violations`: the number of violations in the current simulation step (including those identified by an inspection and those not identified by an inspection).
4. `rp-trn-compliance`: the compliance with the rules in the current simulation step (including those confirmed by an inspection and those not confirmed by an inspection).
5. `rp-trn-inspct-viol`: the number of rule violations ascertained by inspections conducted in the current simulation step.
6. `rp-trn-inspct-comp`: the compliance with the rules confirmed by the inspections carried out in the current simulation step.

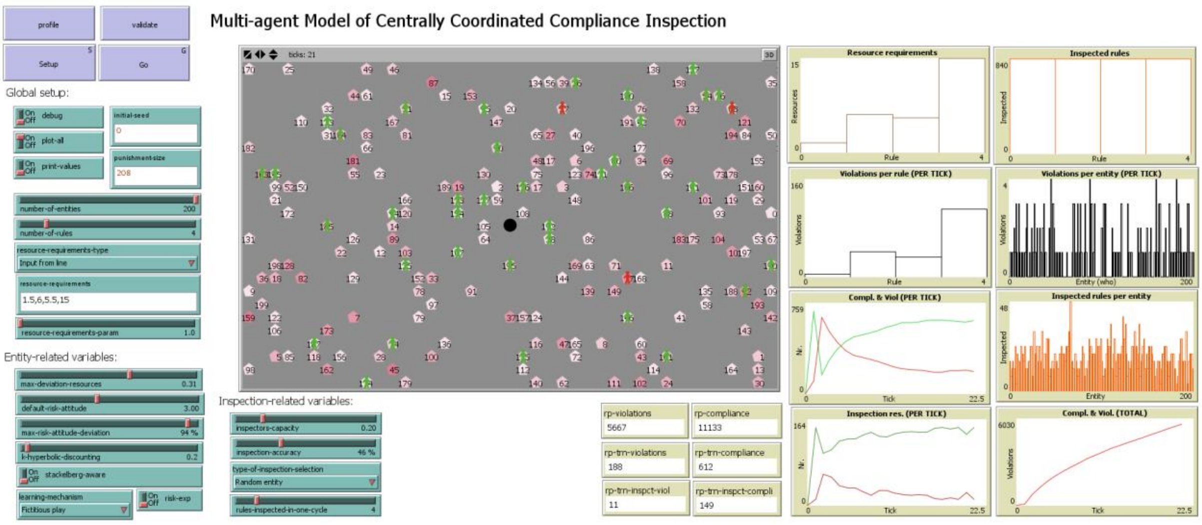

Figure 4.8 – User interface of the computer simulation of the ICARUS model

The presentation of the simulation results enables various insights and conclusions. For example, the results on the display (Figure 4.8) show that certain inspections in the current simulation step identified violations. Quantitative verification (variable `rp-trn-inspct-viol`) shows that there are 11 inspections. Also, the colour of the entity icons indicates that some entities violate more than one rule (this information is not known to the inspection agency).

The right side of the interface graphically displays the simulation results, with some diagrams referring to the current simulation step (marked *Per TICK* in the chart name), while others display the accumulated simulation results. Taking the situation shown on the display (Figure 4.8) as an example, it is possible to conclude the following:

1. The diagrams provide information on the comparison of the costs of compliance (resource requirements) and the number of violations of individual rules. The histogram of violations is visually similar to the histogram of costs of compliance.
2. The diagrams show that the number of violations decreases over time. Furthermore, large oscillations in the number of violations are visible at the beginning of the simulation, which are probably related to the application of the fictitious play algorithm, which is sensitive to the agent's initial assumptions.

The source code was analyzed in accordance with the recommendations in [64] and is available at the following address: https://github.com/ssmojver/DS.

4.3.4. Computer simulation environment and tools

The developed model was implemented into a computer simulation via the NetLogo environment [60]. NetLogo was chosen as the simulation environment due its ease of use, significant maturity of the environment, wide application in the research community, reasonable performance [65], availability of additional tools for analysis of developed models and easy integration with the programming language R [66].

Simulations that were performed with the goal of assessing the general validity of the model were initiated mainly via the *BehaviorSpace* module which is an integral part of the NetLogo environment and allows configuration and conduct of numerous computer simulations, for all possible combinations of input parameter values within given ranges (Cartesian product) [67, p. 289]. The specific validity of the model was analysed using the *BehaviorSearch* tool [68], which can be integrated with the NetLogo environment. *BehaviorSearch* allows the exploration of the parameter space of models implemented in NetLogo using several algorithms.

5. Model analysis

This chapter contains the description and results of the analysis of the ICARUS model and the computer simulation by which the model was implemented. The performed analysis includes estimation/calibration of the model parameters, model verification, general and specific validation, and sensitivity analysis.

Model **calibration**, in the context of agent-based models, is the process of finding (input) parameters for which the model (i.e. output data) has results that are acceptable, i.e. that correspond to reality [67]. In addition to calibration, the term **parameter estimation** is also used, since the calibration procedure includes setting up the model parameters and limiting the parameter space [69].

Model **verification** is the assessment whether the developed model corresponds to the target conceptual model, i.e. assessment whether the model is correctly implemented [67, p. 311]. Verification evaluates whether all relevant entities and relationships from the conceptual model are correctly transferred into a computer simulation [57, p. 98]. Verification can also be viewed as an "internal" validation of the model, which confirms that the model behaves as the authors intended [70]. Verification of a model may involve development of unit tests to test certain parts of the source code [103, p. 317].

Model **validation** is an assessment of whether the developed model explains a real phenomenon [103, p. 311]. Validation can also be described as a process of assessing how useful a developed model is for a particular purpose [33] . Empirical validation of the model involves estimating the extent to which the model represents the (unknown) process that generated the collected empirical data [71]. The validity of a model is the level of homomorphism between the developed system (model) and the system that this model represents [108]. Validation is not a process with the clear dichotomous result "validated" or "not validated" [67]. The validation results have to be interpreted taking into account the characteristics and limitations of the model, as well as the characteristics of the data used for validation. Approaches to model validation can be divided according to levels (micro-validation and macro-validation) and according to details (principle validation and empirical validation). Micro-validation assesses the behaviour of agents, and macro-validation assesses the behaviour of the system [67]. "Face" validation involves visual analysis of the model results [183].

Sensitivity analysis can be described as analysis of the impact of changes in model parameters on the model results [103, p. 323]. Sensitivity analysis determines how sensitive or robust the model is in relation to the initial parameters [103, p. 321]. It can identify the parameters that have the greatest impact on the model, which can facilitate the identification of the most significant processes in the model [73]. Sensitivity analyses of agent-based models can be divided into the following approaches [73][74][185]: global sensitivity analysis, local sensitivity analysis and screening methods. Screening methods rank parameters according to their (relative) importance, but cannot quantify differences in influences [75]. Local sensitivity analysis quantifies the impact of small variations in input parameters on the model results [73]. Global sensitivity analysis changes the input parameters over a wider range of values and quantifies their impact on the model results, while considering changes in multiple parameters simultaneously [75][73]. Morris' elementary effects screening method is particularly suitable for agent-based models and simulations [76]. A more detailed description of the Morris' method can be found in [76, pp. 110–112] and [73].

It should be emphasized that calibration (estimation) and validation of agent-based models are closely related. Likewise, model sensitivity analysis is often a key process within validation [77].

Indirect calibration [71][78] is a commonly used agent-based model analysis method which, despite its name, consists of 4 steps which include calibration, verification, and model validation:

1. In the first step, the modeller determines the so-called "stylized facts" (which, for example, are observed in reality) that should be explained or reproduced by the developed model.
2. In the second step, the model is developed. The modeller is trying to achieve that the micro-states of the model are as similar as possible to the available empirical and experimental data.

3. In the third step, the parameter space of the model is limited by applying empirical data – for example, by applying Monte Carlo analysis.
4. The last step includes the analysis of parameters, i.e. the analysis of the sensitivity of the model.

Bloomquist describes [53, p. 136] 4 levels of validation of the agent model (following the proposal of Axtell and Epstein), where each subsequent level provides a higher level of confidence in the validity of the model:

0. level: The model is a "caricature" of reality. Models validated at level 0 show only the basic characteristics of reality. For example, the direction of change is identical in the model and in reality.
1. level: The model is qualitatively consistent with the empirical macro-structures. For example, the basic characteristics of the populations in the model are consistent with the characteristics of the populations in reality.
2. level: The model is quantitatively aligned with the empirical macro-structures. For example, a model can generate quantitative results that are comparable to actual results.
3. level: The model is quantitatively aligned with the empirical micro-structures. In practice, this would mean that the model can predict the behaviour of individual agents.

The analysis of the ICARUS model was performed via a combination of parameter estimation, verification, validation and sensitivity analysis methods, primarily by recording the results of repeated simulations. The performed verification procedures include comparison of qualitative and quantitative results of the simulation with the expected results. The micro and macro states of the model, the behaviour of individual agents and groups of agents, and the behaviour of systems, entities, inspection agency, and inspections were all considered in this process.

Model validation mostly follows the indirect calibration method, and is divided into general and specific validation. **General validation** primarily assesses the quality of the model's alignment with the empirical macro-structures (level 1), which primarily includes "face" validation and sample comparison. Within the **specific validation** the extent to which the model is quantitatively consistent with the empirical macro-structures (level 2) is assessed for 3 specific cases and the simulated data are compared with actual, historical data.

Based on the (secondary) empirical data on compliance inspections the initial estimation of parameters was performed, i.e. the initial calibration of the developed multi-agent model [67][73]. Verification and validation of the model were performed by structural verification, i.e. by checking the consistency of the model with the relevant description of the system (parts) and by comparing the results of the model at the macro and micro level with secondary empirical data on compliance inspections. The "face" validation further included visual analysis of the model results [12], and was conducted based on the empirical research insights, using various data visualization techniques. Quantitative data on environmental inspections in Denmark [41][37], occupational safety inspections in the USA [32] and on banking supervision in Italy [29][79] was compared with the results of computer simulations. The parameter space search was performed within the specific validity assessment using genetic algorithm [80, p. 77]. The sensitivity analysis of the model parameters was performed using the Morris's elementary effects screening method [134] and, among others, included variation of seeds, stochastic elements and model parameters.

The exact procedures that were used in verification, validation and sensitivity analysis of the model are described below.

Statistical analysis, data visualization and part of the sensitivity analysis were performed using the R programming language [81] and RStudio environment [82]. The following R packages were used: plyr [83], polycor [84], tidyverse [85], reshape2 [86], gridExtra [87], MASS [88], RNetLogo [66], sensitivity [89] and RColorBrewer [90].

5.1. Model verification

The verification of the model is divided into 3 parts: verification of the characteristics of the system, verification of the behaviour of the entities and verification of the behaviour of the inspection agency and inspections. The micro and macro states of the model, the behaviour of individual agents and groups of agents, and the behaviour of systems, entities, inspection agencies, and inspections were all considered in that process. Since verification assesses the correctness of the implementation, verification tests were generally not repeated many times because there was no need for overriding the stochastic nature of the ICARUS model.

5.1.1. System parameters

5.1.1.1. Verification test: Generating costs of compliance

The verification test aims to determine whether the compliance costs are generated according to the uniform or exponential distribution, and whether they are within the given limits.

The test was performed via a computer simulation of the model implemented in NetLogo environment, with the distribution of compliance costs given as a random variable, with a uniform or exponential distribution of values. Three ranges of possible values of the random variable for uniform distribution ([0,1], [0,3] and [0,5]) and 3 values of the arithmetic mean of the exponential distribution (1, 3 and 5) were tested. The simulation encompassed 100 rules (provisions). 100 rules imply the existence of 100 different areas of inspection, which is not realistic, but a large number of rules allows for a meaningful display of data on the quantile-quantile (QQ) chart and a comparison of the obtained results with the expectations.

The test was performed with the input parameters contained in the table below (Table 5.1). The values of other input parameters do not affect the test results since only the parameter generation was tested and the test was performed in the simulation setup phase.

Table 5.1 – Input parameters: compliance costs generation

Input parameter:	Values:
`resource-requirements-type`	`{Uniform distribution, Exponential distribution}`[15]
`resource-requirements-param`	`[1 3 5]`[16]
`number-of-rules`	`100`

Figure 5.1 shows QQ graphs of compliance costs generated randomly, with a uniform distribution. The results are in line with the expectations, i.e. the resulting graphs show that the random numbers generated are within the expected ranges and their distribution does not differ from the distribution expected for the random variable with a uniform distribution. It should be noted that – due to the limitations of the user interface in the configuration of the computer simulation – the resource requirements after generation were rounded to one decimal, which affected the grouping of the obtained data on QQ charts.

[15] Parameter values in curly braces represent a list of different values of the input parameter with which the simulation was started. In this case, the simulation was started with 2 values of the variable `resource-requirements-type`: `Uniform distribution` and `Exponential distribution`.

[16] The values of the parameters in square brackets separated by a spaces represent the vector of values that is passed as input parameter.

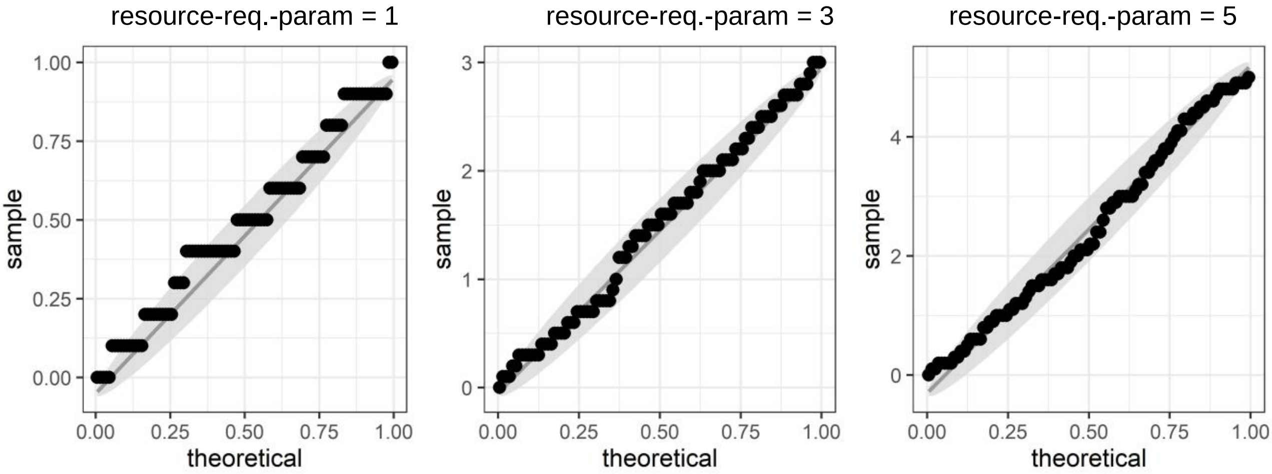

Figure 5.1 – Model verification: QQ graphs of compliance costs generated with uniform distribution, for the given value of the input parameters; the shaded area is 95% confidence interval.

Figure 5.2 displays QQ graphs of randomly generated compliance costs, via exponential distribution. The results are in line with expectations, i.e. the resulting graphs show that the generated random numbers do not differ, by values and distributions, from those expected for a random variable with exponential distribution of random values, for given arithmetic means.

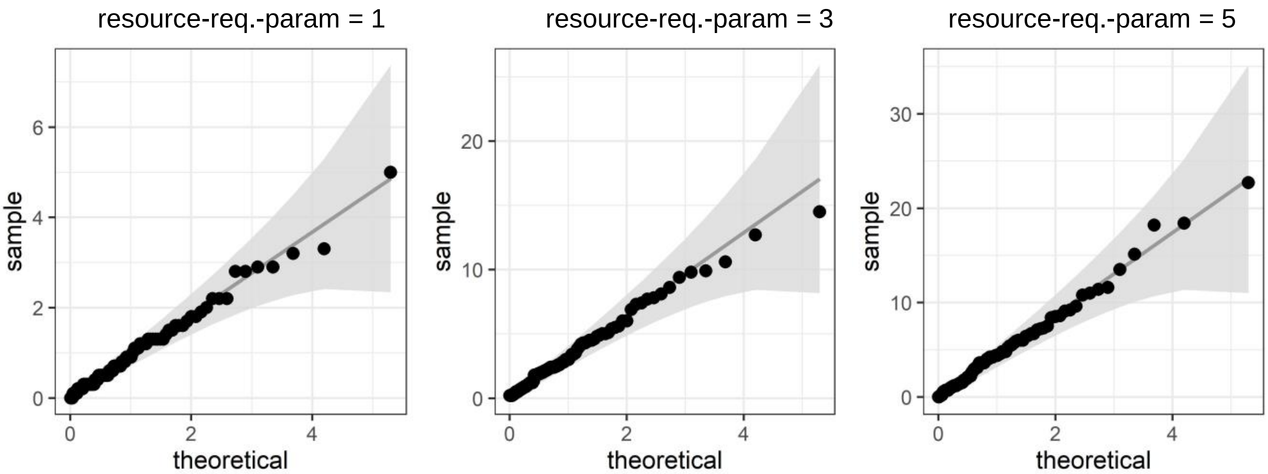

Figure 5.2 – Model verification: QQ graphs of generated compliance costs with exponential distribution, for the given value of the input parameter; the shaded area is 95% confidence interval

5.1.2. Entity behavior

5.1.2.1. <u>Verification test: Differences in entities' compliance costs</u>

The verification test aims to determine whether the deviation of the entities' compliance costs in relation to the "global" compliance costs is in line with expectations.

The test was performed via a computer simulation of the model implemented in NetLogo environment, in a simple configuration with only two rules that have compliance costs of 2 and 5, and the maximum deviation of compliance costs is 0.2 and 0.4. Given the values of these parameters, it is expected that the distribution of compliance costs will be a random variable with a uniform distribution of values in ranges $2 \pm 20\%$ and $5 \pm 20\%$, $2 \pm 40\%$ and $5 \pm 40\%$. 200 entities were generated in the simulation to enable display of data on the QQ chart and comparison of the obtained and expected results.

The test was performed with the input parameters contained in the table below (Table 5.2). The values of other input parameters do not affect the test results since only the parameter generation was tested and the test was performed in the simulation setup phase.

Table 5.2 – Input parameters: variations in compliance costs

Input parameter:	Values:
max-deviation-resources	{0.2, 0.4}
resource-requirements	[2 5]
number-of-rules	2
number-of-entities	200

Figure 5.3 shows QQ graphs of compliance costs generated randomly, with a uniform distribution. The results are in line with the expectations, i.e. the resulting graphs show that the random numbers generated are within the expected ranges and their distribution does not differ from the distribution expected for the random variable with a uniform distribution.

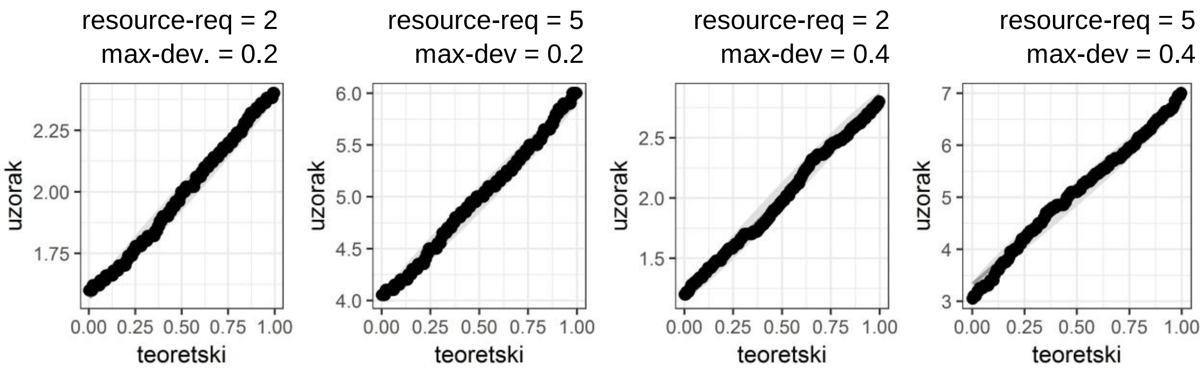

Figure 5.3 – Model verification: QQ graphs of generated compliance costs with uniform distribution of random variable, for the given input parameters; the shaded area is 95% confidence interval.

5.1.2.2. Verification test: Risk appetite

The verification test aims to determine whether the entity's risk attitude is generated correctly according to a uniform or exponential distribution, and whether the generated values are within the given limits. In both variants of the test, 200 entities were generated. The test was performed via a computer simulation of the model implemented in NetLogo environment, with the distribution of risk appetite given as a random variable, with a uniform or exponential distribution of random values.

The underlying risk preferences of the entities in the uniform distribution test have values of 1 or 5, and the maximum deviation from the underlying risk preference is 30% or 60%. Given the values of these parameters, it is expected that the distribution of the entity's risk appetite will be a random variable with a uniform distribution of values in the ranges $1 \pm 30\%$, $1 \pm 60\%$, $5 \pm 30\%$ and $5 \pm 60\%$.

The underlying risk preferences of the entities in the exponential distribution test should be a random variable with an exponential distribution of random values, with arithmetic means of 0.5, 1 and 5.

The test was performed with the input parameters in tables Table 5.3 and Table 5.4. The values of other input parameters do not affect the test results since only the parameter generation was tested and the test was performed in the simulation setup phase.

Table 5.3 – Input parameters: generating the underlying risk preferences with uniform distribution of random variable

Input parameter:	Values:
risk-exp	OFF
default-risk-attitude	{1, 5}

max-risk-attitude-deviation	{30%, 60%}
number-of-entities	200

Table 5.4 – Input parameters: generating the underlying risk preferences with exponential distribution of random variable

Input parameter:	Values:
risk-exp	ON
default-risk-attitude	{0.5, 1, 5}
number-of-entities	200

Figure 5.4 displays QQ graphs of randomly generated risk appetites of entities with uniform distribution. The results are in line with expectations, i.e. the resulting graphs show that the random numbers generated are within the expected ranges and their distribution does not differ from the distribution expected for a random variable with a uniform distribution.

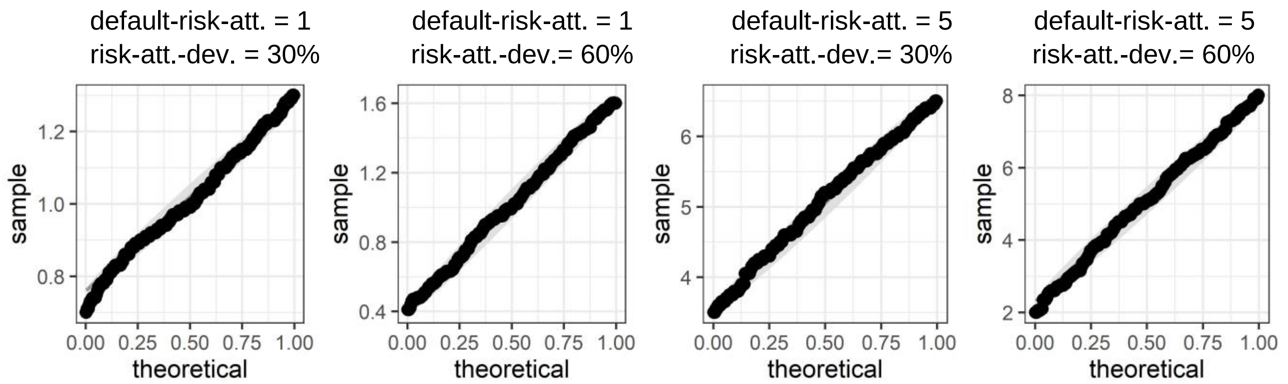

Figure 5.4 – Model verification: QQ graphs of generated risk-taking preferences with uniform distribution, for the given input values; the shaded area is 95% confidence interval

Figure 5.5 displays QQ graphs of randomly generated risk appetites with exponential distribution. The results are in line with expectations, i.e. the resulting graphs show that the generated random numbers do not differ, by values and distributions, from those expected for a random variable with exponential distribution of random values, for given arithmetic means.

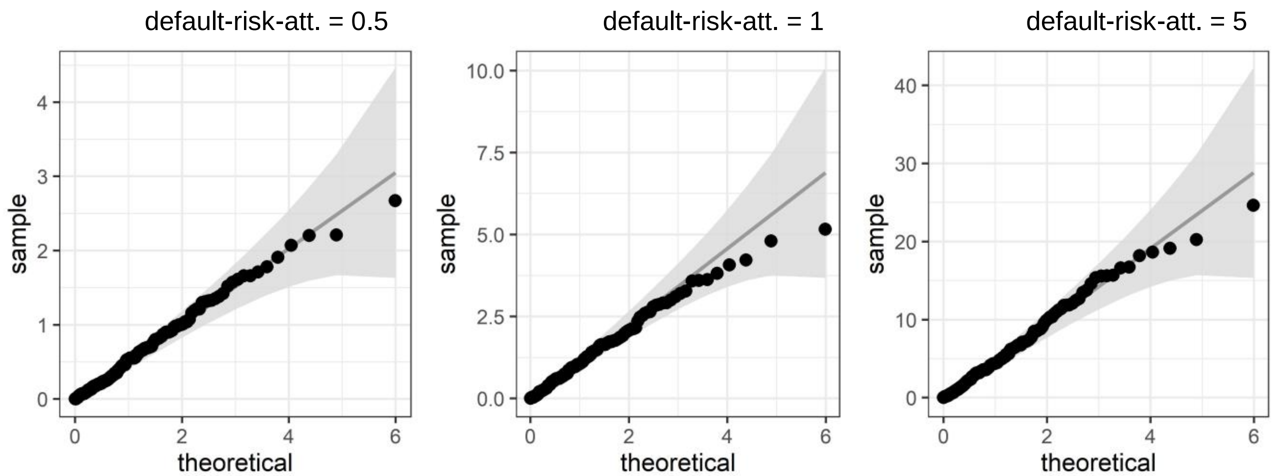

Figure 5.5 – Model verification: QQ graphs of generated risk-taking preferences with exponential distribution, for the given input values; the shaded area is 95% confidence interval

5.1.2.3. Verification test: Entity learning strategy – fictitious play

The verification test aims to determine whether the entities are learning in accordance with the fictitious play model. In the fictitious play, the agents adjust their compliance/violation strategy in each step according to their own knowledge (in that step). In the context of the ICARUS model, this means that, after each simulation step, the entities supplement the inspection history information with the information from the last step and calculate a new assessment of the probability that they will be inspected in the next step. If the inspection agency implements a random selection strategy, and if the entities remember the history of inspections perfectly, it is expected that over time (in line with the law of large numbers) their assessment of probability of inspection will converge to the actual probability of inspection. This test verifies that expectation.

The implementation of the fictitious play was assessed in a simple 10-entity configuration. The inspection agency uses the random entity selection strategy. The capacity of the inspection agency is 0.2. Therefore, the inspection agency will, in each move, direct the inspection of compliance of all rules in 2 randomly selected entities. The entities are completely risk-neutral and there is no discounting of history. Other simulation parameters are not significant for this test. After setting up the simulation, 200 simulation steps were performed. In each step of the simulation, the perception of each entity about the probability of inspection was recorded and graphically presented, which allows for the comparison of achieved and expected results.

The test was performed with the input parameters in the Table 5.5. The values of the other input parameters do not affect the test results, and the results were recorded for all the entities, after 200 simulation steps.

Table 5.5 – Input parameters: fictitious play learning strategy

Input parameter:	Values:
`type-of-inspection-selection`	`Random entity`
`inspectors-capacity`	`0.2`
`number-of-entities`	`10`
`default-risk-attitude`	`1`
`max-risk-attitude-deviation`	`0%`
`k-hyperbolic-discounting`	`0`

Figure 5.6 graphically displays the test results.

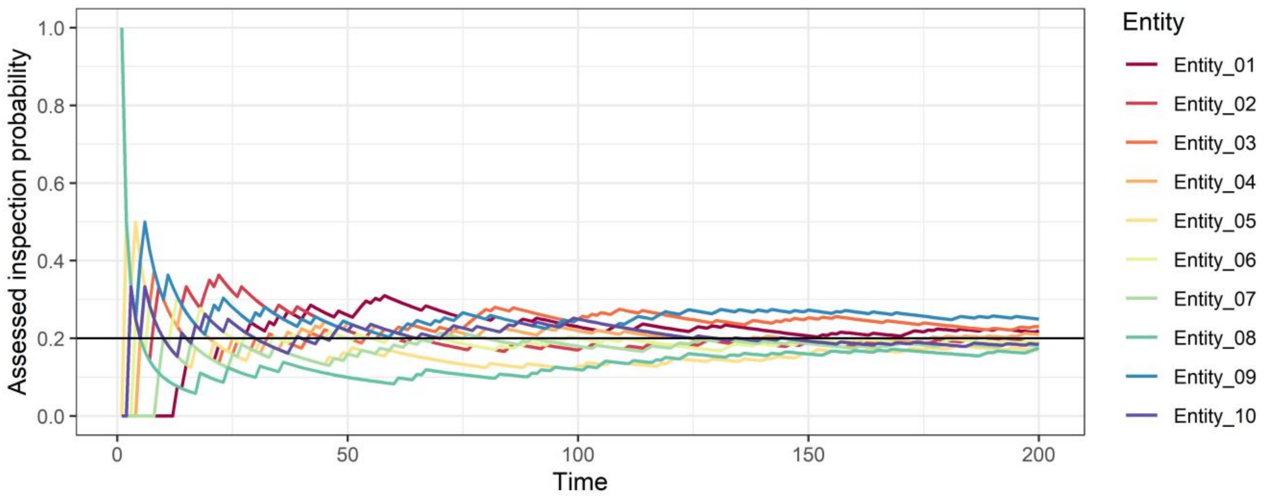

Figure 5.6 – Model verification: change in the entities' perception of probability of inspection with the actual probability of inspection of 0.2; the x-axis displays the simulation steps, and the y-axis displays the perceived probability of inspection; the coloured lines display the perception of each entity about the probability of inspection (in each step), the black horizontal line shows the actual inspection probability.

As expected, the entities start with large variations in the perceived probability of inspection – that is, with the perception of the probability of inspection 1 (there will be inspections) or 0 (there will be no inspections). By collecting additional data (gaining experience), the perceived probability increasingly converges towards the actual probability of inspection (0.2), which was expected.

5.1.2.4. Verification test: Time discounting

The verification test aims to determine whether the time discounting mechanism is properly implemented. Time discounting, in the context of the developed model, means that entities attach less importance to experiences from the distant past compared to experiences from the more recent past. The impact of past experiences weakens over time, with the rate of this reduction depending on the discounting coefficient κ. The test assesses whether the impact of past experience on the perception of the probability of inspection decreases in line with the set value of the coefficient κ.

Verification of the implementation of the time discounting mechanism was performed on a customized computer simulation of the model, to ensure a deterministic outcome of inspections for a pre-selected entity. After setting up the simulation, 100 simulation steps were completed. The selected entity was inspected in every of the first 50 steps and never in the last 50 steps of the simulation. Since entities learn according to the fictitious play model, the selected entity's perception of the probability of inspection is expected to be 1, in the first 50 steps. After that step, it should start decreasing, with the rate of decrease dependant on the value of the coefficient κ. 7 repetitions of the simulation were performed, for different values of κ. The entities remember the entire history of inspections, and are risk-neutral. After setting up the simulation, 100 simulation steps were performed, and in each step, the selected entity's perception of the probability of inspection was recorded.

The test was conducted with the input parameters contained in the table below (Table 5.6). The values of other input parameters do not affect the test results. After configuring the simulation for each of the 7 possible values of the `k-hyperbolic-discounting` variable, 100 simulation steps were performed, and the results of the selected entity's perceptions of the probability of inspection were recorded, in each simulation step.

Table 5.6 – Input parameters: time discounting

Input parameter:	Values:
learning-mechanism	Fictitious play
default-risk-attitude	1
max-risk-attitude-deviation	0%
k-hyperbolic-discounting	{0, 0.1, 0.2, 0.5, 1, 2, 5}

Figure 5.7 graphically shows the simulation results. The perception of the inspection probability[17] for all 7 simulations up to the step 50 was 1, after which it began to decrease, with the faster decline for higher values of the discount coefficient κ. Visually, the results of the simulations are in line with the expectations. The results of the numerical test are shown in the table below (Table 8.7).

[17] The initial expected inspection probability (in step 1) is not shown as it is random (there is no history of inspections in the step 1).

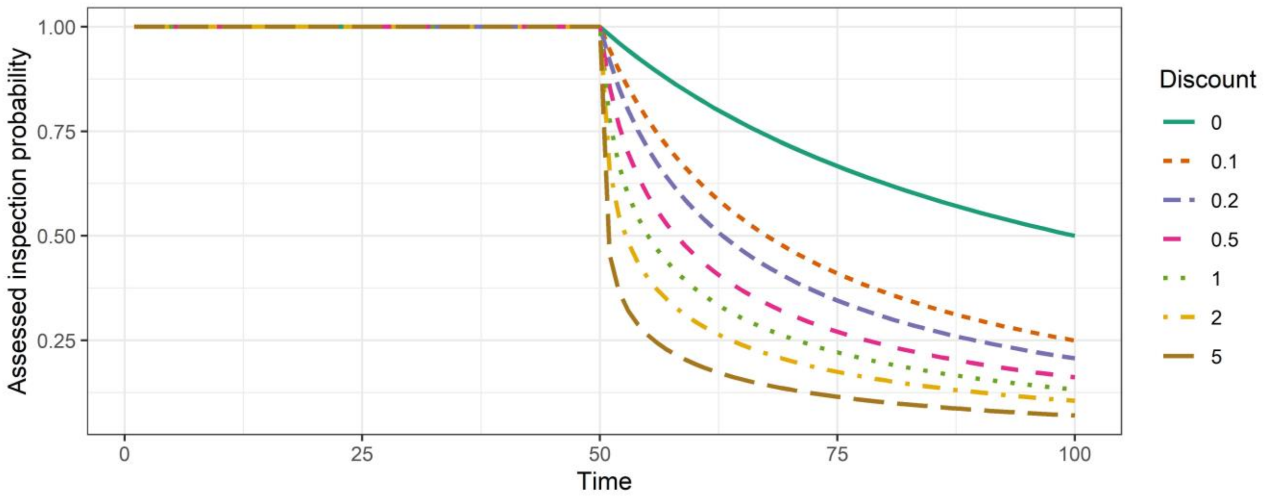

Figure 5.7 – Model verification: change of the entity's assessment of the inspection probability for a given value of the time discount coefficient in each step of the simulation; the x-axis displays the simulation steps and the y-axis the probability of conducting an inspection

Table 5.7 – Model verification: perceptions of a selected entity about the probability of inspection, recorded in a given simulation step (t) for a given value of the time discount coefficient (κ)

t	κ = 0.1	κ = 0.5	κ = 1
60	0.639	0.454	0.374
70	0.467	0.312	0.256
80	0.365	0.239	0.195
90	0.297	0.193	0.158
100	0.253	0.164	0.135

All values recorded in the table are equal to the expected values, i.e. the values obtained by including the corresponding data in the hyperbolic discounting formula (hyperbolic function). For example, for $\kappa = 0.5$ and $t = 90$, the entity remembers the previous 90 moves, being supervised in the first 50 moves and not in the next 40 moves. The perception of the entity about the probability of inspection in a given step can be calculated by normalizing the sum of the discounted experiences of the entities about the conducted inspections, i.e.:

$$V = \frac{\sum_{t=41}^{90} \frac{1}{1 + 0.5 * t}}{\sum_{t=1}^{90} \frac{1}{1 + 0.5 * t}} = \frac{1.581253}{8.187119} = 0.193139 \quad (22)$$

This result, taking into account the accuracy of the calculation that is achievable in the simulation, is equal to the relevant result in the table (Table 8.7) .

The results of the tests indicate that the time discounting mechanism is correctly implemented in the ICARUS model.

5.1.2.5. Verification test: Entity's awareness of inspections with Stackelberg leadership

Stackelberg leadership, in the context of the ICARUS model, means that the inspection agency has – in some credible way – announced to the entities that it will conduct inspections with frequency in proportion to the costs of achieving and maintaining compliance with the provisions. That is, the entities, when assessing the likelihood of an inspection of compliance with a specific rule, assume that the frequency of inspections of that rule is proportional to the costs of compliance with the rule. The verification test aims to determine whether the mechanism for setting the entity's perception that the frequency of inspection of an individual rule is proportional to the costs of compliance is properly implemented.

The expectation of the test is that in a configuration without Stackelberg leadership, the number of violations will be commensurate with the costs of compliance. In contrast, in the configuration with Stackelberg leadership, the number of violations should be identical for all rules. Namely, the inspection agency uses the stochastic universal sampling strategy, i.e. the probability of inspection of each provision is proportional to the compliance costs. The simulation includes 10 entities, which must comply with 5 rules. The compliance costs are explicitly defined[18]. To facilitate comparison and interpretation of the results, the entities are set to be risk-neutral. Furthermore, the costs of compliance with each individual rule are the same for all entities, which in turn is identical to the inspection agency's assumptions about the costs of compliance with each provision, i.e. $\boldsymbol{c_i} = \boldsymbol{d}, \forall i \in \mathcal{E}$. In addition, the entities do not discount historical data, and learning is consistent with the fictitious play algorithm. The penalty for violation[19] is 38, and the capacity of the inspection agency is 0.25. Other simulation parameters are not relevant for this test.

Verification of the correctness of the implementation of Stackelberg leadership was tested by comparing the number of violations of each individual rule after 100 simulation steps, for two configurations of input parameters. The only difference in configurations is the activation of the Stackelberg leadership parameter. To enable the direct comparison of results in both configurations, an identical (randomly generated) initial seed was set. After setting up the simulation, 100 steps were performed.

The test was started with the input parameters contained in the table below (Table 5.8). The values of other input parameters do not affect the test results. After setting up the simulation for each of the 2 possible values of the `stackelberg-aware` variable, 100 simulation steps were performed, and the number of violations of each rule was recorded.

Table 5.8 – Input parameters: entity's awareness of Stackelberg leadership

Input parameter:	Values:
type-of-inspection-selection	Stochastic universal sampling
number-of-entities	10
number-of-rules	5
resource-requirements-type	Input from line
resource-requirements	[3 5 7 9 11]
default-risk-attitude	0
max-risk-attitude-deviation	0%
max-deviation-resources	0
k-hyperbolic-discounting	0
learning-mechanism	Fictitious play
punishment-size	38
inspectors-capacity	0.25
stackelberg-aware	{ON, OFF}
initial-seed	991345273

Figure 5.8 displays the simulation results. The first bar chart (a) displays the compliance costs of each rule (chart from the NetLogo simulation[20]). The next 2 bar charts display the number of violations of each rule in a configuration without Stackelberg leadership (b) and with Stackelberg leadership (c). The results show that entities violate low-cost rules less than high-cost rules if they believe that each rule is equally likely to be subjected to compliance inspection. On the other hand, if entities consider that probability of inspecting the rules is proportional to the compliance costs, then the number of violations of all rules is identical.

18 The values in the vector were chosen arbitrarily, taking into account the desired visualization of the simulation results. The specific amounts are not significant for the verification results.

19 The penalty was set based on the previous simulation results, given the defined compliance costs, with the aim of easily identifying the expected patterns in the simulation results.

20 Bar charts created through computer simulation in NetLogo resemble histograms due to the limitations of the environment itself. That is, in a NetLogo environment, it is not possible to set a space between columns in a chart.

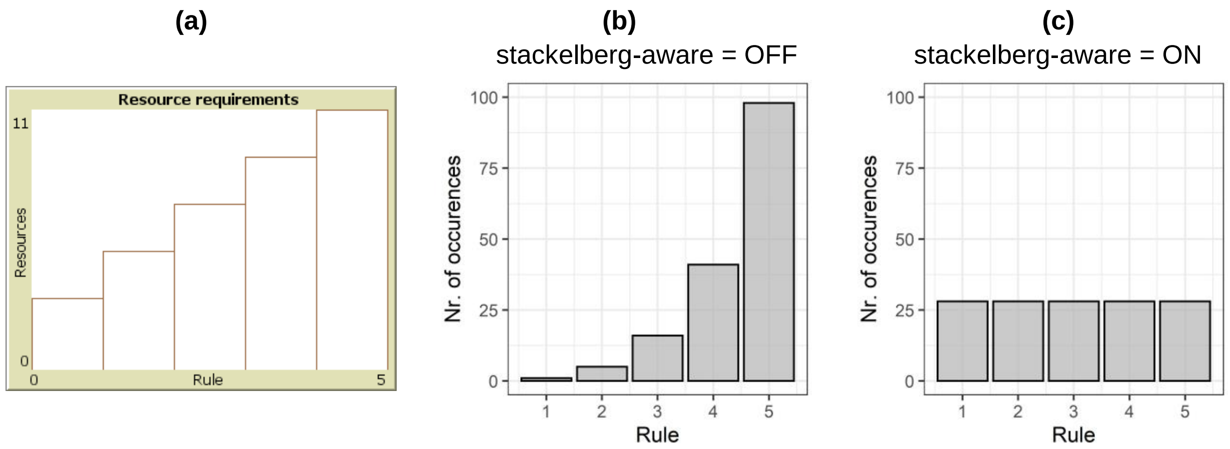

Figure 5.8 – Model verification: comparison of the given compliance costs (resource requirements) of fulfilling each rule (a) with the number of violations of each rule in the configuration without Stackelberg leadership (b) and with Stackelberg leadership (c)

It should be noted that the total number of violations is not necessarily lower than in the previous situation. That is, the number of violations of rules with low compliance costs is now higher, given that the entities believe that the probability of inspecting the compliance of these rules is very small.

From the presented results it follows that the simulation results are in line with the expectations.

5.1.3. Agency behaviour

This chapter presents the results of verification tests of the agency's behaviour, which primarily includes assessment whether the inspection strategies have been implemented in the computer simulation in a correct way.

5.1.3.1. <u>Verification test: Random selection strategy</u>

The test aims to determine whether the random selection strategy (*Random*) is correctly implemented in the computer simulation. The random selection strategy should randomly select the entities and rules for inspection, i.e. each entity and each rule should have exactly the same probability of being selected for inspection. Therefore, as the number of simulation steps increases, the number of inspections of each entity should – in line with to the law of large numbers – converge to the same value, dependant on the capacity of the inspection agency. The same goes for the number of inspections of each rule.

Assessment of the random selection strategy implementation was conducted in an environment with 50 entities that should comply with 5 rules. The inspection agency uses a random selection strategy, and the inspection capacity is 0.2. Other input parameters are not relevant for this test and are set to arbitrary values. After setting up the simulation, 100 steps were performed. Verification of the correctness of the implementation of the random selection strategy was tested in the following way: after the 1st, 5th, 25th and 100th simulation step, the cumulative number of inspections of each individual rule (for all entities) and cumulative number of inspections of each individual entity (for all rules) were recorded.

Verification of the correctness of the implementation of the inspection strategy was performed by reviewing the data generated on the graphs in the NetLogo simulation. Given the input parameters, it is expected that – according to the law of large numbers – after 100 steps, on average, an individual rule will be monitored 1,000 times, and an individual entity 100 times. The test was ran with the input parameters contained in the table below (Table 5.9).

Table 5.9 – Input parameters: random selection strategy implementation

Input parameter:	Values:
type-of-inspection-selection	Random
number-of-entities	50
number-of-rules	5
inspectors-capacity	0.2

Figure 5.9 shows the test results. It is visible from the graphs that the simulation results are in line with the expectations, i.e. after 100 steps, on average, every individual rule was monitored approximately 1,000 times, and every individual entity 100 times. Additionally, the data on the graphs in the first row (the number of inspections of individual rules) converge faster towards the expected result due to 10 times more events.

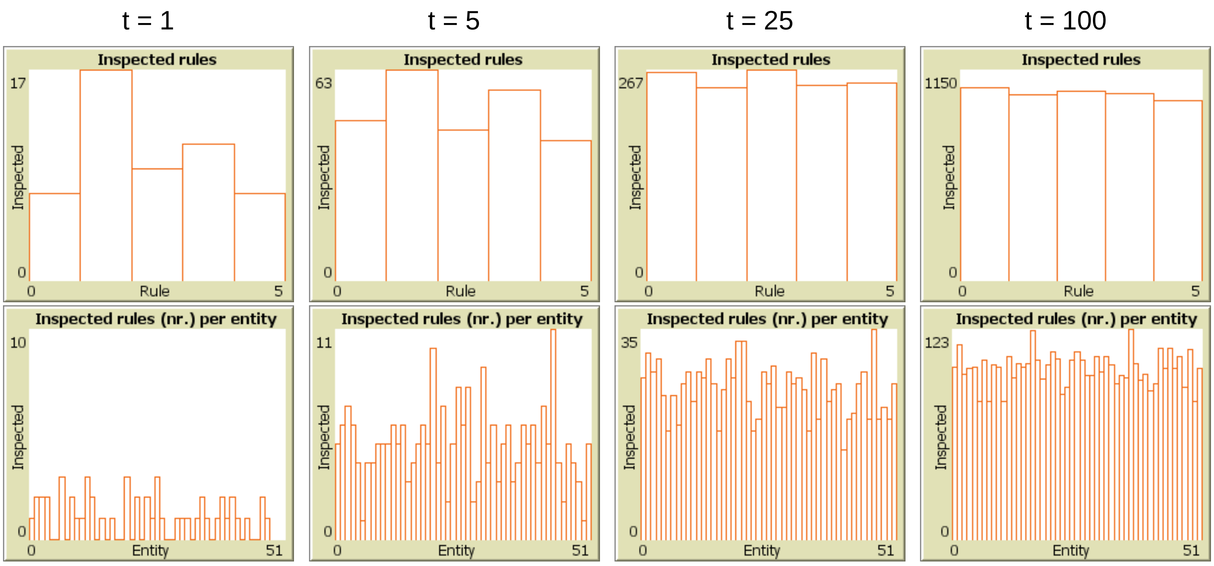

Figure 5.9 – Model verification: number of inspections of each rule (1st row) and number of rules inspected in each entity (2nd row), after the stated number of simulation steps, with the random selection inspection strategy

5.1.3.2. <u>Verification test: Random entity selection strategy</u>

The goal of this test is to assess whether the Random entity selection strategy (*Random entity*) is implemented correctly in the computer simulation. The strategy should randomly select entities for inspection and perform compliance inspection with every provision that this entity must fulfil. Therefore, each entity should have the same probability of being inspected at every step of the simulation, and this probability is equal to the capacity of the inspection agency. As the number of simulation steps increases, the number of inspections of each entity should – in accordance with the law of large numbers – converge to the same value (inspection agency capacity * simulation step). This convergence should be slower than when applying the random selection strategy, due to the lower available capacity of inspectors – given that in the random entity selection strategy inspections cover all the rules in the inspected entity.

Verification of implementation of the random entity selection strategy was performed in an environment with simulation parameters identical to those in the previous test, except for the parameters of the inspection agency's strategy. After setting up the simulation, 1,000 simulation steps were performed. Correctness was tested by recording the (cumulative) number of inspections in each individual entity.

Verification of correctness of this inspection strategy was performed by reviewing the data generated on the graphs in the NetLogo simulation. Given the input parameters, it is expected that – in accordance with the law of large numbers – after 1,000 steps, 1,000 rules (on average) will be inspected at an individual entity.

The test was ran with the input parameters contained in the table below (Table 5.10). The values of other input parameters do not affect the results. After setting up the simulation, 1,000 steps were recorded, and the number of inspected rules per entity was recorded after the 1st, 2nd, 3rd, 4th, 5th, 6th, 100th, and 1,000th simulation step.

Table 5.10 – Input parameters: verification of the random entity selection strategy implementation

Input parameter:	Values:
type-of-inspection-selection	Random entity
number-of-entities	50
number-of-rules	5
inspectors-capacity	0.2

Figure 5.10 displays the results of the test. The graphs show the number of rules inspected in each entity, after the 1st, 2nd, 3rd, 4th, 5th, 6th, 100th, and 1,000th simulation step. The first 6 charts show that each inspection covers 5 rules in an entity (all rules) in one step. The last 2 graphs show that over time, the number of inspections of each entity converges – according to the law of large numbers – to the same (expected) value. That is, after 1,000 steps, on average, each provision will be covered by inspections 1,000 times, and an entity will be inspected 100 times. The presented results are in line with the expectations.

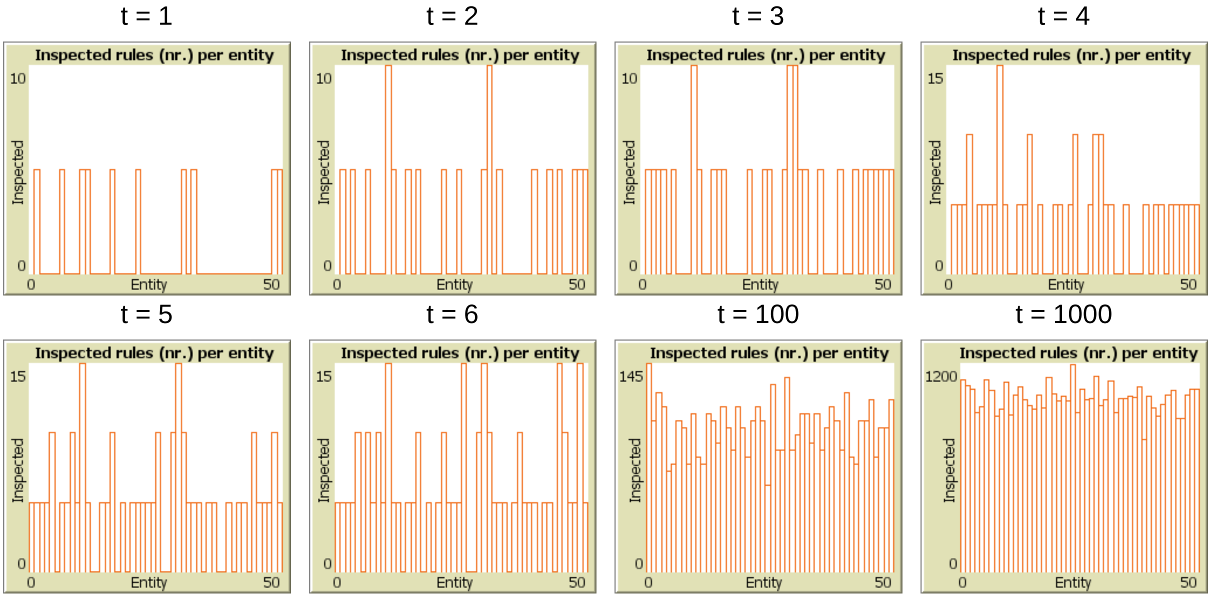

Figure 5.10 – Model verification: number of inspections of each rule and number of rules inspected in each entity, after the specified number of simulation steps, with the random entity selection strategy

5.1.3.3. <u>Verification test: Cyclical selection strategy</u>

The verification test aims to determine whether the cyclical selection strategy for inspection of entities and rules is properly implemented in the computer simulation. The application of the cyclical selection strategy ensures that the inspection agency will inspect the compliance of all the entities with all the rules within a given time period.

Verification of correctness of implementation of the cyclical selection strategy was conducted in an environment with 10 entities that must comply with 6 provisions. The inspection agency uses a cyclical selection strategy and 3 rules are inspected in one cycle. The capacity of the inspection agency is 0.20. Other simulation parameters are not relevant for this test and are set arbitrarily. In line with the defined parameters, the total number of entity-rule combinations is 60, and the inspection agency can inspect 12 entity-rule combinations in one step, first inspecting 3 randomly selected rules in all the entities, and then inspecting the next 3 rules. Thus, in the 1st step, the inspection agency should inspect 3 rules in 4 randomly selected entities. In the 2nd step, it should inspect the same 3 rules in 4 other, randomly selected entities. In the 3rd step, the inspection agency should conduct inspection of those 3 rules in the 2 remaining entities, and inspection of the other 3 rules in 2 randomly selected entities. I.e., after 5th step, the inspection cycle should be completed.

Verification of correctness of implementation of the cyclical selection strategy was tested in the following way: after the 1st, 2nd, 3rd, 4th and 5th simulation step, the cumulative number of inspections of each individual rule (for each entity) and the cumulative number of inspections of each individual entity (inspection of all rules) were recorded. The verification was performed by reviewing the graphs generated in the simulation in NetLogo and comparing the results with expectations.

The test was ran with the input parameters contained in the table below (Table 5.11). The values of other input parameters do not affect the test results. After setting up the simulation, 5 simulation steps were performed, and rules and entities that were inspected were recorded, after each simulation step.

Table 5.11 – Input parameters: cyclical selection strategy implementation

Input parameter:	Values:
type-of-inspection-selection	Cycle
number-of-entities	10
number-of-rules	6
rules-inspected-in-one-cycle	3
inspectors-capacity	0.2

Figure 5.11 shows the following test results:

- 1st step: Provisions 1, 3 and 6 were inspected in entities 2, 5, 6 and 8;
- 2nd step: Provisions 1, 3 and 6 were inspected in entities 1, 3, 7 and 10;
- 3rd step: Provisions 1, 3 and 6 were inspected in entities 4 and 9, Provisions 2, 4 and 5 were inspected in entities 2 and 8;
- 4th step: Provisions 2, 4 and 5 were inspected in entities 3, 5, 6 and 10;
- 5th step: Provisions 2, 4 and 5 were inspected in entities 1, 4, 7 and 9, which completes the inspection cycle.

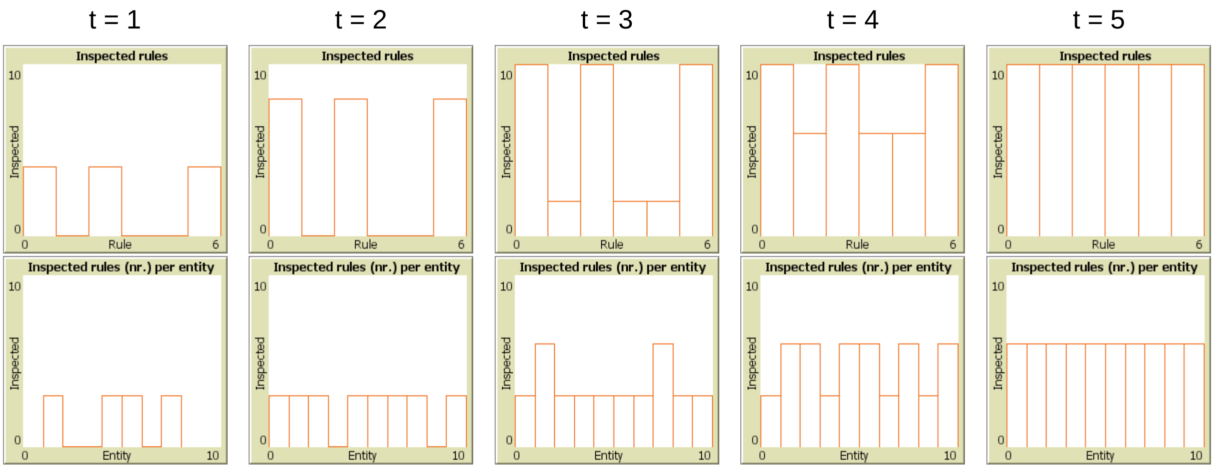

Figure 5.11 – Model verification: number of inspections of each rule (1st row) and number of rules inspected in each entity (2nd row), after the specified number of simulation steps with the cyclical selection inspection strategy

The displayed results are in line with the expectations.

5.1.3.4. Verification test: Stochastic universal sampling strategy

The verification test aims to determine whether the Stochastic universal sampling strategy is properly implemented in the computer simulation. By applying the Stochastic universal sampling (SUS) strategy, inspection agency selects the rules for inspection, with the probability of inspecting a specific rule being proportionate to the agency's assumptions about the costs of compliance with the relevant rule.

The verification test was conducted in an environment with 100 entities that must comply with 8 rules. The compliance costs (resource requirements) are explicitly defined[21]. The inspection agency applies the SUS inspection strategy and its inspection capacity is 0.20. Other simulation parameters are not relevant for this test and are set arbitrarily. With the increasing number of simulation steps, the distribution of the number of inspections of individual rules should – in accordance with the law of large numbers – converge towards the distribution of compliance costs. After setting up the simulation, 50 steps were performed. The test was ran with the input parameters contained in the table below (Table 5.12).

Table 5.12 – Input parameters: Stochastic universal sampling strategy implementation

Input parameter:	Values:
type-of-inspection-selection	Stochastic universal sampling
number-of-entities	100
number-of-rules	8
resource-requirements-type	Input from line
resource-requirements	[2 10 5 7 1 6 12 3]
inspectors-capacity	0.2

Figure 5.12 displays the test results.

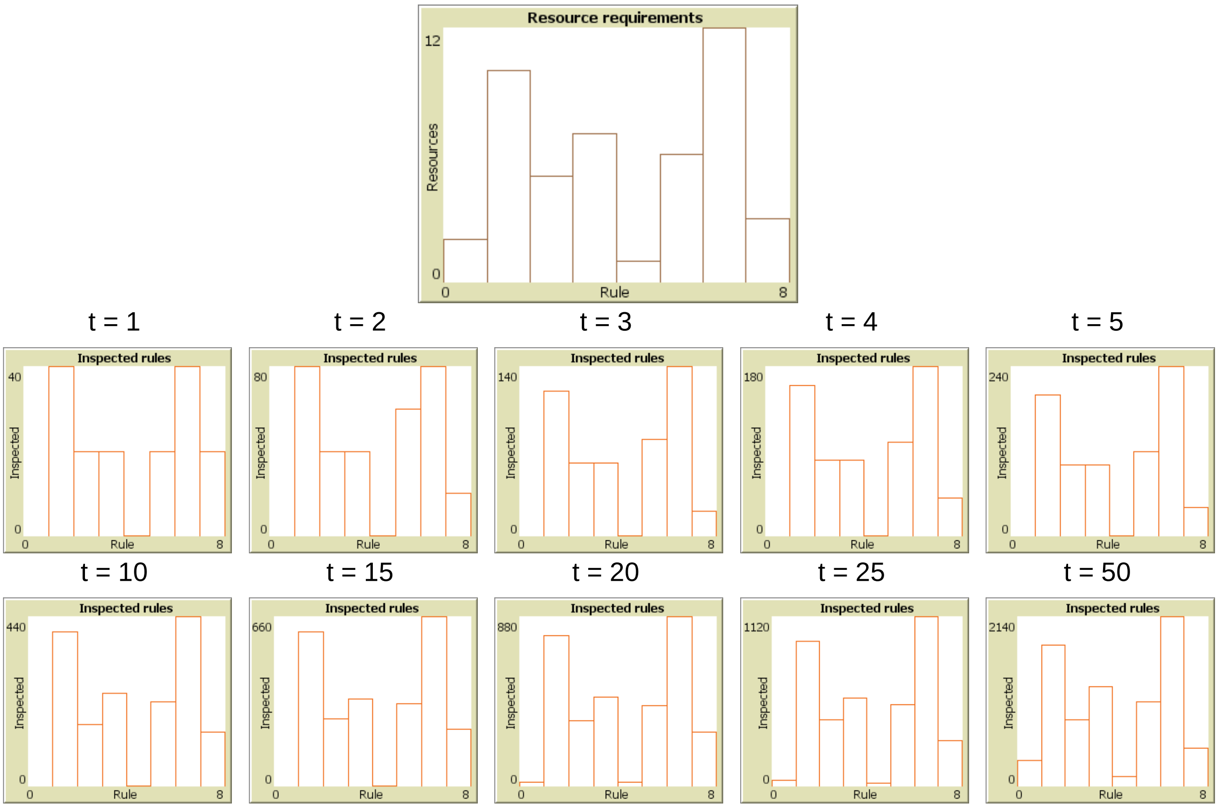

Figure 5.12 – Model verification: comparison of the compliance costs (resource requirements – 1st row) and the number of inspections of each individual rule, after the stated simulation step (2nd and 3rd row), when applying the Stochastic universal sampling inspection strategy

[21] The vector values were chosen arbitrarily, keeping in mind the desired visualization of the simulation results. The specific amounts are not significant for the verification results.

The graphs show that the frequency of inspections of individual rules converges, over time, to the distribution of compliance costs of those rules, when applying the SUS inspection strategy. The results of the simulation are in line with the expectations.

5.1.3.5. Verification test: Accuracy of inspections

The verification test aims to determine whether the accuracy of the inspections is properly implemented in the computer simulation. Actual inspections are not always accurate, i.e. they can misjudge compliance and conclude that the entity is in compliance with a certain provision, even though in reality it violates it. The model therefore includes a parameter that sets the expected inspection accuracy.

The test of implementation of accuracy of the inspections was conducted in an environment with 100 entities that must comply with 5 rules. The compliance costs are explicitly defined[22]. Entities are risk-takers, and the risk appetite is the same for all the entities. The increased risk appetite was set in order to get more violations in the simulation. Entities do not discount history, and learning is consistent with the fictitious play. The penalty[23] for a violation is 100. The inspection agency uses the random entity selection strategy, and its inspection capacity is 0.20. In order to verify the correctness of the inspection accuracy setting, the simulation was run 5 times – for 5 different inspection accuracy values. Other simulation parameters are not relevant for this test and are set arbitrarily. After setting up the simulation, 25 simulation steps were performed. A sufficiently large number of inspections should be performed so that at a given simulation step and for given values of the input parameters, the results of the inspections – in accordance with the law of large numbers – converge to the set accuracy of inspections.

Verification was performed by comparing the ratio of the number of violations that the inspections accurately identified and the total number of violations that were inspected, for the given accuracy of inspections.

The test was ran with the input parameters contained in the table below (Table 5.13). The values of other input parameters do not affect the test results. After setting up the simulation, 25 steps were completed, after which the total number of violations that were subjected to inspections and the number of violations that the inspections accurately identified were recorded.

Table 5.13 – Input parameters: inspection accuracy implementation

Input parameter:	Values:
type-of-inspection-selection	Random entity
number-of-entities	100
number-of-rules	5
resource-requirements-type	Input from line
resource-requirements	[3 5 7 9 11]
default-risk-attitude	2
max-risk-attitude-deviation	0%
k-hyperbolic-discounting	0
learning-mechanism	Fictitious play
punishment-size	100
inspectors-capacity	0.2
inspection-accuracy	{100%, 90%, 80%, 70%, 60%}

Table 5.14 contains a comparison of the data generated in the simulation and the expected values.

22 The vector values were chosen arbitrarily, keeping in mind the desired visualization of the simulation results. The specific amounts are not significant for the verification results.

23 The value was chosen with the desired visualization of the simulation results in mind.

Table 5.14 – Model verification: comparison of set and achieved inspection accuracy; in the column "Difference" is the absolute difference (percentage points) of the given and achieved accuracy, and the column "95% CI" states the 95% confidence interval, based on the calculated standard error.

Set accuracy	Sample	Detected violations	Achieved accuracy	Difference	95% CI
100 %	695	695	100.0 %	0.0%	±0.0%
90 %	645	717	90.0 %	0.0%	±2.2%
80 %	571	719	79.4 %	0.6%	±2.9%
70 %	505	723	69.8 %	0.2%	±3.3%
60 %	432	721	59.9 %	0.1%	±3.6%

The presented results show that the differences between expected and achieved values are relatively small, and all differences are very well within the acceptable 95% confidence range. Hence, the simulation results are in line with the expectations.

5.2. Model validation

5.2.1. General validation

The general validity of ICARUS was assessed primarily by analysing graphical representations of the simulation results (for example, the impact of increasing the penalty on increasing/decreasing the total number of violations in the system). The considered empirical macro-structures and micro-structures are based on the secondary data that is described in more detail in the earlier chapters.

The developed model contains significant stochasticity. Namely, the values of many internal model variables are partially or completely stochastic, to enable modelling of agent heterogeneity, uncertainty in model assumptions and specific combinations of input parameters, and for modelling agents' bounded rationality. This practice is common in agent-based modelling [11][91]. Due to this stochasticity, two runs of the simulation of the developed model, started with the same input parameters, are unlikely to have the same results i.e. the simulations will not end with the same output values (unless they are started with the same input parameters and the same initial seed). Therefore, results of several simulations repeated with the same input parameters were analysed as part of the general validation, with the stochastic elements of the simulation being randomly generated in each iteration[24]. In most tests, simulations were repeated 100 times with the same input parameters, in line with a general recommendation by Nikolić and co-authors [57, p. 111]. In two general validation tests, simulations were repeated 1,000 times, in order to generate a larger number of rare events.

Most validation tests were conducted by analysing the data recorded after 25 simulation steps. I.e., although many analyses of analytical models (for example, application of game theory) [92][93], and agent models [18] rely on the analysis of equilibrium – achieving which would require a large number of simulation steps – empirical data relevant to this paper refer to shorter time periods (e.g., 3 to 15 years). In addition, it is necessary to take into account that inspection strategies must be able to show concrete results within a reasonable time, in order to be politically feasible. The exception to the 25 steps limit is the test described in the last sub-chapter in which 30 simulation steps were performed, in order to increase the number of relevant data points.

The general validity was assessed primarily by comparing the expected patterns and values with the results of a large number of simulation runs of the ICARUS model conducted for a given range of values of input parameters. The ranges of parameter values were determined in line with the available empirical data or – if such data is not available or applicable to the developed model – based on estimation of range that would be relevant for a particular test.

[24] Random values in the NetLogo simulation environment are actually pseudo-random, since seed and all seed derivatives are generated through a deterministic process [67].

The validation tests utilize random entity selection and cyclical strategies. These strategies are comparable to the real-world inspection strategies and therefore can utilize and/or be compared with the available empirical data. The results of simulations that implement random selection and stochastic universal sampling strategies are generally not directly comparable with the available empirical data. However, for the sake of completeness, the results of the validation tests include all applicable inspection strategies. The seed for generation of stochastic elements is not predefined and is randomly set for each simulation.

The following table (Table 5.15) contains parameters that are equal (unchanged) in all general validation assessments (tests).

Table 5.15 – Input parameters that are unchanged in all general validation tests

Input parameter:	Values:
learning-mechanism	Fictitious play
risk-exp	OFF
stackelberg-aware	OFF

5.2.1.1. General validation: Increase in penalty leads to a decrease in violations

Empirical research shows that increase in penalty has the effect of increasing compliance with regulations, i.e. reducing the number of violations, which is explained in more detail in chapter 3.2 and specifically in [30]. This validation test aims to determine whether that connection is manifested in the computer simulation of the model.

Since the empirical data underlying this test does not limit the parametric space of the model, the input parameters of the computer simulation were set taking into account: the need to show the heterogeneity of agents; the need to avoid setting overly idealized conditions; and the need for appropriate visualization of the simulation results.

The validation test includes 100 entities that must comply with 6 rules. The compliance costs are explicitly defined. In order to model the heterogeneity of compliance costs between entities, each entity's costs vary (up to 25%) from other entities and diverge (also, up to 25%) from the inspection agency's perception of the compliance costs. Entities also differ in their risk appetites. That is, the entities are on average risk-neutral, but there are differences in their risk appetite of ±50%. Thus, risk appetite varies from 0.5 (risk-averse entities) to 1.5 (risk-taking entities). Entities do not have a perfect memory (recall) of the past, i.e. historical data are discounted along $\kappa = 0.5$. Entities learn according to the fictitious play model. The capacity of the inspection agency is 0.25. Inspections in 90% of cases accurately identify non-compliances. When cyclical strategy is applied, 3 rules are inspected in each cycle. That is, after the inspection of the same 3 (randomly selected) rules in all entities, the inspection of the remaining 3 rules is carried out.

The validation test was performed by recording the results of computer simulations in which the prescribed penalty was changed in the range from 30 to 80 with increments of 1. The validation test included all 4 inspection strategies and each computer simulation was repeated 100 times, with the same input parameters.

The test was performed with the input parameters contained in the table below (Table 5.16).

Table 5.16 – Input parameters: increase in penalty leads to a decrease in violations

Input parameter:	Values:
type-of-inspection-selection	{Random, Random entity, Cycle, Stochastic Universal Sampling}
number-of-entities	100
number-of-rules	6
resource-requirements-type	Input from line
resource-requirements	[1 3 5 7 9 11]
max-deviation-resources	0.25
default-risk-attitude	1
max-risk-attitude-deviation	50%

k-hyperbolic-discounting	0.5
punishment-size	[30, 1, 80][25]
inspectors-capacity	0.25
inspection-accuracy	90%
rules-inspected-in-one-cycle	3

Figure 5.13 displays the validation results. The graph displays the standard deviation, not the standard error, as to show the variability of the simulation data and not the accuracy of the calculated arithmetic mean. It should be emphasized that the standard error (with 95% CI) is very small and would not be visible even if it was displayed. The same principle applies to all the figures of this type.

The results show a decrease in the total number of violations in the system with increasing penalties, regardless of the chosen inspection strategy. That is, test results are in line with the expectations.

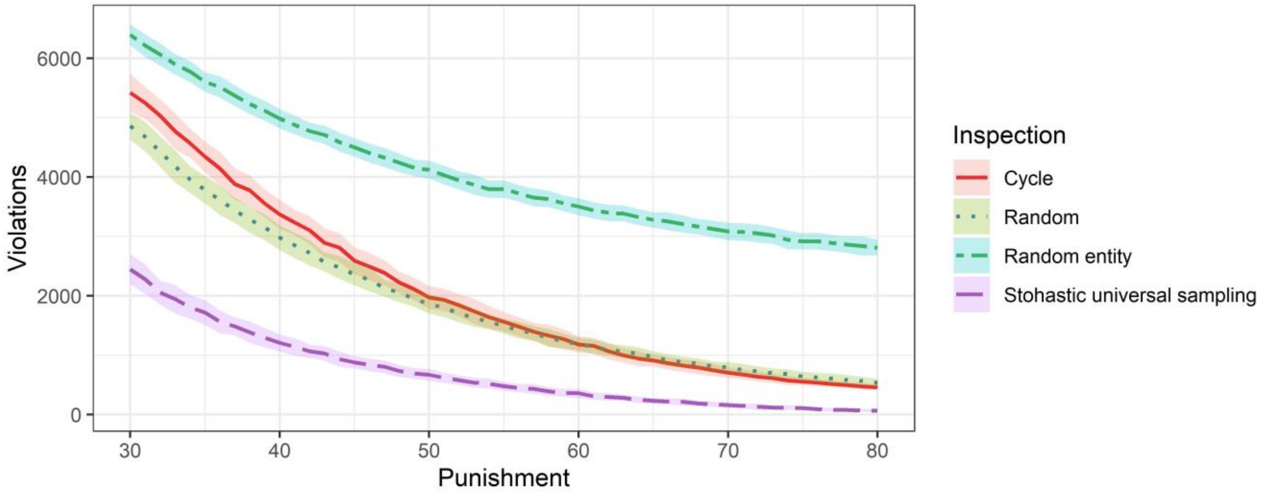

Figure 5.13 – General validation: the average total (cumulative) number of violations in the simulation (y-axis) after 25 steps for a given penalty value (x-axis), for the given inspection strategy; the coloured line represents the arithmetic mean ($\bar{x}$) of the results of 100 repetitions of the simulation with the same input parameters, and the shaded area around the lines is the range $\pm 1\sigma$

5.2.1.2. General validation: Increase in inspections leads to a reduction in violations

Empirical research shows that conducting inspections has the effect of increasing compliance with regulations, i.e. of reducing the number of violations of the provisions, which is explained in more detail in Chapter 3.2. Validation aims to determine whether this relationship is manifested in the computer simulation of the model.

The validation test was performed taking into account the same assumptions and with the same input parameters as the previous test (an increase in penalty leads to a reduction in violations). The only differences in the input parameters were caused by the different test objective. The penalty level is fixed to 40, and the change in the number of inspections was modelled via a change in the capacity of the inspection agency from 0.1 to 0.5 with increments of 0.01.

The test was performed with the input parameters contained in the table below (Table 5.17).

[25] Parameter values in square brackets separated by commas define a series of uniformly increasing values. The first value indicates the beginning of the interval, the second value the increment step, and the last value represents the end of the interval. In this case, it is the series: 30, 31, 32,..., 79, 80.

Table 5.17 – Input parameters: increase in inspections leads to a reduction in violations

Input parameter:	Values:
`type-of-inspection-selection`	`{Random, Random entity, Cycle, Stochastic Universal Sampling}`
`number-of-entities`	`100`
`number-of-rules`	`6`
`resource-requirements-type`	`Input from line`
`resource-requirements`	`[1 3 5 7 9 11]`
`max-deviation-resources`	`0.25`
`default-risk-attitude`	`1`
`max-risk-attitude-deviation`	`50%`
`k-hyperbolic-discounting`	`0.5`
`punishment-size`	`40`
`inspectors-capacity`	`[0.1, 0.01, 0.5]`
`inspection-accuracy`	`90%`
`rules-inspected-in-one-cycle`	`3`

Figure 5.14 displays the validation results. The graph displays a decrease in the total number of violations in the system when an increase in the capacity of the inspection agency occurs, i.e. with an increase in the number of conducted inspections, regardless of the chosen inspection strategy. Therefore, it can be concluded that the test results are in line with expectations.

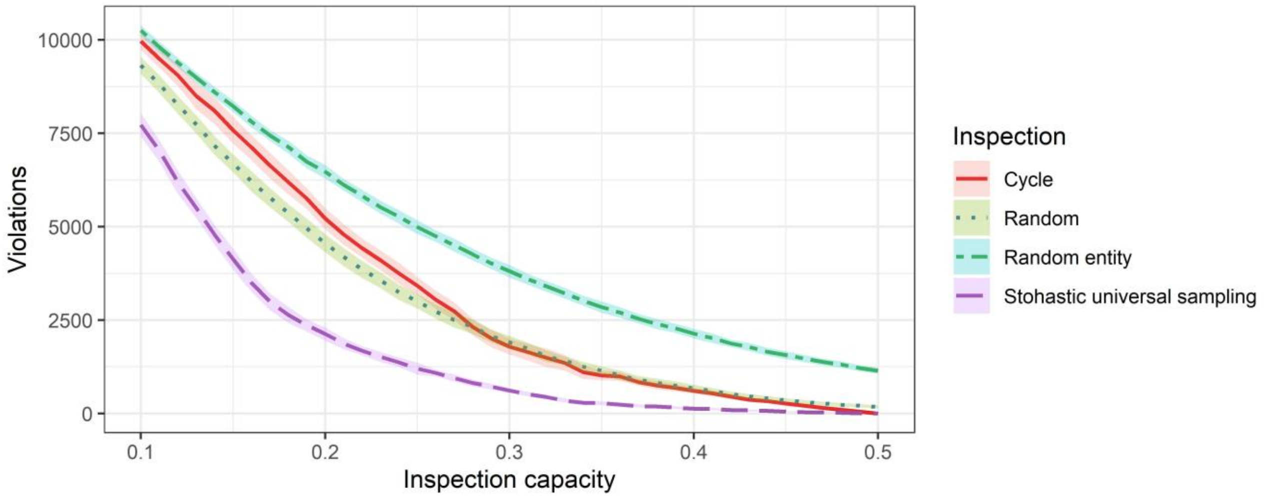

Figure 5.14 – General validation: the average total (cumulative) number of violations in the simulation (y-axis) after 25 simulation steps, for a given capacity of the inspection agency (x-axis) and the given inspection strategy, with the penalty of 40; the coloured line represents the arithmetic mean ($\bar{x}$) of the results of 100 repetitions of the simulation with the same input parameters, and the shaded area around the lines is the range $\pm 1\sigma$

5.2.1.3. <u>General validation: A low penalty has a marginal impact on the level of violations</u>

Empirical data shows that conducting inspections has a very low impact on compliance with regulations, i.e. on reducing the number of violations in situations where penalties for non-compliance are very low [45]. This validation test aims to determine whether this regularity is manifested in the computer simulation of the model.

The validation test was performed taking into account the same assumptions and with the same input parameters as the test in the previous chapter. However, in order to model an environment with very low punishment, the value of the penalty was reduced to 3. The test results were collected in the same way as in the previous chapters.

The test was performed with the input parameters contained in the table below (Table 5.18).

Table 5.18 – Input parameters: low penalty has a marginal impact on the level of violations

Input parameter:	Values:
type-of-inspection-selection	{Random, Random entity, Cycle, Stochastic Universal Sampling}
number-of-entities	100
number-of-rules	6
resource-requirements-type	Input from line
resource-requirements	[1 3 5 7 9 11]
max-deviation-resources	0.25
default-risk-attitude	1
max-risk-attitude-deviation	50%
k-hyperbolic-discounting	0.5
punishment-size	3
inspectors-capacity	[0.1, 0.01, 0.5]
inspection-accuracy	90%
rules-inspected-in-one-cycle	3

The results are presented graphically (Figure 5.15). The results show a decrease in the total number of violations in the system with an increase in the capacity of the inspection agency, i.e. with an increase in the number of inspections, regardless of the chosen inspection strategy, but this decrease is much smaller than in the previous test (Figure 5.14).

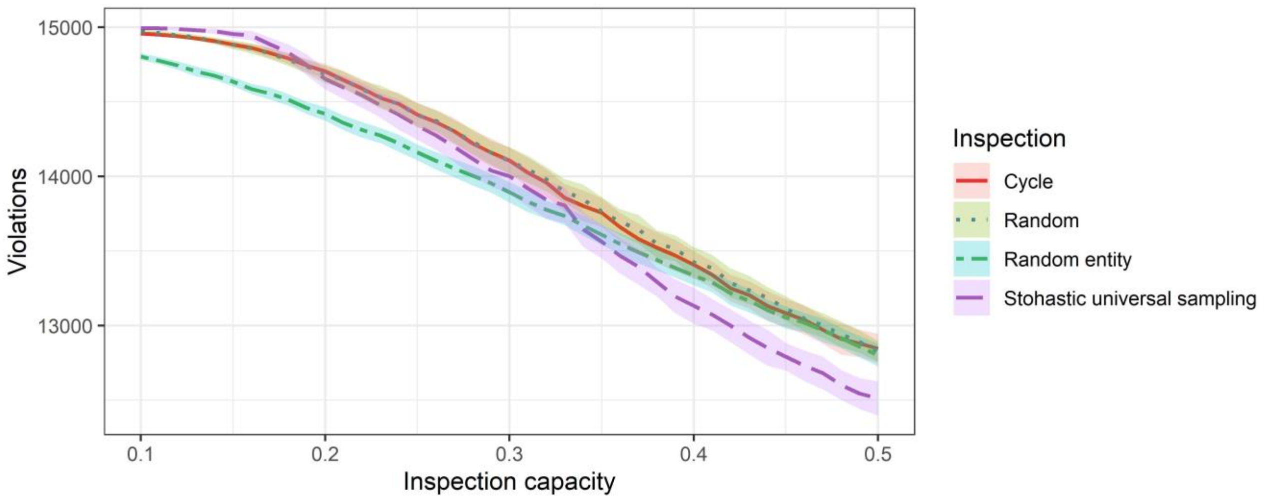

Figure 5.15 – General validation: the average total (cumulative) number of violations in the simulation (y-axis) after 25 simulation steps, for a given capacity of the inspection agency (x-axis) and the given inspection strategy, with the penalty of 3; the coloured line represents the arithmetic mean ($\bar{x}$) of the results of 100 repetitions of the simulation with the same input parameters, and the shaded area around the lines is the range $\pm 1\sigma$

That is, by reducing the penalty to 3, all inspections achieved a much smaller reduction in the total number of violations when compared to the test with penalty = 40, irrespective of the applied inspection strategy. Table 5.19 displays a comparison of the effect of changes in inspection capacity for different levels of penalties, while applying the random entity selection strategy.

Table 5.19 – General validation: comparison of the average total number of violations and the percentage reduction after 25 steps of the simulation for a given capacity of the inspection agency and the amount of the penalty, while applying the random entity selection strategy; the absolute amounts shown are the arithmetic means ($\bar{x}$) of the results of 100 simulation iterations with the same input parameters

Punishment	Inspection capacity		Violations reduction
	0.1	0.5	
40	10,254.98	1,147.36	88.81%
3	14,803.03	12,801.85	13.52%

The presented data show that a fivefold increase in inspection capacity led to an almost tenfold decrease in the number of violations at the punishment level of 40. On the other hand, a fivefold increase in inspection capacity at the punishment level of 3 led to a decrease in the number of offenses of less than 15%. From these data, it can be concluded that inspections have a very low impact on the overall compliance level or on reduction in the number of violations in situations where the penalty for non-compliance is very low. Hence, it can be concluded that the test results are in line with the expectations.

5.2.1.4. General validation: Higher compliance costs lead to more violations

Empirical research shows that higher regulatory compliance costs lead to more violations, which is described in more detail in Chapter 3.2, and specifically in [41] and [37]. The validation test aims to determine whether this regularity is manifested in the computer simulation of the model.

The validity was tested taking into account the same assumptions and with the same input parameters as the first general validity test. However, some changes in the input parameters were made because of the goal of the test. Thus, the penalty level is fixed at 40 and the initial compliance costs are explicitly defined as in the previous scenarios. However, a change in the compliance costs can be made in different ways, as there are several rules. Therefore, for the purposes of performing this validity test, 3 scenarios were constructed. In each scenario, the costs of complying with 6 rules are defined according to a specific algorithm, for 20 different configurations of compliance costs. The scenarios are described below.

1. *Scenario: Linear increase in costs of compliance with all the rules*

In this scenario, the costs of complying with every rule increase incrementally by 0.25 in each parameter configuration. Accordingly, the arithmetic mean of the costs of compliance with all the rules increases incrementally by 0.25 from configuration to configuration.

2. *Scenario: Linear increase in costs of compliance with some rules*

In this scenario, the costs of complying with the rules 2, 4 and 6 increase by 0.5 in each configuration, and the costs of complying with the rules 1, 3 and 5 remain unchanged. The arithmetic mean of the costs of compliance with all the rules is therefore increased by 0.25 from configuration to configuration.

3. *Scenario: Linear increase and decrease of costs of compliance*

In the 3rd (and most complex) scenario, the costs of complying with the rules 2, 3 and 4 increase by 0.75 in each configuration, the cost of complying with the rule 1 remains unchanged. The cost of complying with the rule 5 decreases by 0.25 in each configuration, and the cost of complying with the rule 6 is reduced by 0.5 in each configuration. The arithmetic mean of the costs of compliance with all the rules is therefore increased by 0.25 from configuration to configuration.

Accordingly, in all the scenarios, the arithmetic mean of the costs of compliance with all the rules increases incrementally by 0.25 in each configuration, but the costs of complying with different rules change in different ways.

Analogously to the previous validation tests, the test was performed by recording results of computer simulations in which the compliance costs parameter changed according to the described principles. The test was performed with the input parameters contained in the table below (Table 5.20).

Table 5.20 – Input parameters: higher compliance costs lead to more violations

Input parameter:	Values:
type-of-inspection-selection	{Random, Random entity, Cycle, Stochastic Universal Sampling}
number-of-entities	100
number-of-rules	6
resource-requirements-type	Input from line
max-deviation-resources	0.25
default-risk-attitude	1
max-risk-attitude-deviation	50%
k-hyperbolic-discounting	0.5
punishment-size	40
inspectors-capacity	0.25
inspection-accuracy	90%
rules-inspected-in-one-cycle	3

The vectors of compliance costs, i.e. the values of the variable `resource-requirements` in all the scenarios and all configurations are shown below (Table 5.21).

Table 5.21 – 20 possible values (configurations) of the compliance costs vector, for 3 considered scenarios of changes in compliance costs

Compliance costs configuration	Compliance costs scenarios		
	1. Scenario	2. Scenario	3. Scenario
A	[1 3 5 7 9 11]	[1 3 5 7 9 11]	[1 3 5 7 9 11]
B	[1.25 3.25 5.25 7.25 9.25 11.25]	[1 3.5 5 7.5 9 11.5]	[1 3.75 5.75 7.75 8.75 10.5]
C	[1.5 3.5 5.5 7.5 9.5 11.5]	[1 4 5 8 9 12]	[1 4.5 6.5 8.5 8.5 10]
D	[1.75 3.75 5.75 7.75 9.75 11.75]	[1 4.5 5 8.5 9 12.5]	[1 5.25 7.25 9.25 8.25 9.5]
E	[2 4 6 8 10 12]	[1 5 5 9 9 13]	[1 6 8 10 8 9]
F	[2.25 4.25 6.25 8.25 10.25 12.25]	[1 5.5 5 9.5 9 13.5]	[1 6.75 8.75 10.75 7.75 8.5]
G	[2.5 4.5 6.5 8.5 10.5 12.5]	[1 6 5 10 9 14]	[1 7.5 9.5 11.5 7.5 8]
H	[2.75 4.75 6.75 8.75 10.75 12.75]	[1 6.5 5 10.5 9 14.5]	[1 8.25 10.25 12.25 7.25 7.5]
I	[3 5 7 9 11 13]	[1 7 5 11 9 15]	[1 9 11 13 7 7]
J	[3.25 5.25 7.25 9.25 11.25 13.25]	[1 7.5 5 11.5 9 15.5]	[1 9.75 11.75 13.75 6.75 6.5]
K	[3.5 5.5 7.5 9.5 11.5 13.5]	[1 8 5 12 9 16]	[1 10.5 12.5 14.5 6.5 6]
L	[3.75 5.75 7.75 9.75 11.75 13.75]	[1 8.5 5 12.5 9 16.5]	[1 11.25 13.25 15.25 6.25 5.5]
M	[4 6 8 10 12 14]	[1 9 5 13 9 17]	[1 12 14 16 6 5]
N	[4.25 6.25 8.25 10.25 12.25 14.25]	[1 9.5 5 13.5 9 17.5]	[1 12.75 14.75 16.75 5.75 4.5]
O	[4.5 6.5 8.5 10.5 12.5 14.5]	[1 10 5 14 9 18]	[1 13.5 15.5 17.5 5.5 4]
P	[4.75 6.75 8.75 10.75 12.75 14.75]	[1 10.5 5 14.5 9 18.5]	[1 14.25 16.25 18.25 5.25 3.5]
Q	[5 7 9 11 13 15]	[1 11 5 15 9 19]	[1 15 17 19 5 3]
R	[5.25 7.25 9.25 11.25 13.25 15.25]	[1 11.5 5 15.5 9 19.5]	[1 15.75 17.75 19.75 4.75 2.5]
S	[5.5 7.5 9.5 11.5 13.5 15.5]	[1 12 5 16 9 20]	[1 16.5 18.5 20.5 4.5 2]
T	[5.75 7.75 9.75 11.75 13.75 15.75]	[1 12.5 5 16.5 9 20.5]	[1 17.25 19.25 21.25 4.25 1.5]

The test results are presented graphically, for 3 considered scenarios (Figure 5.16, Figure 5.17 and Figure 5.18). Unlike the previous tests, the results are – due to the characteristics of the data on the x-axis – presented using custom scatter plots. As in the previous tests, the standard error (with 95% CI) is very small.

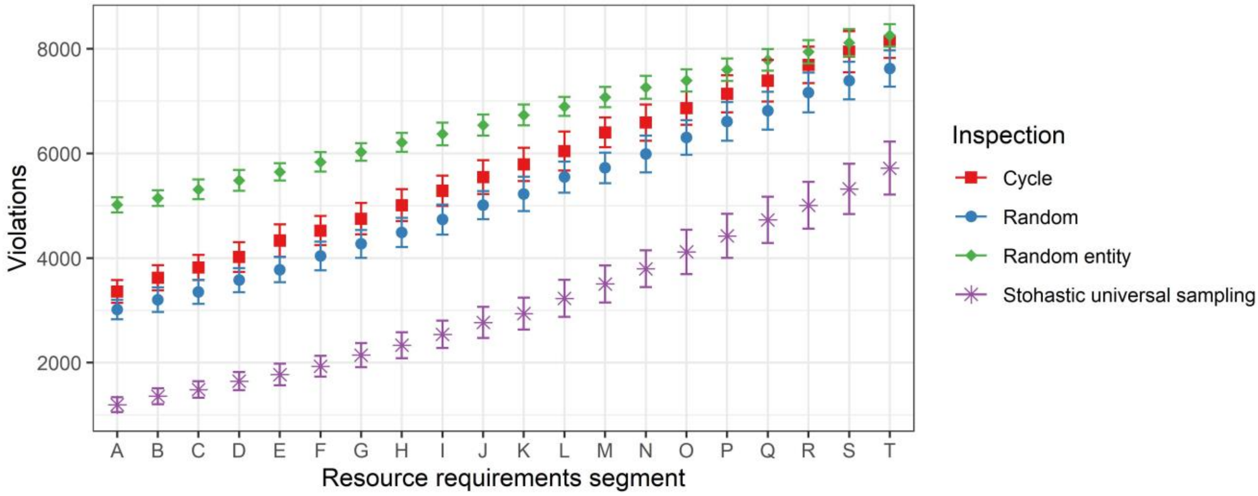

Figure 5.16 – General validation: the average total (cumulative) number of violations after 25 simulation steps, for the given inspection strategy, with the compliance costs vector changing according to the scenario 1 (linear increase of all compliance costs); the centre of the data points represents the arithmetic mean ($\bar{x}$) of 100 simulation results repeated with the same parameters, and the upper and lower limits represent the range $\pm 1\sigma$

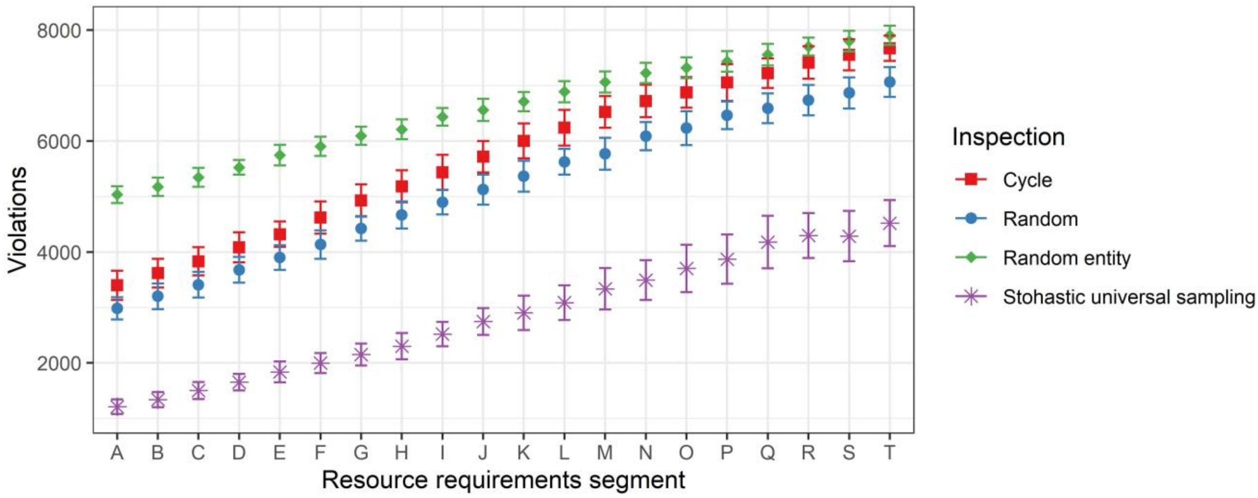

Figure 5.17 – General validation: the average total (cumulative) number of violations after 25 simulation steps, for a given inspection strategy, with the compliance costs vector changing according to the scenario 2 (linear increase of a part of compliance costs); the centre of the data points represents the arithmetic mean ($\bar{x}$) of 100 simulation results repeated with the same parameters, and the upper and lower limits represent the range $\pm 1\sigma$

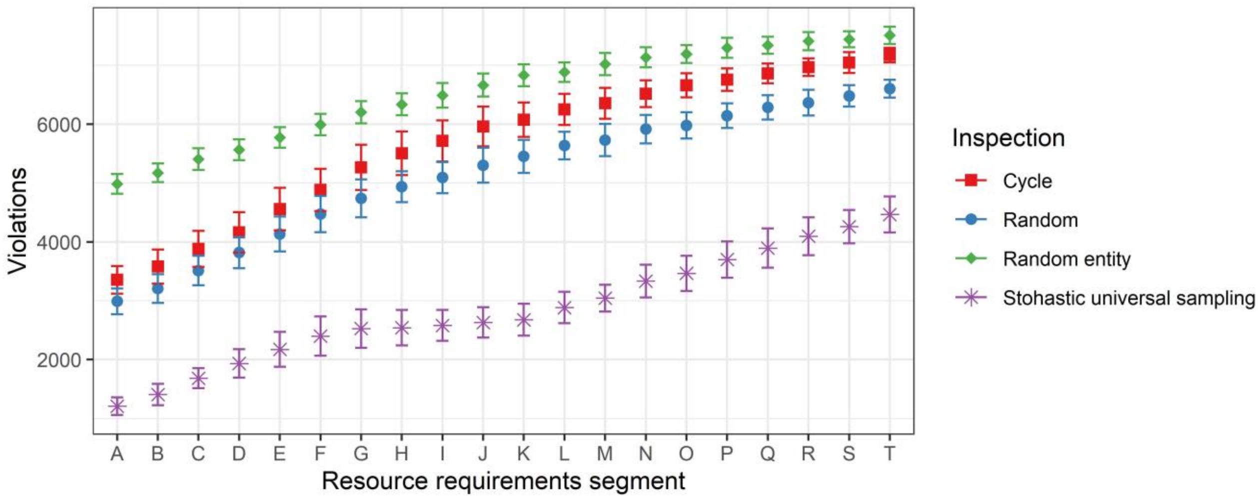

Figure 5.18 – General validation: the average total (cumulative) number of violations after 25 simulation steps, for a given inspection strategy, with the compliance costs vector changing according to the scenario 3 (linear increase and decrease of compliance costs); the centre of the data points represents the arithmetic mean ($\bar{x}$) of 100 simulation results repeated with the same parameters, and the upper and lower limits represent the range $\pm 1\sigma$

The presented results show that higher (on average) compliance costs lead to a higher number of violations, regardless of the chosen inspection strategy. Therefore, it can be concluded that the results are in line with the expectations.

5.2.1.5. General validation: Increase in the perceived probability of inspection leads to a decrease in violations

The empirical research shows that a higher perceived probability of an inspection, i.e. the belief that an inspection is more likely, leads to a reduction in the total number of violations, which is explained in more details in Chapter 3.2 and specifically in [45]. The validation test aims to determine whether this regularity is manifested in the computer simulation of the model.

The ICARUS model does not contain input parameters that could directly affect entity's perception of the likelihood of inspections. However, an entity's propensity to take risks (risk appetite) can be used as a mechanism that indirectly influences the entity's belief in the likelihood of an inspection. Since, the perceived probability of inspection is inversely proportional to the risk appetite. The relationship between these parameters is explained in more detail in Chapter 4.2.2.

This test was performed with the same input parameters as the first general validation test (Chapter 5.2.1.1), except for the following differences: entity's underlying risk appetite varies from 0.1 to 3, with increments of 0.1, and the penalty is set at 40. The risk appetite of an individual entity may differ from the default risk appetite by $\pm 50\%$.

The test was performed with the input parameters contained in the table below (Table 5.22).

Table 5.22 – Input parameters: increase in the perceived probability of inspection leads to a reduction in violations

Input parameter:	Values:
`type-of-inspection-selection`	`{Random, Random entity, Cycle, Stochastic Universal Sampling}`
`number-of-entities`	`100`
`number-of-rules`	`6`
`resource-requirements-type`	`Input from line`
`resource-requirements`	`[1 3 5 7 9 11]`
`max-deviation-resources`	`0.25`

default-risk-attitude	[0.1, 0.1, 3]
max-risk-attitude-deviation	50%
k-hyperbolic-discounting	0.5
punishment-size	40
inspectors-capacity	0.25
inspection-accuracy	90%
rules-inspected-in-one-cycle	3

Figure 5.19 displays the results of the validation test. The results show an increase in the total number of violations in the system with an increase in the underlying risk appetite. Hence, the results are in line with the expectations.

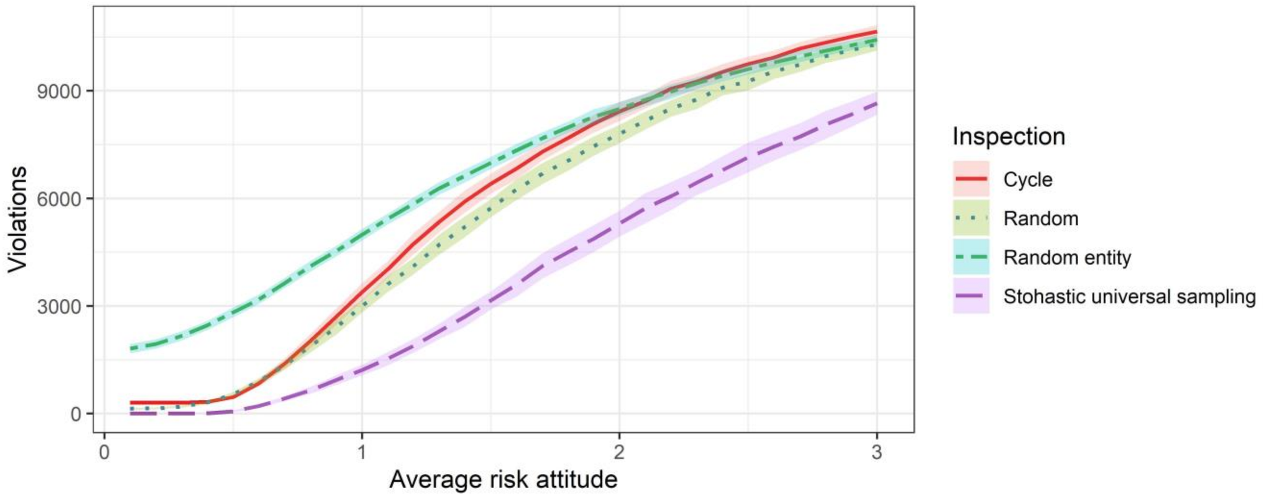

Figure 5.19 – General validation: the average total (cumulative) number of violations in the simulation (y-axis) after 25 steps with a given average risk attitude, for the given inspection strategy; the colored line represents the arithmetic mean ($\bar{x}$) of the results of 100 repetitions of the simulation with the same input parameters, and the shaded area around the lines is the range $\pm 1\sigma$

5.2.1.6. <u>General validation: There are fewer violations after an inspection</u>

Empirical research shows that inspected entities are less likely to break the rules after an inspection, compared to other (similar) non-inspected entities, as explained in more detail in Chapter 3.2 and, in particular, in [45]. This validation test aims to determine whether this regularity is manifested in the computer simulation of the model.

Unlike the previously described validation tests that examined system-level results (macro-level), this test assesses the regularities in the behaviour of individual entities (micro-level). Therefore, the design of the test has been partially adjusted.

This test was performed with the same input parameters as the first general validation test, but with the following differences: instead of 100 iterations of the computer simulation with the same input parameters, 1,000 iterations were performed, the inspection capacity was reduced to 0.2 and the penalty was set at 40. 25 simulation steps were performed.

Validity was assessed on the data that was collected in the following way: in each step of the simulation, data of one entity was recorded (random selection of one entity out of 100, in each step of the simulation). The following data was collected for that entity: inspection strategy, current step in the simulation (1 to 25), indicator of whether that entity was inspected in the previous simulation step, and the number of violations of rules in the current step (0 to 6). The number of recorded violations does not depend on the inspections of entities at the time (simulation step) of data collection (a random sample of all entities, and not just inspected entities). Accordingly, 25,000 data points were collected for each inspection strategy.

The number of simulation repetitions has been increased and the capacity of inspectors has been reduced (as compared to the previous validation tests) in order to increase the number of data points with a random inspection strategy and with an indicator that no inspection of the entity in question was performed in the previous step. Namely, due to the characteristics of the random inspection strategy, such data points are relatively rare. The validity of the stochastic universal sampling strategy cannot be verified by this test, given that in the configuration with 6 rules and inspection capacity of 0.2 each entity will be inspected at each step, i.e. there will be no entities that were not inspected in the previous step.

The test was performed with the input parameters contained in the table below (Table 5.23).

Table 5.23 – Input parameters: there are fewer violations after an inspection

Input parameter:	Values:
type-of-inspection-selection	{Random, Random entity, Cycle}
number-of-entities	100
number-of-rules	6
resource-requirements-type	Input from line
resource-requirements	[1 3 5 7 9 11]
max-deviation-resources	0.25
default-risk-attitude	1
max-risk-attitude-deviation	50%
k-hyperbolic-discounting	0.5
punishment-size	40
inspectors-capacity	0.2
inspection-accuracy	90%
rules-inspected-in-one-cycle	3

Figure 5.20 displays the test results. The results are presented using boxplot (BP) diagrams since the presented data are not normally distributed, which is especially pronounced for the random entity selection strategy. The presented data confirms that the entities that were not subject to inspection in the previous step (group A) violated rules more than the entities that were subject to inspection in the previous step (group B). This regularity is visible in all the steps of the simulation. In addition, this regularity is particularly pronounced when random entity selection or cyclic strategy are applied. As already stated, these 2 inspection strategies are comparable to actual (real-world) inspection strategies and therefore particularly relevant for the validation. Hence, it can be concluded that the test results are in line with the expectations.

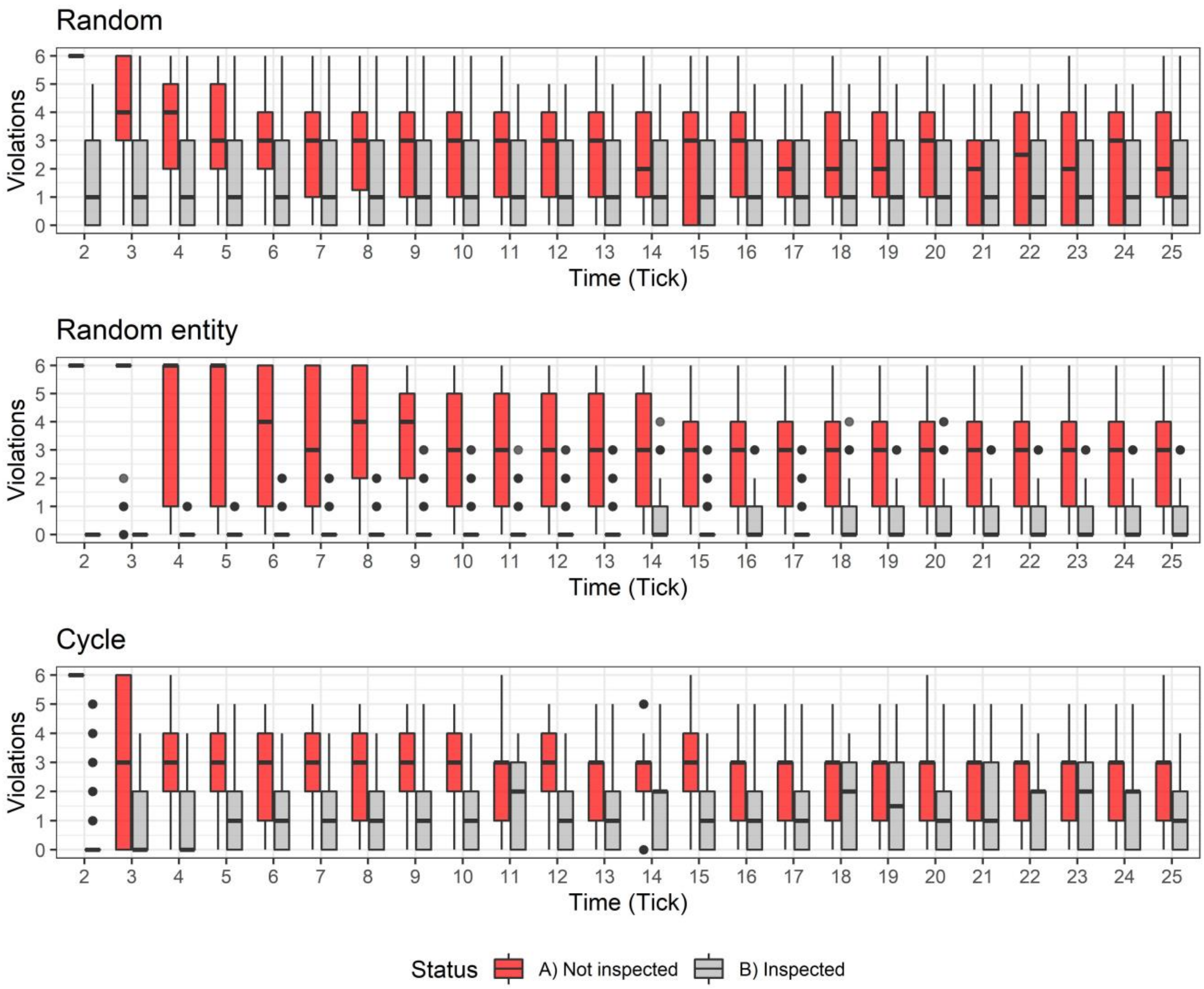

Figure 5.20 – General validation: comparison of the number of violations by individual entities that were not inspected in the previous simulation step (group A – red column) and the entities that were inspected in the previous simulation step (group B – gray column); each bar presents a statistical summary of the violations in a given step of the simulation, steps 2 to 25 (data for step 1 of the simulation is not shown because there was no inspection history at that time); each graph shows the data for the specified inspection strategy

5.2.1.7. General validation: There are fewer violations after the penalty

Empirical research shows that entities, after being punished for non-compliance, violate the rules less than other comparable entities, which is explained in more detail in Chapter 3.2 and, in particular, in [30]. This validation test therefore seeks to determine whether this regularity is also manifested in the computer simulation of the model.

Like the validation test in the previous chapter, this test assesses the regularities in the behaviour of individual entities (micro-levels) and was conducted with the same input parameters.

The validity was assessed on the data that was recorded as follows: in each step of the simulation, data on one entity was collected (random selection of one entity out of 100). The following data was recorded for that entity: inspection strategy, entity's risk appetite, an indicator whether in the previous step of the simulation, the entity was penalized for non-compliance, and the number of violations in the current step (from 0 to 6). Analogously to the previous test, the number of violations that was recorded is not related to whether the entity was inspected at the time of the data collection.

The simulation was repeated 1,000 times with the same parameters and the inspection capacity was set to 0.2, to increase the number of entities in the sample that were penalized for non-compliance in the previous step, especially for the application of the random entity selection inspection strategy and for entities with a low risk appetite. Due to low risk appetite and low inspection capacity when applying the random entity selection strategy, such data points are relatively rare. Therefore, 25,000 data points were collected for each inspection strategy.

The observed entities' risk appetites ranges are grouped into 5 categories. Risk grouping was performed because the number of violations and the expectation that not all entities will violate equally have to be taken into account. That is, the entities with a higher risk appetite are expected to break the rules more than entities with a lower risk appetite, assuming that deviations in risk appetite are evenly distributed, as confirmed by the verification test described in Chapter 5.1.2.2.

The test was ran with the same input parameters as the previous test (Table 5.23 contains the input parameters).

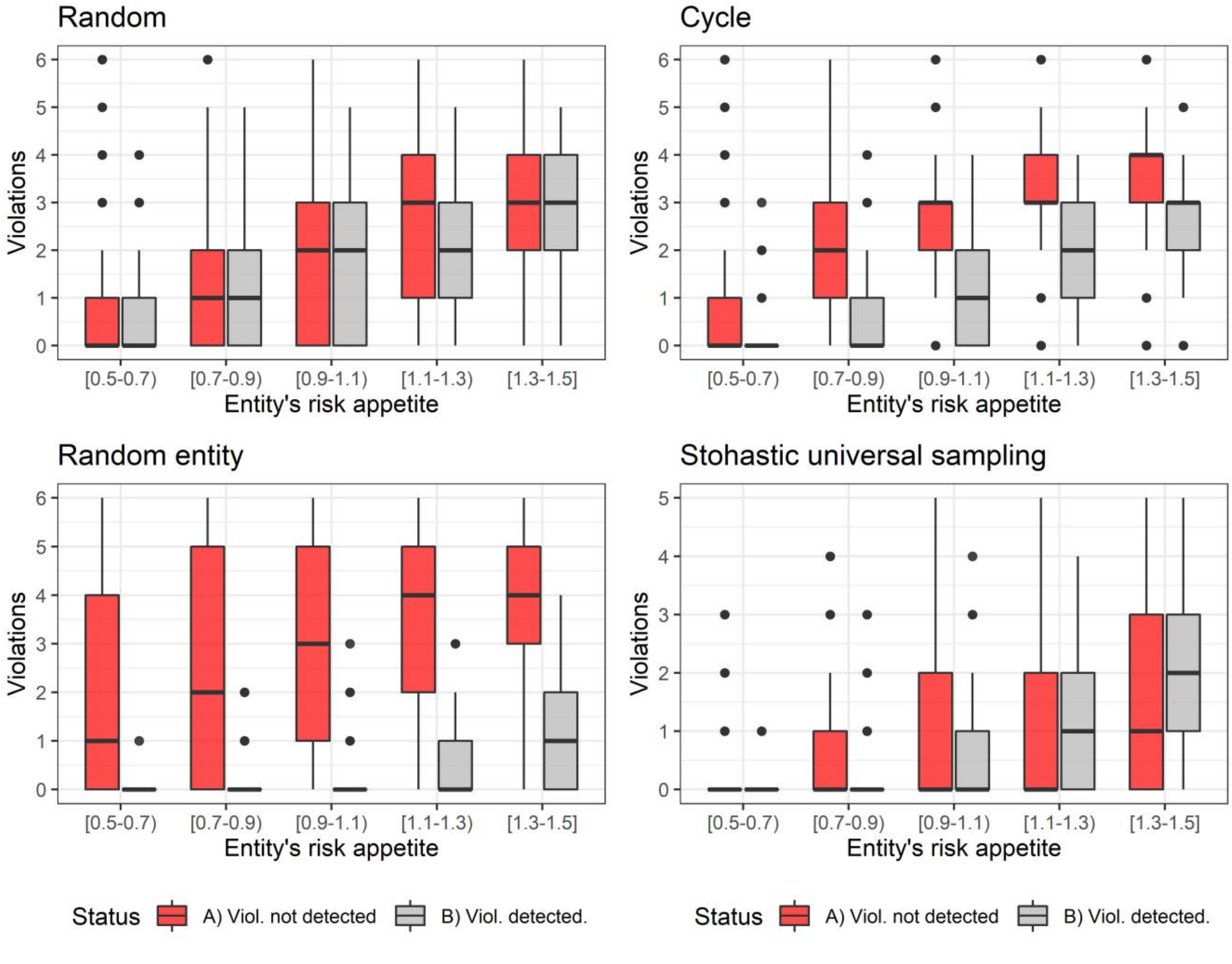

Figure 5.21 – General validation: comparison of the number of violations by individual entities that were not punished in the previous step of the simulation (group A – red column – contains data of entities whose inspection did not detect violations in the previous step, regardless of whether (1) the entity was not inspected at all, (2) the entity was inspected but only compliant provisions were covered by the inspection or (3) the inspection erroneously concluded that there are no violation present, although the entity actually violated inspected provisions) or have been punished (group B – gray column – contains data of entities that were caught violations); the x-axis shows the risk appetite ranges of the inspected entities grouped into 5 categories, and each bar presents a statistical summary of the violations of the provisions of the entities whose risk appetite is within the given range; each graph shows the data for the specified applied inspection strategy

Figure 5.21 shows the results of the validation test. Looking at the BP diagrams displayed from left to right, it is visible that as the entity's risk appetite grows, so does the level of violations. Furthermore, entities in the Group A violated more rules than entities in the Group B, when random entity and the cyclic inspection strategies were applied, for the same range of risk appetites. The same regularity is visible when the random selection strategy is applied, but the differences are marginal. This regularity is not visible when the stochastic universal sampling inspection strategy is applied. However, since – in line with the introductory remarks in chapter 5.2.1 – only the random entity strategy and the cyclical strategy are relevant for the model validation, it can be concluded that the test results are in line with the expectations.

5.2.1.8. General validation: A longer period since the last inspection leads to more violations

Empirical research shows that in situations where the compliance of entities with the rules has not been monitored for a long time, there is an increase in the number of violations. This regularity is explained in more detail in Chapter 3.2 and, specifically, in [30] and [32]. The validity test aims to determine whether this regularity can be observed in the simulation of the model.

Analogously to the previous 2 validation tests, this test also assesses the regularities in the behaviour of individual entities (micro-levels). The test was performed with the same input parameters as the test in section 5.2.1.6, with 2 modifications. The number of entities in the simulation was reduced to 20. A reduction in the number of entities was done to shorten the duration of the simulation, and the change does not affect the result. In applying the cyclical inspection strategy, the number of rules inspected in one cycle has been set to 6. This change ensures that, for a cyclical inspection strategy with inspection capacity of 0.2, two inspections of each entity will be "separated" by 6 steps (ticks). 30 simulation steps were performed in the simulation. The number of simulation steps has been increased from 25 to 30 to increase the number of data points where entities have not been inspected for a long time, when applying random inspection strategies.

The validity was assessed on the data that was collected in the following way. In each step of the simulation, data about one entity was recorded (one entity was randomly selected out of 20 entities) and the following data was collected: applied inspection strategy, number of steps since the entity was last inspected (from 0 onwards) and the number of violations in the current step (0 to 6). As in the previous two tests, the violations status was recorded regardless of whether the entity was inspected or not, at the time of data collection.

Furthermore, analogously to the previous two validation tests, 1,000 simulation iterations were performed with the same input parameters and the inspection capacity was set to 0.2. These values were set primarily with the aim of increasing the number of violations by entities that were not inspected in several consecutive steps, when applying random inspection strategies. As in the validation test in chapter 5.2.1.6, due to the characteristics of the random selection strategy, such data is relatively rare. Furthermore, validation of the stochastic universal sampling strategy cannot be performed by this test since in a 6-rule configuration and with inspection capacity of 0.2, each entity will be inspected at each step, and there will be no data points for entities that were not inspected in the previous step. For each inspection strategy, 30,000 data points were collected, given the 30 simulation steps.

The test was performed with the input parameters contained in the table below (Table 5.24).

Table 5.24 – Input parameters: a longer period since the last inspection leads to more violations

Input parameter:	Values:
type-of-inspection-selection	{Random, Random entity, Cycle}
number-of-entities	20
number-of-rules	6
resource-requirements-type	Input from line
resource-requirements	[1 3 5 7 9 11]
max-deviation-resources	0.25
default-risk-attitude	1
max-risk-attitude-deviation	50%
k-hyperbolic-discounting	0.5

punishment-size	40
inspectors-capacity	0.2
inspection-accuracy	90%
rules-inspected-in-one-cycle	6

Figure 5.22 contains the validation results shown in the BP diagram. The number of steps in which entities are not monitored varies, depending on the applied inspection strategy.

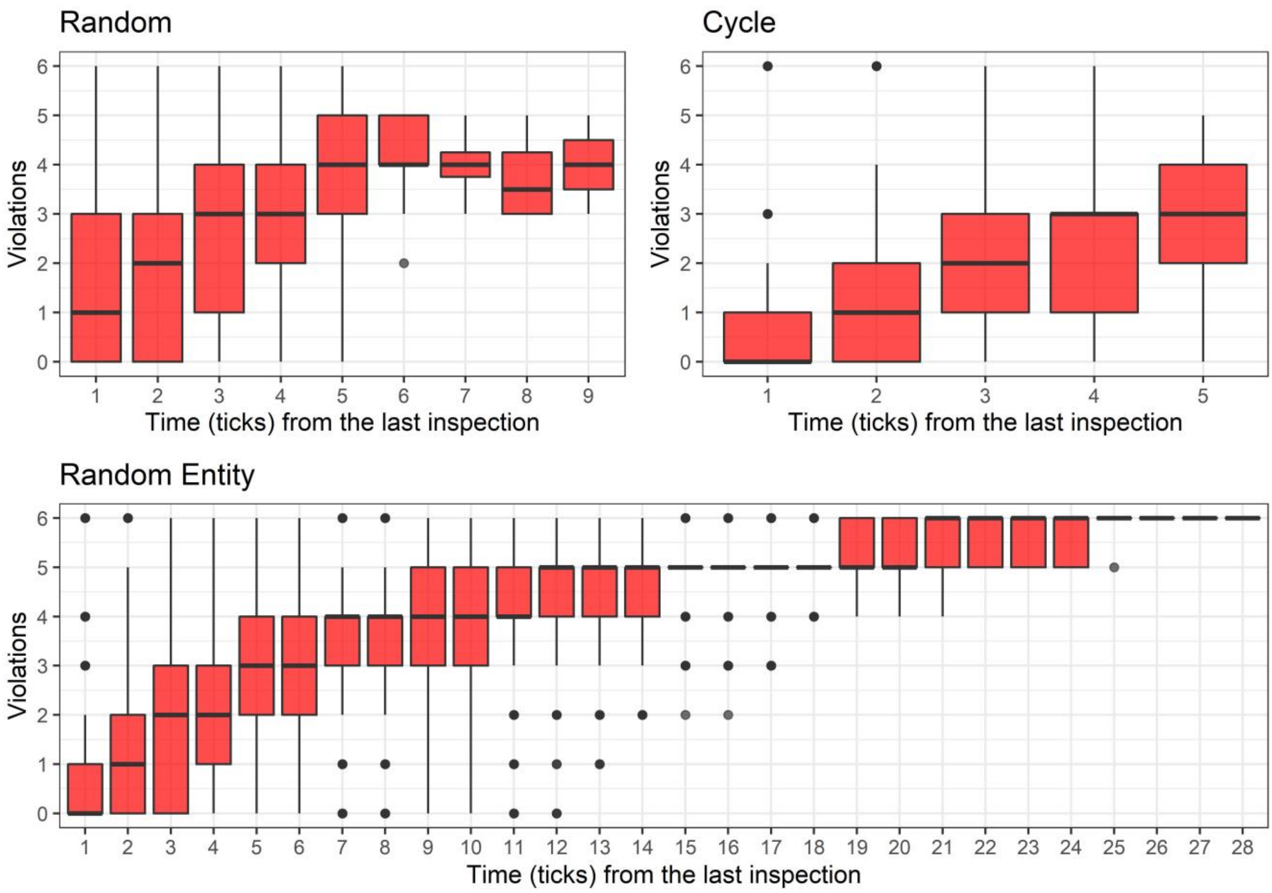

Figure 5.22 – Validation: comparison of the number of violations by individual entities (y-axis) that were not inspected during the last x (x-axis) simulation steps; each individual bar shows a statistical summary of the violation; each graph shows the data for the specified applied inspection strategy

When applying the cyclical inspection strategy, given the set number of rules inspected in one cycle, 6 steps will pass between every two inspections of the same entity. By applying the random entity selection strategy, each entity has a 20% probability of being inspected in each step (given the value of the inspectors-capacity parameter), and there could be entities that are not inspected at all. However, the entities will not be inspected in more than 20 consecutive moves only very rarely (for any given entity, $p \sim 1\%$). Therefore, after the value 23 on the x-axis, less than 25 data points were identified for each further value. By applying the random selection strategy, the occurrence of entities that are not inspected in several consecutive steps is significantly less common. That is, there are only 2-4 data points for situations where the entity is not monitored for 7 or more consecutive steps, and such data points can be considered as outliers. The BP diagrams show that the number of violations of rules increases with the number of consecutive steps in which the entity was not inspected. This regularity is visible for all 3 strategies, taking into account that data points for situations where the entity was not inspected for 7 or more consecutive steps while applying the random selection strategy are too rare to be statistically relevant.

5.2.2. Specific validation

The specific validity assessment encompassed analysis of quantitative agreement of the model with empirical macro-structures (2. level of validity according to Axtel and Epstein). Furthermore, the parametric space of the model was analysed and bounded based on the collected empirical data (3rd step in indirect calibration method). In other words, within the specific validity assessment, an attempt was made to find a valid combination of input parameters for which, within given limitations, the computer simulation of the model will reproduce the expected quantitative results. This included the following steps:

1. Quantitative indicators that should be replicated and limitations of the input parameters of the model were identified, based on the analysed empirical research.
2. Additional NetLogo procedures have been designed, to record the values of the identified quantitative indicators in the computer simulation of the model and to compare them with the expected values[26].
3. By applying a tool for the parameter space exploration, the values of the input parameters of the model were identified, for which the obtained (output) values of the considered quantitative indicators were sufficiently aligned with the expectations.
4. The computer simulation of the model was repeated 100 times with the input parameters identified in the previous step, and the results were compared with the expectations.

Given the size of the parameter space and the duration of the computer simulation, analysis of all possible combinations of input parameters was not feasible. Therefore, heuristic methods of parameter space exploration and searching for satisfactory solutions were applied. The parameter space exploration was tested using the simulated annealing [80, p. 25] and the genetic [80, p. 77] algorithms. The genetic algorithm had consistently better results (compared to simulated annealing). Hence, all specific validity checks were performed using the genetic algorithm. Mutation rate, population size, crossover rate, population model, and tournament size were all set in line with the recommendations of the author of the *BehaviorSearch* tool [68], that was used in the parameter space exploration. The mutation of each parameter was set separately, and depends on the type of the parameter.

The adequacy of the results was assessed via the defined objective ("fitness") function, i.e. by comparing the results of the simulation with the predefined (expected) values. The objective function is a one-dimensional global search function, which must contain all model variables whose values are evaluated. An example of the evaluation function as used to verify the specific validity of the developed model is described below.

Suppose that empirical data shows that after the passage of time analogous to υ simulation steps, it is determined that A entities violate rule 1, B entities violate rule 2, and C entities violate rule 3. The validation procedure should collect data on the number of entities that violate each of these 3 rules after υ simulation steps ($a(\upsilon)$, $b(\upsilon)$ and $c(\upsilon)$). The parameter space exploration procedure should identify the input parameters of the simulation for which the difference between the obtained and desired results is minimal. That is, a possible evaluation function for that simulation might be given by:

$$f(input\ parameters) = arg\ min(|A - a(\upsilon)| + |B - b(\upsilon)| + |C - c(\upsilon)|) \tag{23}$$

If additional determinants such as the relative importance of the expected results, standard deviations, etc. are known, they can also be included in the evaluation function (for example, by multiplying the differences between expected and obtained results with additional factors).

Specific validity of the ICARUS model was assessed on 3 case studies: the environmental protection inspections in Denmark (Scenario DK-E), inspections of compliance with the Occupational Safety and Health Administration regulations in the USA (Scenario US-W) and the supervision of banks in Italy (Scenario IT-B). The key data from these case studies, the relevance of the data for the validation of the model and the expected results are described below.

[26] Procedures for analysis and comparison of indicators are specific to a given situation, i.e. for quantitative indicators that should be reproduced – hence the name "specific" validity.

In all the scenarios, the parameter space exploration process included the following input parameters: `punishment-size`, `resource-requirements`, `max-deviation-resources`, `default-risk-attitude`, `max-risk-attitude-deviation`, `k-hyperbolic-discounting`, `inspection-accuracy` and `risk-exp`.

Table 5.25 contains the values of all input parameters (input parameters with pre-set values as well as the values of the remaining parameters, obtained through the exploration of parametric space). The details on the values obtained by the exploration process are explained in the following sub-chapters. Significant differences in some parameters are visible. E.g., it is interesting to note that the initial risk appetite (`default-risk-attitude`) varies significantly, from scenario to scenario, but in all scenarios it is greater than 1. Furthermore, the maximum deviation in risk appetite (`max-risk-attitude-deviation`) is relatively high in all 3 scenarios.

The parameters in the table were applied as input parameters of the relevant specific validation assessments, the results of which are described below.

Table 5.25 – Input parameters for the three analysed scenarios; input parameters with pre-set values are marked in light-grey

	Input parameter	Scenario		
		DK-E	US-W	IT-B
Pre-set values	inspectors-capacity	0.33	0.2	0.2
	learning-mechanism	Fictitious play		
	number-of-entities	100	200	200
	number-of-rules	6	10	4
	rules-inspected-in-one-cycle	6		
	stackelberg-aware	FALSE		
	type-of-inspection-selection	Cycle	Random entity	Random entity
Explored variables	default-risk-attitude	1.4	4.8	3
	inspection-accuracy	57	97	46
	k-hyperbolic-discounting	0.1	0.1	0.2
	max-deviation-resources	0.4	0.25	0.31
	max-risk-attitude-deviation	82	96	94
	punishment-size	136	166	208
	resource-requirements	[23 12 14 5 8 11]	[1.8 1.4 1.8 1.1 1.7 1.9 9.6 9.3 9.7 9.6]	[1.5 6 5.5 15]
	risk-exp	FALSE		

5.2.2.1. Specific validation: DK-E scenario: Inspection of environmental regulations in Denmark

Winter and May analysed empirical data on the compliance of Danish farmers with environmental regulations and published their findings in 2 papers: [41] and [37]. The research was based on a case study in which the compliance of livestock breeders (total population: 45,000) with the newly adopted environmental regulations was analysed. In the analysed case, local supervisory authorities had to conduct inspection of compliance of all farmers with the adopted regulations within 3 years. The collected empirical data includes the results of inspections conducted by local supervisory authorities (in each of Denmark's 258 municipalities), official data from the national environmental agency and responses to a survey sent by the authors to 1,652 Danish farmers. Based on the collected data, the authors conducted a comprehensive analysis. Part of the data from the analysis, which is conceptually harmonized and applicable to the ICARUS model, were used in the assessment of model parameters and in validation.

Table 5.26 contains the considered indicators and their expected values. The DKv1 indicator refers to the percent of entities that comply with all the provisions at all times (i.e. at each step of the simulation). This includes entities whose compliance was determined by inspection as well as entities that were not inspected in a particular inspection step. The DKv2 indicator refers to the percent of entities (in the entire population) that complied with all the provisions after the last step of the simulation (i.e., at the end of simulation). Indicators DKv3A, DKv3B and DKv3C refer to the entire simulation period and to all the entities.

Table 5.26 – Analysed indicators and their expected values in the DK-E validation scenario

Indicat.	Description	Expected
DKv1	% of entities that comply with all the provisions at all times	26 %
DKv2	% of entities that have complied with all provisions after the last cycle	73 %
DKv3A	How often is rule A violated (including detected and undetected violations)	18 %
DKv3B	How often is rule B violated (including detected and undetected violations)	5 %
DKv3C	How often is rule C violated (including detected and undetected violations)	3%

The values of the input parameters of the simulation were determined in one of 3 ways:

- they were set in line with the data contained in the cited publications ([41] and [37]),
- they were set arbitrarily (if their value was not relevant to the validation test or if changing their value should not affect validity of the model) or
- their value was determined based on the exploration of the parametric space.

The number of entities was (arbitrarily) set to 100, as changing the number of entities should not affect the validity of the model. The number of rules is set at 6, as the publications identify 6 key areas of inspection [41, p. 648]. The inspection agency conducts a cyclical inspection, whereby all 6 rules are inspected in one cycle, and the capacity of the inspection agency is 0.33. These parameters were set with regard to the requirement that all obligees must be inspected at least once within 3 years [41, p. 648]. That is, these values ensure that the inspection agency will inspect compliance of each entity with every rule once every 3 years.

Validity was assessed by recording the simulation results after 6 steps (i.e. after 2 full cycles). The initial cycle was conducted to establish a realistic initial perception of the entities on the likelihood of inspections (taking into account the dependence of the fictitious play algorithm on the initial assumptions).

Table 5.25 contains the input parameters of the simulation. Table 5.27 contains a comparison of the results of the computer simulation of the model and the expected values.

Table 5.27 – Specific validation: expected and simulated results for each considered criterion (indicator), for DK-E scenario; in the column "Achieved" is the arithmetic mean of the results ($\overline{x}$) of 100 repetitions of the simulation with the same input parameters, in the column "Diff." is the absolute difference (percentage points) of achieved and expected results, and the column "95% CI" states 95% confidence interval, based on the calculated standard error.

Indicat.	Description	Expected	Achieved	Diff.	95%CI
DKv1	% of entities that comply with all the provisions at all times	26 %	25,0 %	1,0 %	±8,6%
DKv2	% of entities that have complied with all provisions after the last cycle	73 %	77,4%	-4,4 %	±8,7%
DKv3A	How often is rule A violated (including detected and undetected violations)	18 %	19,5 %	-1,5 %	±7,5%
DKv3B	How often is rule B violated (including detected and undetected violations)	5 %	5,4 %	-0,4 %	±4,3%
DKv3C	How often is rule C violated (including detected and undetected violations)	3%	3,2 %	-0,2 %	±3,3%

The presented results show that the differences between expected and achieved values are relatively small, and all differences are within the acceptable 95% confidence range. In accordance with all of the above, it can be concluded that the achieved results are in line with the expectations.

5.2.2.2. Specific validation: US-W scenario: USA occupational safety inspections

Ko, Mendelhof and Gray in their work [32] present broad statistical data on inspections of compliance with the occupational safety regulations in the USA. These inspections are conducted by the federal agency OSHA (Occupational Safety and Health Administration) in 26 USA states. The data contained in the study are based on the official OSHA data on 549,398 compliance inspections conducted from 1972 to 2006. According to the authors, numerous studies have shown which OSHA data are credible and the authors use a subset of that data. Part of the data from this publication, which is conceptually harmonized and applicable to the ICARUS model, were used in the assessment of model parameters and in the model validation.

Table 5.28 contains the analysed indicators, their descriptions and the ranges of values observable in the empirical data. The USv1 indicator refers to the percent of entities that did not comply with at least one rule during the first inspection (the standard deviation of the parameters is not presented in the empirical data). Indicators USv3A, USv3B and USv3C refer to the average decrease in the number of violations committed by individual entities, which were detected by inspections. These 3 indicators are defined through target ranges, and the validation procedure searched for a combination of input parameters for which the obtained values will be within the stipulated ranges. All 4 indicators refer only to the entities that were inspected for compliance.

Table 5.28 – Analysed indicators and their expected values in the US-W validation scenario

Indicator	Description	Expected
USv1	% of entities that do not comply with one or more rules at the time of the first inspection.	83 %
USv3A	Average decrease in the number of violations identified in the 2nd inspection compared to the 1st inspection	40-50 %
USv3B	Average decrease in the number of violations identified in the 3rd inspection compared to the 2nd inspection.	5-20 %
USv3C	Average decrease in the number of violations identified in the 4th inspection compared to the 3rd inspection.	5-20 %

As in the previous test, the values of the input parameters of the simulation were set in line with the data contained in the cited publication [32], they were set arbitrarily or were determined based on the exploration of the parametric space.

The number of entities is arbitrarily set to 200 and the number of rules is set to 10. As in the previous test, changing the number of entities should not affect the ability to validate the model. In this validity test, the number of rules was determined arbitrarily, as no relevant empirical data is available. The inspection agency implements the inspection strategy by random selection of entities, i.e. compliance with all the rules is assessed at the entities selected for inspection. The capacity of the inspection agency is 0.2. In line with the above, the inspection agency randomly selects 20% of the entities in each period and inspects compliance of these entities with all the rules. These inspection strategy parameters were set to mimic the data presented in [32]. It should be emphasized that the test sets a relatively large number of entities and rules due to the characteristics of the validity test itself. Namely, the validation test analyses, among other things, changes in compliance of an individual entity. Setting more entities and rules ensures that there are enough data points to assess – for example – the number of violations observed in the 4th inspection and compare it with the observations of the 3rd inspection. On the other hand, it is desirable to shorten the total duration of the simulation, given the large number of simulations that are conducted as part of the specific validation assessment.

Due to the need for shortening the total duration of the simulation and the characteristics of the selected inspection strategy and inspection capacity, the validity was assessed by analysing the results recorded in the last 22 steps of the simulation. Since the entity's perception of the probability of an inspection depends on its history, the initial 10 steps of the simulation represent a kind of initialization, i.e. they were conducted in order to establish a realistic initial perception of an entity on the probability of inspection. Thereafter, a further 12 simulation steps were performed (steps 11 to 22), and the validation test included only entities that were not inspected in steps 8, 9, and 10, to simulate the effect of the first inspection. Finally, the validation test analysed only the data generated in the last 12 steps of the simulation, taking into account the characteristics of the empirical data.

Table 5.25 contains the input parameters. Table 5.29 contains a comparison of the results of the computer simulation of the model and the expected values.

Table 5.29 – Specific validation: expected and simulated results for each considered criterion (indicator), for the US-W scenario; in the column "Achieved" is the arithmetic mean ($\bar{x}$) of the results of 100 repetitions of the simulation with the same input parameters

Indicator	Description	Expected	Achieved
USv1	% of entities that do not comply with one or more rules at the time of the first inspection.	83 %	85.0 %
USv3A	Average decrease in the number of violations identified in the 2nd inspection compared to the 1st inspection	40-50 %	41.8 %
USv3B	Average decrease in the number of violations identified in the 3rd inspection compared to the 2nd inspection.	5-20 %	19.0 %
USv3C	Average decrease in the number of violations identified in the 4th inspection compared to the 3rd inspection.	5-20 %	17.0 %

The presented results show that difference between the expected and realized value of the USv1 indicator is relatively small – only 2 percentage points. The obtained results for indicators USv3A, USv3B and USv3C are within given ranges. In line with all of the above, it can be concluded that the achieved results are sufficiently in line with the expectations.

5.2.2.3. <u>Specific validation: IT-B scenario: Supervision of banks in Italy</u>

Brogi [29] and Muré and Pesic [79] pooled and analysed data on bank supervision in Italy collected and published by Banca d'Italia (central bank of Italy). Among other activities, Banca d'Italia supervises banks and, within that, conducts on-site supervision (performs inspections) of Italian banks. Both cited studies focused on statistics on inspections and administrative penalties carried out and issued by Banca d'Italia between 1998 and 2009. Administrative penalties are issued when the supervised bank does not comply with certain regulations. Both publications are based on quantitative data on inspections of around 900 Italian banks that had banking licenses in that period. Part of the data from these publications, which are conceptually harmonized and applicable to the developed model, were used in the assessment of model parameters and in validation.

Analogous to the previous two tests, the values of the input parameters were set in line with the data contained in the cited publications [29][79], the values were set arbitrarily or were determined based on the exploration of the parametric space. Table 5.30 contains the analysed indicators, their descriptions and the ranges of expected values (based on the analysis of the empirical data).

Table 5.30 – Analysed indicators and their expected values in the IT-B validation scenario

Indicator	Sanctioned in the observed period	Expected
ITv1	Once	61-73 %
ITv2	2 times	19 -26 %
ITv3	3 times	8-10 %
ITv4	4 times	1-3 %
ITv5	5 times	0-2 %

The inspection agency implements the random entity selection inspection strategy, i.e. compliance with all the rules is assessed in the selected entities. The capacity of the inspection agency is 0.2, in line with the empirical data according to which Banca d'Italia carried out direct supervision of 20% of banks each year (on average, with small deviations). The number of entities was arbitrarily set to 200, and the number of rules was set to 4. As in the previous two tests, the change in the number of entities should not affect the validity of the model. Analogous to the previous chapter, the number of entities is set as relatively large in order to obtain a sufficient number of data points to conduct the validation test. Due to the given inspection strategy and inspection capacity, entities that will be monitored 5 times in 12 years will be relatively rare. The number of rules was also set arbitrarily, as no relevant empirical data was available. However, the number of rules is set lower than in the validation test in the previous chapter, as the number of rules should not affect the feasibility of the test and given the desirable shortening of the total simulation duration.

As in the test described in the previous chapter, the initial 10 steps of the simulation were conducted to establish a realistic initial perception of the entities on the likelihood of inspection, since the assessment of every entity on the likelihood of inspection depends on the history (fictitious play learning strategy). In the environment that was analysed in the cited studies, the entities (banks) already operated in that environment prior to the data collection and were subject to inspections by the same inspection agency (Banca d'Italia). After the initialization period, a further 12 simulation steps (based on the empirical data; steps 11 to 22) were performed, the results of which were analysed in the validation test.

All considered indicators (Table 5.31) refer to the number of sanctions (penalties) observed in the 12 analysed simulation steps. Namely, after these 12 steps, the percent of entities that were sanctioned only once, 2 times, 3 times, 4 times or 5 times was determined. Sanctioning means that during the inspection, it was determined that the entity violates one or more provisions. Ranges of expected values were identified in the empirical data, and in the validation test an attempt was made to find combinations of input parameters for which the obtained values will be within the given ranges. All indicators refer to all the entities, i.e. not only to the subset of entities that were inspected.

Table 5.25 contains the input parameters for the simulation. Table 5.31 contains a comparison of the results of computer simulation of the model and the expected values. All considered indicators are within the expected ranges. In line with all of the above, it is can be concluded that the achieved results are in line with the expectations.

Table 5.31 – Specific validation: expected and simulated results for each considered criterion (indicator), for the IT-B scenario; in the column "Achieved" is the arithmetic mean ($\bar{x}$) of the results of 100 repetitions of the simulation with the same input parameters

Indicator	Sanctioned in the observed period	Expected	Achieved
ITv1	Once	61-73 %	65.1 %
ITv2	2 times	19-26 %	25.6 %
ITv3	3 times	8-10 %	8.2 %
ITv4	4 times	1-3 %	1.1 %
ITv5	5 times	0-2 %	0.1 %

5.2.3. Sensitivity analysis

The sensitivity analysis of the developed model was performed using the Morris' elementary effects screening method [76]. Test parameters were set in line with the recommendations of Thiele and co-authors [73].

The impact of (changes in) 10 input parameters on the total number of violations at the end of the simulation was analysed, for each inspection strategy[27]. The considered ranges of values are displayed in the Table 5.32. The test was conducted in the following manner. 100 random values were selected from each range of input parameters, and the model simulation was ran with every parameter combination. After 25 simulation steps, the average total (cumulative) number of violations was recorded. For each combination of parameters, 50 elementary effects were calculated, and the simulation was repeated 50 times for each combination of input parameters.

Table 5.32 – Input parameters for sensitivity analysis of the ICARUS model

Input parameter	Minimum	Maximum
default-risk-attitude	0.1	6
inspection-accuracy	30	100
inspectors-capacity	0.1	0.5
k-hyperbolic-discounting	0.1	5
max-deviation-resources	0.01	0.5
max-risk-attitude-deviation	0.01	0.99
number-of-entities	10	200
number-of-rules[28]	3	15
punishment-size	30	200
resource-requirements-param	1	5
rules-inspected-in-one-cycle[29]	1	10

The graphs below (Figure 5.23 – Figure 5.27) display the results of the Morris' elementary effects screening method for the 5 inspection strategies that were considered in the test (the SUS strategy was ran with and without Stackelberg leadership). Each graph consists of two parts. The first figure shows the ratio of μ and μ^* which provides information on the magnitude of the influence of given parameters on the results of the model and the direction (sign) of these influences. The second figure shows the ratio of σ and μ^* and provides information on the variation of the influence of given parameters on the results, i.e. to what extent the influence of each parameter depends on the values of other parameters.

In each figure, 10 distinct letters mark 10 (analysed) input parameters. The placement of a letter signifies the impact of a particular input parameter on the results of the sensitivity analysis. The same input parameters are assigned to the same letters in all the figures except for the Figure 5.25, since the input parameters applicable to the cyclical selection strategy (displayed in Figure 5.25) differ somewhat from the input parameters applicable to other inspection strategies. Furthermore, the scaling of the x-axis and the y-axis is adjusted to the values of the indicators, and significant differences in the maximum values of σ are visible on different graphs.

[27] Depending on the scenario, 10 of the 11 possible parameters were analysed.
[28] The values of this parameter were changed in all inspection strategies except the cyclic selection strategy.
[29] The value of this parameter is relevant only for the cyclical selection inspection strategy. When applying this strategy, the value of the `number-of-rules` parameter is set to 10.

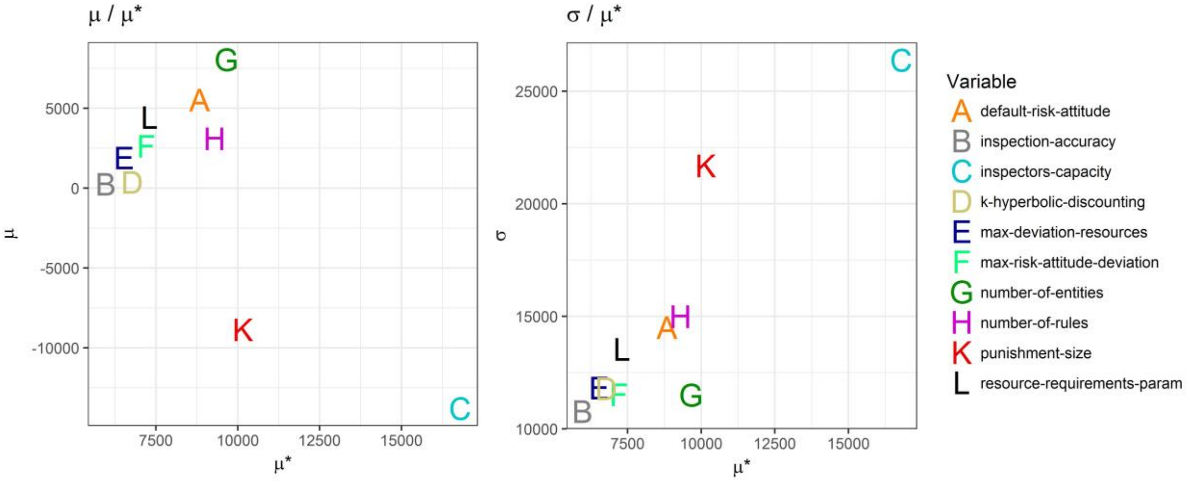

Figure 5.23 – Sensitivity analysis: sensitivity to changes in the 10 input parameters when applying the random entity selection strategy; μ is the arithmetic mean of the elementary effect, μ^* is the estimation of the arithmetic mean of the absolute values, and σ is the standard deviation

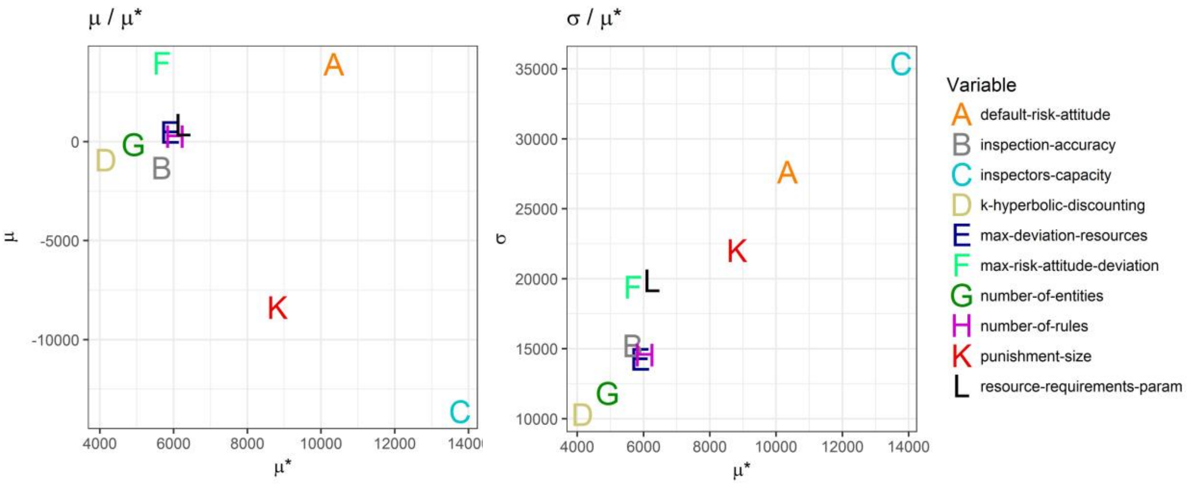

Figure 5.24 – Sensitivity analysis: sensitivity to changes in the 10 input parameters when applying the random selection strategy; μ is the arithmetic mean of the elementary effect, μ^* is the estimation of the arithmetic mean of the absolute values, and σ is the standard deviation

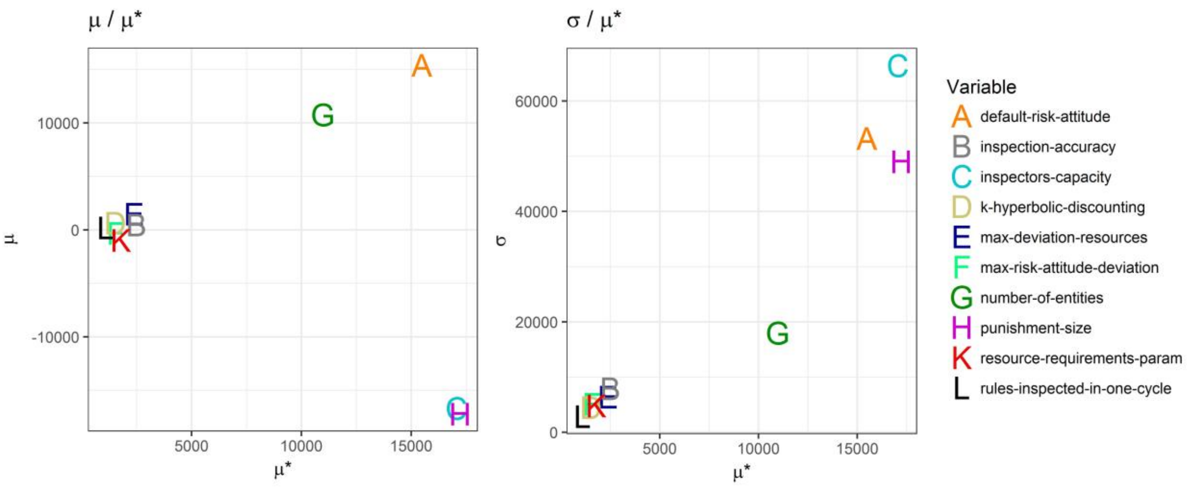

Figure 5.25 – Sensitivity analysis: sensitivity to changes in the 10 input parameters when applying the cyclical selection strategy; μ is the arithmetic mean of the elementary effect, μ^* is the estimation of the arithmetic mean of the absolute values, and σ is the standard deviation

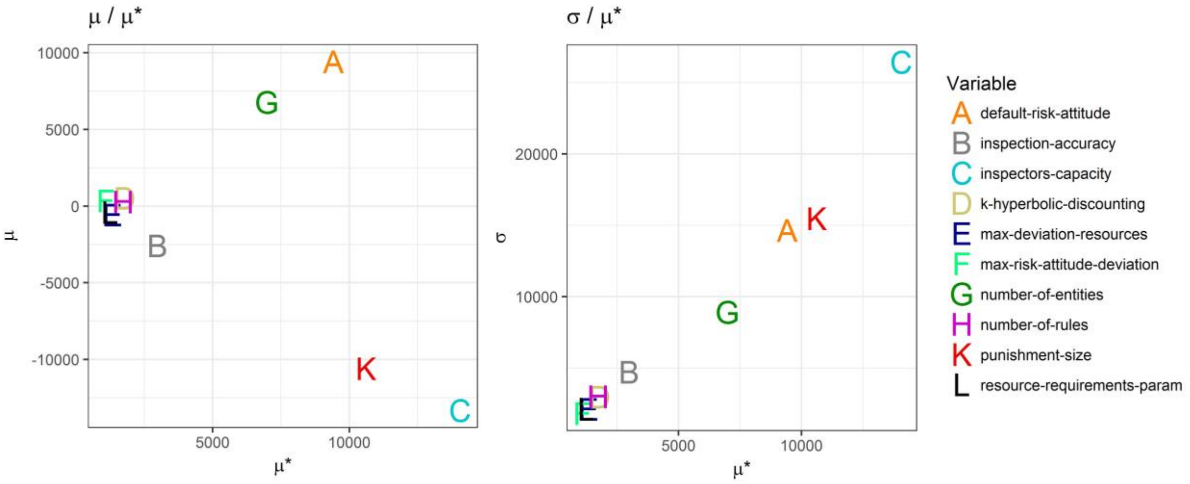

Figure 5.26 – Sensitivity analysis: sensitivity to changes in the 10 input parameters when applying the stochastic universal sampling (SUS) strategy; μ is the arithmetic mean of the elementary effect, μ^* is the estimation of the arithmetic mean of the absolute values, and σ is the standard deviation

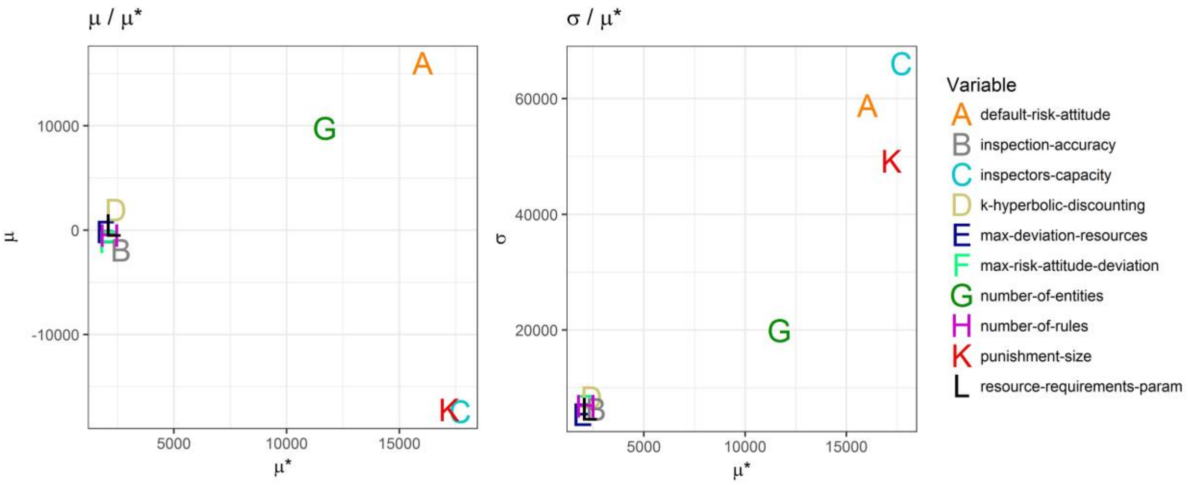

Figure 5.27 – Sensitivity analysis: sensitivity to changes in the 10 input parameters when applying the SUS strategy with Stackelberg leadership; μ is the arithmetic mean of the elementary effect, μ^* is the estimation of the arithmetic mean of the absolute values, and σ is the standard deviation

The presented figures support a number of conclusions about the influence of input parameters on the results of the model. Firstly, figures do not show a situation in which μ is small and μ^* is high. That is in line with the expectations, since such a situation would suggest that the direction of the influence of a parameter on the results of the model changes. That is, such a combination would mean that changes in parameter values do not have a consistent impact on the direction of changes in the number of violations. An example of such a situation would be an increase in the penalty that, in some cases leads to a decrease and in other cases to an increase in the number of offenses.

Furthermore, the figures do not shaw a situation in which σ is small and μ^* is high. I.e. the impact of any input parameter on the number of violations is not independent of the values of other input parameters. This result is also in line with the expectations, given that the theoretical background and validation show that different parameters have an impact on the total number of violations.

Several input parameters have a small or negligible impact on the number of violations. These include the following: `inspection-accuracy`, `k-hyperbolic-discounting`, `max-deviation-resources`, `max-risk-attitude-deviation`, and `number-of-rules`.

The `inspectors-capacity` parameter consistently has the greatest impact on the simulation results. That is, increase in the inspection capacity has a large (and monotonous) impact on reduction of the number of violations. This parameter is also characterized by a high σ, which suggests that the influence of this parameter on the simulation results strongly depends on the values of other model parameters.

The impact of the `punishment-size` parameter on the simulation results is very similar to the impact of `inspectors-capacity`. Namely, although the size of the impact varies – depending on the chosen inspection strategy – the increase in punishment undoubtedly causes a decrease in the number of violations. The value σ indicates that the impact of penalty on the number of violations depends on the value of other parameters, but the size of the impact is less than for the `inspectors-capacity` parameter.

The third parameter with a pronounced effect on the simulation results is the initial risk attitude (`default-risk-attitude`). Unlike the previous 2 parameters, an increase in the initial propensity to take risks leads to an increase in the number of violations in the system. This impact is significant and monotonous. As for the previous 2 parameters, the value σ indicates that the influence of the parameter on the results depends greatly on the values of other parameters.

The `number-of-entities` parameter has a significant impact on the results of the model when applying all strategies except the random selection strategy. The reasons are clear – a larger number of entities, with other parameters unchanged, leads to a higher absolute (but not relative) number of violations in the system.

6. Simulation results

This chapter contains graphical representation and statistical analyses of the data generated through computer simulation of the ICARUS model, which includes the application of descriptive and inferential statistical methods and testing of the hypotheses.

6.1. Analyses and testing methods

Analyses of simulation data and testing of hypotheses were performed in 2 steps. The first step, included descriptive data analysis (graphical presentation of the simulation results), and the second step involved performing appropriate statistical tests and drawing conclusions. The hypotheses were tested on the results of a computer simulation of the ICARUS model that were obtained for different combinations of input parameters.

In testing H.1, the results of the inspections of compliance in proportion to the compliance costs, i.e. the application of the stochastic universal sampling strategy (*SUS*) were compared with the results of inspections using two random strategies: random selection inspection (*Random*) and random entity selection inspection (*Random entity*).

In testing H.2, the results of the inspections using stochastic universal sampling strategy (*SUS*) were compared with the results of the application of the cyclic selection strategies. Two variants of cyclic selection strategies were considered. In the first variant (*Cycle*), all the rules are inspected in one cycle, i.e. the cycle is completed only when the inspections have covered all the rules in all the entities. In the second variant (*Cycle-S*), half of the rules are inspected in one cycle, i.e. inspections first inspect half of the rules in all theentities, and then inspect the remaining rules in all the entities.

In testing H.3, the results of the inspections using stochastic universal sampling (*SUS*) strategy were compared with the results when applying the same strategy, but with Stackelberg leadership. That is, when assessing the probability of inspection of compliance with a specific rule, an entity assume that the frequency of inspection of that rule is proportional to (its) costs of compliance (*SUS-Stackelberg*).

In line with all of the above, the results of simulations with 6 inspection strategies (*SUS*, *Random*, *Random entity*, *Cycle*, *Cycle-S* and *SUS-Stackelberg*) were compared in the tests of hypotheses. The dependent variable in all the tests is the total number of violations in the system.

Given the large number of possible combinations of input parameters, it was necessary to identify a subset of parameters with which to test the set hypotheses. In the specific validation tests (Chapter 5.2.2), 3 combinations of input parameter values (DK-E, US-W and IT-B scenarios) were identified with which the computer simulation of the ICARUS model reproduced the quantitative results identified in the empirical research. Therefore, the posed hypotheses were tested in DK-E, US-W and IT-B scenarios, with input parameters set accordingly.

After starting the simulation, 25 simulation steps were conducted for each scenario and for each inspection strategy. The total number of violations in the system was then recorded. Simulations were repeated 100 times, with every combination of input parameters, given the stochastic nature of the ICARUS model and in accordance with the already mentioned recommendation [57, p. 111].

The applied descriptive methods include quantile-quantile (QQ) comparison diagrams, boxplot (BP) diagrams, and scatterplot diagrams. The QQ diagrams were used to compare the expected values (x-axis) and the values obtained by the simulation (y-axis) and enabled assessment of whether the data generated in the simulation deviates significantly from the normal distribution. The BP diagrams provide a statistical summary of the generated data.

The graphical representations of the simulation results for each scenario consist of two parts. The first (upper) part shows the quantile-quantile (QQ) comparisons of each sample, and the second (lower) part of the graph presents a comparative display of results of 6 implemented inspection strategies. The names of the data groups (applied inspection strategies) are on the x-axis of each graph, and the number of violations is presented on the y-axis. Each point represents a recorded number of violations (`rp-violations`) after 25 simulation steps, for a given inspection strategy. The horizontal jitter is not based on the generated data, but is presented to enable a better insight into the data distribution.

The hypotheses were tested in 2 steps: using the Kruskal-Wallis test (one-way analysis of variance by ranks or H test) and, after determining the significance of the results of that test, the Nemenyi *post-hoc* test.

The Kruskal-Wallis test was applied since the collected data did not meet all the prerequisites for the application of the one-way analysis of variance (ANOVA), i.e. it could not be presumed that the populations from which the samples were drawn have a normal distribution [94, p. 685]. The Kruskal-Wallis test is a nonparametric test that can be applied to samples (i.e. dependent variables) that do not meet the stated prerequisite for the use of one-way ANOVA. The Kruskal-Wallis test has less statistical power than one-way ANOVA, but this difference is small [95, p. 97]. The Tukey and Kramer test (also known as the Nemenyi test) was used to perform the so-called *post-hoc* test – to compare pairs of samples and determine the significance of their differences [96, p. 3] – which was performed only if the results of Kruskal-Wallis test were significant. Since the results also contained ties, χ^2 approximation was applied [96].

6.2. DK-E scenario

Figure 6.1 shows the results of inspections for the 6 described inspection strategies, for the DK-E scenario. The QQ graphs show that the results of simulations with random and cyclic strategies do not deviate significantly from the normal distribution. However, the results of inspections using the *SUS-Stackelberg* strategy significantly deviate from the normal distribution. An explanation of this deviation can be derived from the graph below.

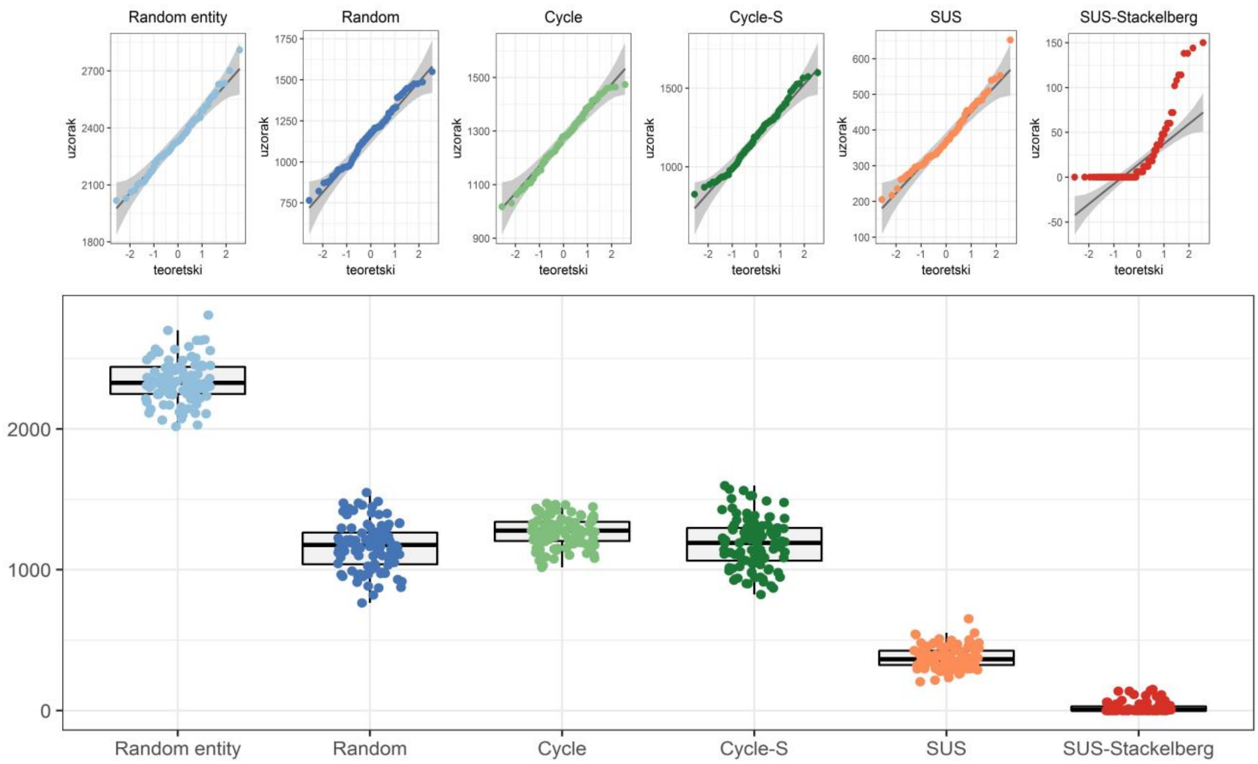

Figure 6.1 – Simulation results for the DK-E scenario, for 6 selected inspection strategies; the top figures display QQ charts – the shaded area is 95% confidence interval; the lower figure displays the results of 100 simulations and corresponding BP diagrams for 6 applied inspection strategies; the horizontal shift (jitter) of the individual results is only for visualization purposes

The presented data shows that *SUS-Stackelberg* strategy is so successful that (with the given input parameters) in many inspections there were no violations, even after 25 simulation steps. Since there cannot be less than 0 violations, a large number of inspections with 0 violations affect the characteristics of the distribution, i.e. they distort it. The results shows that inspections with the *SUS* strategy are more successful than inspections in which random and cyclic strategies were applied. I.e., simulations with the *SUS* strategy resulted in a lower total number of violations. Inspections with the *SUS-Stackelberg* strategy resulted in even fewer violations. The strategy with the (significantly) worst results is the strategy of random selection of entities (*Random entity*). Simulations in which *Cycle* and *SUS* strategies have been applied have less variance than strategies in which *Random*, *Random entity* or *Cycle-S* strategies have been applied, and simulations in which *SUS- Stackelberg* inspection strategy has been applied have the least variance.

Following the conclusions drawn from the presented graphs, the Kruskal-Wallis H test was performed. The results of the test show that there are statistically significant differences between the data groups ($\chi^2 = 522.37;\ df = 5;\ p < 2.2 \cdot 10^{-16}$). The *post-hoc* analysis using the Nemenyi test, with χ^2 approximation, gives the results shown below (Table 6.1):

Table 6.1 – Comparison of the pairs of simulation result for the DK-E scenario applying the Nemenyi test, with χ^2 approximation, for independent samples

	Cycle	Cycle-S	Random	Random entity	SUS
Cycle-S	0.5990	-	-	-	-
Random	0.3461	0.9990	-	-	-
Random entity	$1.2 \cdot 10^{-8}$	$1.0 \cdot 10^{-14}$	$< 2 \cdot 10^{-16}$	-	-
SUS	$< 2 \cdot 10^{-16}$	$2.1 \cdot 10^{-11}$	$5.1 \cdot 10^{-10}$	$< 2 \cdot 10^{-16}$	-
SUS-Stackelberg	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$	0.0052

Comparisons of the *SUS* strategy with random and cyclic strategies (4th row of Table 6.1) and comparison of *SUS-Stackelberg* and *SUS* strategy (cell at bottom-right of Table 6.1) are relevant for the posed hypotheses. The presented results show statistically significant differences in the number of violations in the application of the *SUS* strategy compared to the application of the *Random*, *Random entity*, *Cycle* and *Cycle-S* strategies. Furthermore, the results show that there are statistically significant differences in the application of the *SUS-Stackelberg* strategy compared to the *SUS* strategy.

6.3. US-W scenario

Figure 6.2 shows the results of simulations of the 6 set inspection strategies for the USA occupational safety inspections scenario (US-W). From the QQ graph it can be observed that the results of the *Random* strategy simulations differ significantly from the normal distribution. The graph below provides information on the simulation results. Firstly, it is evident that the random selection of entities strategy (*Random entity*) again provides the worst results, i.e. the application of this strategy achieves the highest number of violations of the rules. Furthermore, as in the previous scenario, simulations in which the *SUS* strategy was applied are more successful than simulations in which random and cyclic strategies were applied. However, simulations in which the *SUS-Stackelberg* strategy was applied resulted in a higher number of violations than simulations in which the *SUS* strategy and even the *Random* strategy were applied. The results of simulations in which the *SUS -Stackelberg* strategy was applied show a large variance: the total number of violations varies from 9,000 to over 16,000.

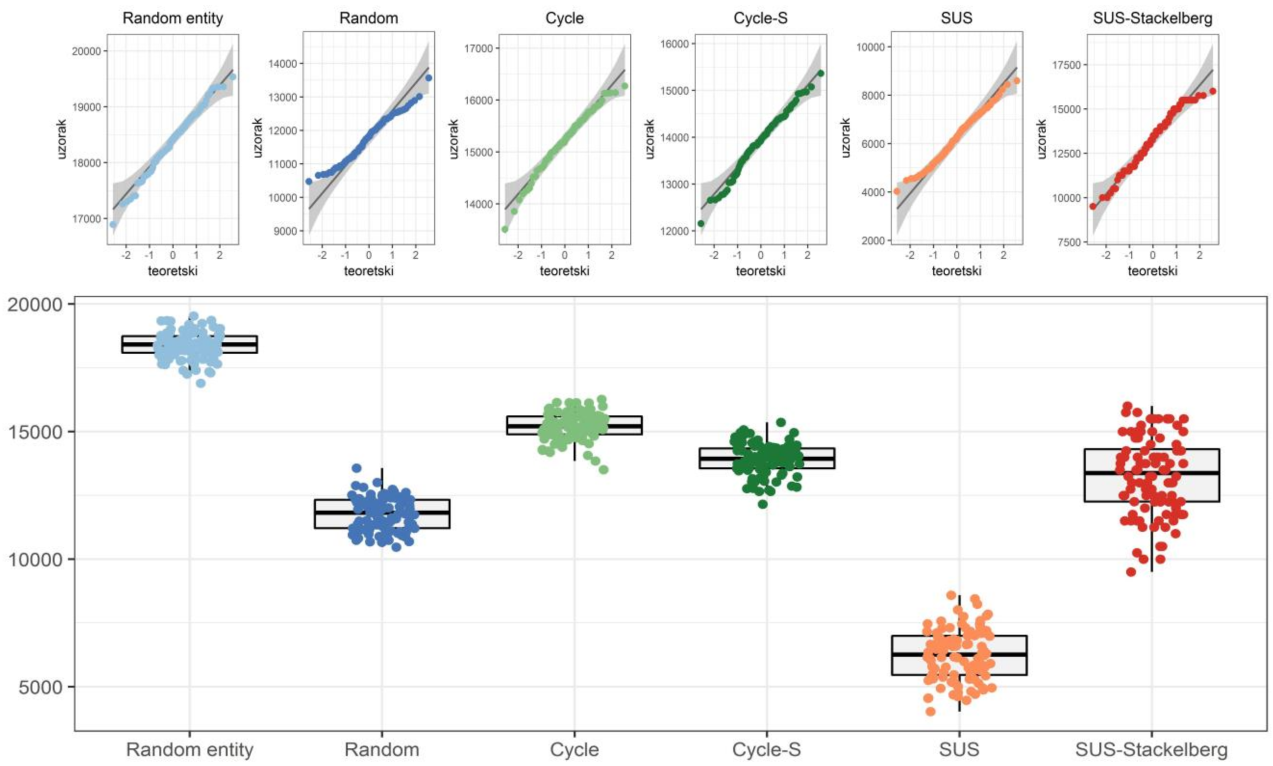

Figure 6.2 – Simulation results for the US-W scenario, for 6 selected inspection strategies; the top figures display QQ charts – the shaded area is 95% confidence interval; the lower figure displays the results of 100 simulations and corresponding BP diagrams for 6 applied inspection strategies; the horizontal shift (jitter) of the individual results is only for visualization purposes

The results of the Kruskal-Wallis H test show that there are statistically significant differences between the data groups ($\chi^2 = 532.71;\ df = 5;\ p < 2.2 \cdot 10^{-16}$). The *post-hoc* analysis using the Nemenyi test, with χ^2 approximation, gives the results shown below (Table 6.2):

Table 6.2 – Comparison of the pairs of simulation result for the US-W scenario applying the Nemenyi test, with χ^2 approximation, for independent samples

	Cycle	Cycle-S	Random	Random entity	SUS
Cycle-S	0.00051	-	-	-	-
Random	$< 2 \cdot 10^{-16}$	$9.7 \cdot 10^{-7}$	-	-	-
Random entity	0.00035	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$	-	-
SUS	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$	0.00020	$< 2 \cdot 10^{-16}$	-
SUS-Stackelberg	$2.3 \cdot 10^{-7}$	0.78994	0.00140	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$

The presented results confirm statistical significance of the conclusion that *SUS* strategy achieves a smaller number of violations than the application of *Random*, *Random entity*, *Cycle* or *Cycle-S* strategies (4th row of Table 6.2). Since it was already evident from the analysis of the visual presentation of the data that the application of the *SUS-Stackelberg* strategy does not achieve a lower number of violations than the application of the *SUS* strategy, the statistical significance of the results in the cell at bottom-right of Table 6.2 is not relevant.

6.4. IT-B scenario

Figure 6.3 shows the results of simulations of the 6 set inspection strategies for the Italian banking supervision scenario (IT-B).

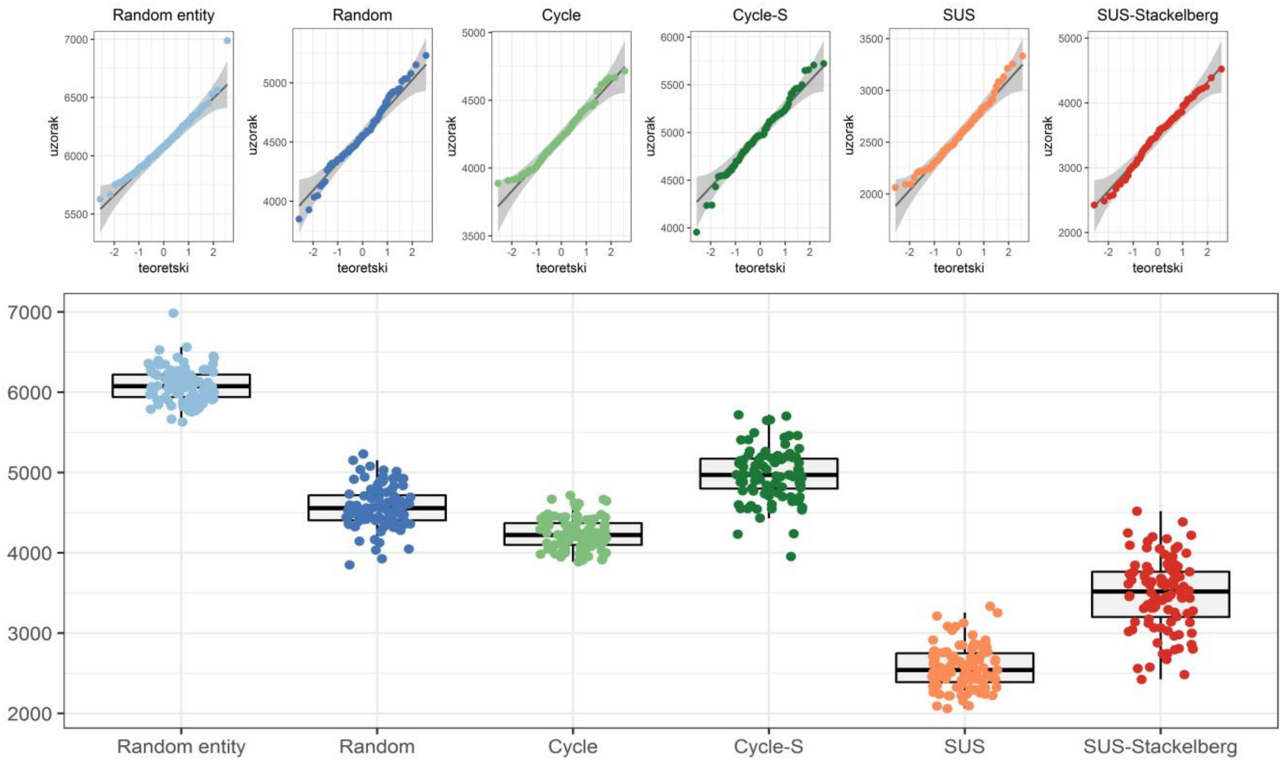

Figure 6.3 – Simulation results for the IT-B scenario, for 6 selected inspection strategies; the top figures display QQ charts – the shaded area is 95% confidence interval; the lower figure displays the results of 100 simulations and corresponding BP diagrams for 6 applied inspection strategies; the horizontal shift (jitter) of the individual results is only for visualization purposes

The strategy of random selection of entities (*Random entity*) again delivers the worst results, since it achieves the highest number of violations of the rules. Inspections in which the *SUS* strategy has been applied are again more successful than inspections in which random and cyclical strategies have been applied. Simulations in which the *SUS-Stackelberg* strategy was applied resulted in a higher number of violations than simulations in which the *SUS* strategy was applied. Additionally, simulations in which the *SUS-Stackelberg* inspection strategy was applied have a relatively large variance and the total number of violations varies from 2,400 to over 4,500, analogously to the situation in Chapter 6.3.

The results of the applied inspection strategies do not deviate from the normal distribution (as can be observed on the QQ graphs), although *Random* and *Cycle* strategies have borderline results. In order to provide a direct comparability with the results of previous tests, the Kruskal-Wallis H test was performed in this scenario as well.

The results of the Kruskal-Wallis H test show that there are statistically significant differences between the groups ($\chi^2 = 551.56$; $df = 5$; $p < 2.2 \cdot 10^{-16}$). The *post-hoc* analysis using the Nemenyi test, with χ^2 approximation, gives the results shown below (Table 6.3).

Table 6.3 – Comparison of the pairs of simulation result for the IT-B scenario applying the Nemenyi test, with χ^2 approximation, for independent samples

	Cycle	Cycle-S	Random	Random entity	SUS
Cycle-S	$3.60 \cdot 10^{-9}$	-	-	-	-
Random	0.03008	0.04042	-	-	-
Random entity	$< 2 \cdot 10^{-16}$	0.00033	$3.50 \cdot 10^{-13}$	-	-
SUS	$4.30 \cdot 10^{-14}$	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$	$< 2 \cdot 10^{-16}$	-
SUS-Stackelberg	0.00115	$< 2 \cdot 10^{-16}$	$1.70 \cdot 10^{-12}$	$< 2 \cdot 10^{-16}$	0.00734

The results confirm the statistical significance of the insight that application of *SUS* strategy achieves fewer violations than the application of *Random*, *Random entity*, *Cycle* or *Cycle-S* strategies. Furthermore, since it was already evident from the analysis of the visual presentation that *SUS-Stackelberg* strategy does not achieve fewer violations than the application of the *SUS* strategy, the statistical significance of the results in the bottom-right cell of the Table 6.3 is not relevant. However, the *SUS-Stackelberg* strategy achieves a statistically significant lower number of violations than the application of *Random*, *Random entity*, *Cycle* or *Cycle-S* strategies.

7. Discussion

This chapter provides an overview of the research findings and associates them with the observations and conclusions from the literature. The limitations of the research and possible directions for further research are also discussed. Finally, the implications of the conducted research on the practical implementation and on evaluation of compliance inspection strategies are discussed.

The first goal of the research was to develop a multi-agent model of centrally-coordinated compliance inspection in a system with multiple supervised organizations (obligees), each of which must comply with multiple provisions (rules). The model should will be applicable to different inspection environments. The developed model was calibrated, verified and its validity was assessed based on the data on compliance inspections with the environmental regulations, banking supervision and compliance inspections with the occupational safety regulations.

General validity tests of the ICARUS model (Chapter 5.2.1) show that the model can successfully recreate the observations and conclusions identified in the analysed empirical research (discussed at length in Chapter 3.2). Empirical macro-observations that an increase in penalties [30][49][97][50][45] and an increase in the level of inspections [30][44][36][120][45] lead to a decrease in the number of violations have been confirmed. When penalties for non-compliance are very low, conducting inspections has a very low impact on compliance [47][45]. The higher the cost of compliance, the higher the level of non-compliance [41][37][43]. The research also confirmed some empirical and theoretical observations about the behaviour of inspected entities (micro-observations): after inspections [45][36] and after punishment [30], entity's level of compliance increases. When entity's perception about the probability of punishment increases [45] and when entity's perception about the probability of inspection increases [120], the level of compliance of that entity with the relevant rules also increases. Furthermore, the general validity tests have shown that prolonging the time since the last inspection of an entity leads to an increase in the level of violations by that entity [47][32].

In line with these observations, it can be concluded that the developed model performs in accordance with the predictions of the economic model of deterrence [32][98, p. 236] and the assumption that entities are when making decisions related to compliance. On the other hand, the developed model "behaves" contrary to the analytical conclusions of Tsebelis' inspection game [51][99][100], which predicts that the amount (level) of the penalty will not affect the level of violations and that more penalties will lead to fewer inspections. Such a result was expected, given that ICARUS does not rely on the initial assumptions of the Tesebelis' model (assumption that entities and inspectors possess complete information, assumption that inspectors are guided by their individual selfish interests, etc.), but on the assumption of bounded rationality of entities.

From all the above, it is evident that the results of general validity assessment of the model are in line with the expected results and the theoretical framework on which the model was based.

The sensitivity analysis of the ICARUS model showed that the total number of violations in the system depends on the input parameters and indicated possible areas of focus for the real-life inspection strategies. Firstly, it showed that inspection capacity has the most significant impact on the total number of violations. That is, the largest reduction in the number of violations is achieved by increasing the capacity of the inspection agency. In addition to this parameter, the total number of violations is significantly affected by the penalty levels (higher penalty leads to fewer violations) and the underlying propensity of the entity to take risks (risk-taking leads to more violations). The importance and differences in the influence of these three input parameters on the model results are in agreement with the results of general validation. Namely, the general validation evaluated, among other things, the relationship between the increase in penalty and number of violations (Chapter 5.2.1.1), the increase in inspections and number of violations (Chapter 5.2.1.2) and the increase in risk appetite and number of violations (Chapter 5.2.1.5). As an example, validity assessment of the random selection strategy shows that a 5-fold increase in inspection capacity led to an almost 10-fold decrease in the number of violations. An increase in punishment of approximately 150% led to a decrease in the number of violations by slightly more than 50%. Finally, an increase in risk appetite of 200% led to an increase in the number of violations of over 100%. Summarily, the results of the general validity assessment and of the sensitivity analysis are aligned, taking into account the direction and magnitude of the impact.

The majority of the remaining input parameters have, as expected, a much lower impact on the total number of violations. For example, the parameters `max-deviation-resources` and `max-risk-attitude-deviation`, according to the model description in Chapter 4.3.2, define ranges of possible deviations of the values of internal variables of individual entities from the globally defined values. Given that values of these parameters affect the level of scattering, i.e. deviations of entity's characteristics in both directions, it was expected that these differences will partly cancel each other out. I.e., that globally (total number of violations in the system) there will be no significant differences in the total number of violations regardless of changes in values of these parameters. The results of the analyses and test are mostly in line with that expectation. It should be noted that the impact of the initial risk appetite on the model results is lower for the random entity selection strategy than for other inspection strategies. The reason is the smaller total number of inspections in the implementation of this strategy.

On the other hand and somewhat surprisingly, the accuracy of inspections and the discount rate also do not have a particularly significant impact on the total number of violations. Given the unexpectedness of these results – especially for the inspection's accuracy parameter – they should be assessed further and verified, primarily through appropriate empirical research.

The influence of the number of rules on the simulation results is more pronounced only when the strategy of random selection of entities is applied. The reason for that can be found in the fact that inspections are relatively rare when this inspection strategy is applied and entities that have not been inspected for a long time are more likely to break the rules. A larger number of rules also means a larger number of rules that will not be covered by inspection. Given the association between the time elapsed from the last inspection and the number of violations – which was confirmed by a general validation test (Chapter 5.2.1.6) – the consequence is a larger number of violations.

The fact that some model parameters have a low impact on the total number of violations (which is the main measure for the analysis and evaluation of the model) indicates the opportunity for developing a simplified compliance inspection model that would not include low impact parameters but would be valid.

The specific validation of the ICARUS model (Chapter 5.2.2) displayed certain similarities, but also some differences in the estimated input parameters for the three considered inspection scenarios (environments). These differences, if they occur in reality and are not just artefacts of the model, could be related to the specifics of the business area in which the entities operate, regulatory environment, cultural characteristics, corporate governance, etc.

The specific validation results show significant differences in the (pre-set and explored) input parameters of the DK-E scenario on the one hand, and the US-W and IT-B scenarios on the other, which lead to larger differences in the results of the simulations. Namely, the DK-E scenario is characterized by a higher inspection capacity, a lower average risk appetite of the entities and a lower penalty (in relation to the compliance costs) compared to the remaining two scenarios. For example, although entities in all three scenarios are, on average, prone to risk taking, the differences in that propensity are significant. Thus, entities in the US-W scenario are – on average – almost three and a half times more likely to take risks than entities in the DK-E scenario. Possible deviations of an individual entity's risk appetite from the average risk appetite are large in all scenarios (82% – 96%). A significantly higher average risk appetite in the US-W scenario compared to the DK-E scenario can also be linked to empirical indicators that the specific validation test attempted to recreate. Namely, although these parameters are not directly comparable, it can be seen that in the DK-E scenario the most frequently violated rule was violated in only 19.5% of cases (steps), while in the US-W scenario as many as 85% of entities did not comply with one or more rules at the time of the first inspection.

Given that inspection capacity, average risk appetite, and penalties have a large impact on the total number of violations, the hypotheses testing found a much lower total number of violations in the DK-E scenario compared to the US-W and IT-B scenarios. This difference can be explained by the following. As already stated, the default risk appetite as well as the maximum deviations of risk appetites are the smallest for the DK-E scenario. Due to that combination of input parameters, entities are more reluctant to take risks and break rules in the DK-E scenario, as compared to the US-W and IT-B scenarios. On the other hand, the inspection capacity in the DK-E scenario is 0.33, which is more than 60% higher than the inspection capacity in the US-W and IT-B scenarios. The influence of these parameters (risk appetite and inspection capacity) on the model results is large, as was confirmed by the sensitivity analysis.

Significant differences are also visible in the accuracy of inspections. In the US-W scenario, only 3% of inspections err in detecting non-compliance, while in the IT-B scenario, over 50% of inspections err in detecting non-compliance. The underlying reason could be in different characteristics of the relevant regulations. Namely, the workplace safety rules are relatively exact (the so-called rule-based regulations [101]) and it is possible to assume that compliance or non-compliance can be determined with comparatively high accuracy. On the other hand, banking regulation is often cited as an example of principle-based regulation [101]. Compliance or non-compliance with the principles is more difficult to assess and differences in interpretation are possible. Related to this, it is possible to assume that in such an environment the inspected entity can sometimes convince the inspector of compliance even though in reality it is non-compliant. On the other hand, the discount rate in all three scenarios is very low, indicating that entities have great recollection.

Comparison of the results of simulations that were conducted with the aim of hypotheses testing (Chapter 6) shows significant variations in the overall level of violations, depending on the analysed scenario and the applied inspection strategy. The results of the hypotheses tests have shown that the application of the Stochastic universal sampling inspection strategy (*SUS*) in all considered scenarios resulted with a significantly lower number of violations than the application of random or cyclical strategies. However, the application of the inspector's leadership (*SUS-Stackelberg* inspection strategy) led to significantly better results than the application of the *SUS* strategy only in the DK-E scenario. That is, the application of the *SUS-Stackelberg* inspection strategy in the DK-E scenario achieved by far the lowest number of violations (when compared to simulations with other inspection strategies), while in the US-W and IT-B scenarios the *SUS-Stackelberg* strategy was significantly inferior to the *SUS* strategy. These results support the following conclusion: **Hypotheses H.1 and H.2 are accepted and hypothesis H.3 is rejected.**

There implications of the results of specific validation tests and hypotheses tests are numerous. As already mentioned, it is possible that DK-E on the one hand, and US-W and IT-B on the other, represent two types of environments. In the first type of environment (DK-E), the optimal strategy of the inspection agency is to conduct inspections in line with the assessed compliance costs, while announcing, in a credible manner, to all the obligees that such inspection strategy will be implemented in practice. In the second type of environment (US-W and IT-B), the optimal strategy of the inspection agency is to conduct inspections in line with the costs of achieving and maintaining compliance with the provisions, but to hide from the entities information about the implemented inspection strategy. Of course, it is debatable how realistic the possibility of long-term suppression of information about the inspection strategy is, especially if the possibility of communication between the entities is introduced into the model.

The results of the research have direct implications for the inspection practice and can be applied to the adaptation of existing and development of new inspection strategies, which is also the societal contribution of this research. The first possible change in inspection practice is to increase inspection capacity or penalties for non-compliance, which should lead to a reduction in the total number of violations. But while increasing inspection capacity would be an obvious improvement (if the goal is to reduce the total number of violations), it is almost always directly related to an increase in the budgets of inspection agencies, which often makes such changes difficult or politically unfeasible. However, the simulation results also show that the accuracy of inspections in identifying violations has a relatively low impact on the total number of violations. This result, combined with the knowledge of sensitivity of results to inspection capacity, suggests that inspection agencies could try to control the costs of their operations by balancing the increase in inspection capacity with a shortening of individual inspections, reduction in scope and exhaustiveness of inspection procedures, reduction in inspectors' cost, etc. E.g., it could be opportune to increase the number and/or scope of inspections at the expense of their quality by conducting inspections that are less in-depth or by hiring less experienced inspectors who would – presumably – entail lower costs. On the other hand, the total number of violations could be affected indirectly. For example, since legislators or supervisors cannot directly change an entity's risk appetite, they may try to influence it by limiting the ways in which employees and owners are rewarded (for example, through remuneration regulations and policies), by changing the licensing procedures for the management and supervisory boards, etc. It is prudent to not that some of the possible courses of action are controversial or go against the accepted practices and common wisdom. Therefore, further research and validation of these recommendations is needed.

The results of hypotheses testing further indicate that it might be opportune to adjust inspection practices by taking into account information on the costs of compliance with the relevant regulations and particular provisions of those regulations, in order to reduce the overall number of violations. In addition, regulators and inspection agencies could try to gain additional information about the environment and the risk appetite of the obligees. If regulators and inspectors conclude that regulated entities have a low propensity for risk-taking and inspection agencies have a relatively high inspection capacity, the inspection agency might consider to credibly inform obligees that inspections will be planned and conducted in accordance with the assessment of costs of compliance with the regulations. Such strategy might reduce the total level of non-compliance in the system.

A further societal contribution of this research is the possibility for other researchers and general public without specialist knowledge of the theoretical basis of the inspection game, agent-based modelling or computer simulations in general to use the developed computer simulation of the ICARUS model. Visualization of the simulation and presentation of quantitative results of the simulation as well as graphical presentation of basic statistical indicators facilitate the understanding of the simulation and its results. These characteristics of the simulation might facilitate the use and adaptation of the model and simulation to other inspection problems. The detailed description included in this paper should allow researchers to modify the model or to implement it relatively easily in other programming languages and simulation environments.

Several limitations of this research have been identified, and present opportunities for further research. Firstly, the developed model – like any model – presents only a simplified version of reality. Therefore, the assumptions of the model as well as the decision-making mechanisms and the behaviour of the entities have been simplified. An obvious example is the propensity to take risks, which is modelled as an individual, time-stable characteristic of an entity. However, in reality the risk appetite may change, based on the experience of the management and the organization itself, changes in the organization's management or internal/external control functions, changes in corporate culture or business environment, etc. Furthermore, although the model is calibrated, verified and validated, these procedures were performed in relation to a limited set of secondary data on compliance inspections. In addition, the validation did not consider whether the model was quantitatively consistent with empirical micro-structures, primarily due to a lack of data necessary for this type of validation.

Further research could extend the model by allowing changes in entity's risk appetite over time, introducing additional learning strategies, enabling communication between entities, introducing a general deterrence mechanism, dividing entities into several groups to which somewhat different inspection strategies are applied, introducing additional inspection strategies that combine elements of more basic inspection strategies, etc. On the other hand, the model could be more focused on a specific area, i.e. it could be adapted to be valid at a higher level – to be quantitatively harmonized with empirical micro-structures in a particular inspection area (the prerequisite for such change is availability of relevant data).

On the other hand, based on the results of the model validation and sensitivity analysis, a new, simpler model of compliance inspection could be constructed, in which parameters with low impact on the total number of violations would be removed. Simpler models are desirable since they allow easier and more robust interpretation of results, assessment of correlation of parameters, and understanding of mechanisms with a key impact on the results. Such a model could be applicable to an even wider range of inspection problems and would be intuitively more understandable.

The model could also be significantly upgraded by introducing an additional step of the game and a new type of agent. Namely, inspection environments are often characterized by an additional step after the inspection itself – legal proceedings. In such environments, supervised entity might have the option to appeal the findings through a judicial process or the judiciary always process the findings of inspections and decides on the justification and level of the penalty. The introduction of these upgrades would significantly change the relationships between agents and could considerably affect their actions and model results.

8. Conclusion

This paper describes ICARUS – a multi-agent model in which an inspection agency centrally coordinates inspections of compliance of a number of entities with a number of rules. The model is based on insights from theoretical research and empirical data and is implemented in a computer simulation in the programming language NetLogo. The paper contains exhaustive description of the model and its implementation in the computer simulation. The model was verified and then validated in two steps. In the first step, the general validity of the model was assessed, i.e. the extent to which the model was qualitatively harmonized with empirical observations at the macro and micro level. In the second step, the specific validity of the model was assessed, i.e. the extent to which the model was quantitatively consistent with the empirical macro-structures, for three examined case studies. The initial estimation/calibration of the input parameters was performed based on the secondary empirical data, and the narrowing of the ranges was performed as part of the specific validity assessment. After validation, sensitivity analysis of the model was performed via a screening method.

Agent-based modelling was shown to be an appropriate method for modelling this inspection problem. It enabled the application of insights about the behaviour of individual agents (micro level) in the development of the model and the analysis of model results at the macro and micro level. Furthermore, agent-based modelling enabled straightforward sophistication of the model, i.e. the introduction of more realistic assumptions into the basic model.

Calibration/assessment of the model parameters, verification and validation have shown that the developed model successfully recreates the regularities and patterns of behaviour identified in theoretical research and empirical data on inspections of compliance with environmental, occupational safety and health regulations as well as banking supervision.

The hypotheses tests performed on the simulation results confirm that by implementing inspection strategy in which the selection of provisions for compliance inspection is performed relative to the costs of compliance with those provisions, the simulation results in a lower level of noncompliance, compared to the use of random selection strategies (H.1) and the cyclical strategy (H.2). The hypothesis that by applying the inspector's (Stackelberg) leadership to inspections in which the selection of provisions for compliance inspection is performed relatively to the costs of compliance with those provisions, the simulation results in a lower level of noncompliance, with respect to the inspection without the inspector's leadership was rejected (H.3).

The research supports the basic assumptions of the economic model of deterrence about the (bounded) rationality of the entities when they make decisions on compliance with regulations. The research has shown that the total number of violations in the system mostly depends on three parameters (decreasing in their impact): inspection capacity (higher inspection capacity leads to less violations), penalty level (higher penalty leads to less violations) and entity's risk appetite (higher risk appetite leads to more violations). The implications of these insights on inspection practice are clear: if inspectors, supervisors and regulators want to reduce the level of non-compliance in the system, it is necessary to increase the inspection capacity, to increase penalties for non-compliance and to try to influence the entities' risk appetites. These insights, combined with the insights that the accuracy of inspections in finding violations and time discounting have a rather low impact on the total number of violations, could facilitate the design of adjustments to existing or the development of completely new inspection strategies. These findings and the possibility of their application in practice also represent a key societal contribution of this research.

Further societal and scientific contribution of the research stems from the scope of application and ease of use of the model and the implemented computer simulation. The developed model and its implementation in the computer simulation are flexible and allow application of the model to inspection problems beyond those analysed and discussed in this paper. The computer simulation is documented in details. A user-friendly graphical user interface for setting up the simulation and monitoring its progress and results enables the use of the model by a wide range of researchers from different disciplines and other users who do not have insights into the theoretical framework of the inspection problem or experience in agent-based modelling or computer simulations.

This paper has also identified the limitations of the conducted research and – related to that – the areas of possible further investigation. Further research could be – on one hand – focused on development of simpler inspection models that could be more widely applicable and intuitive, and – the other hand – on more complex models that could focus on a specific area and/or be better aligned with concrete, real-world situations.

Bibliography

[1] R. Baldwin, M. Cave, and M. Lodge, *Understanding regulation: theory, strategy, and practice*. Oxford University Press on Demand, 2012.

[2] S. S. Simpson, *Corporate Crime, Law, and Social Control*. 2005.

[3] M. L. Benson, 'White-Collar Crime', *21st Century Criminology A Reference Handbook*. SAGE Publications, Inc., pp. 549–558, 2009.

[4] G. Caprio Jr., 'Safe and Sound Banking: A Role for Countercyclical Regulatory Requirements?', Policy Research Working Paper 5198, 2010.

[5] Š. Anić and N. Klaić, *Rječnik stranih riječi*. SA NI-PLUS d.o.o. , 1998.

[6] R. Avenhaus and M. J. Canty, 'Inspection games', in *Computational Complexity: Theory, Techniques, and Applications*, 2012, pp. 1605–1618.

[7] J. Slemrod, 'Cheating ourselves: The economics of tax evasion', *The Journal of Economic Perspectives*, vol. 21, no. 1, pp. 25–48, 2007.

[8] R. Fisman and E. Miguel, 'Corruption, Norms, and Legal Enforcement: Evidence from Diplomatic Parking Tickets', *Journal of Political Economy*, vol. 115, no. 6, pp. 1020–1048, 2007.

[9] S. Smojver, 'Inspecting compliance to many rules: An agent-based model', *Interdisciplinary Description of Complex Systems*, vol. 14, no. 3, pp. 277–295, 2016.

[10] G. S. Becker, *Crime and Punishment : An Economic Approach*, vol. I, no. 2. 1974.

[11] H. Rauhut and M. Junker, 'Punishment Deters Crime Because Humans Are Bounded in Their Strategic Decision-Making', *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 3, p. 1, 2009.

[12] H. Rauhut, 'Stronger Inspection Incentives, Less Crime? Further Experimental Evidence on Inspection Games', *Rationality and Society*, vol. 27, no. 4, pp. 414–454, 2015.

[13] H. Rauhut and S. Jud, 'Avoiding Detection or Reciprocating Norm Violations? An Experimental Comparison of Self- and Other-Regarding Mechanisms for Norm Adherence', *Soziale Welt-Zeitschrift für Socialwissenschaftliche Forschung und Praxis*, vol. 65, no. 2, pp. 153–183, 2014.

[14] K. M. Bloomquist, 'A Comparison of Agent-Based Models of Income Tax Evasion', *Social Science Computer Review*, vol. 24, pp. 411–425, 2006.

[15] G. Zaklan, F. Westerhoff, and D. Stauffer, 'Analysing Tax Evasion Dynamics via the Ising Model', *Journal of Economic Interaction and Coordination*, pp. 1–14, 2009.

[16] P. Pellizzari and D. Rizzi, 'Citizenship and Power in An Agent-Based Model of Tax Compliance With Public Expenditure', *Journal of Economic Psychology*, vol. 40, pp. 35–48, 2014.

[17] K. M. Bloomquist, 'Agent-Based Simulation of Tax Reporting Compliance', 2012.

[18] A. L. Andrei, K. Comer, and M. Koehler, 'An Agent-Based Model of Network Effects on Tax Compliance and Evasion', *Journal of Economic Psychology*, vol. 40, pp. 119–133, 2014.

[19] L. Antunes, J. Balsa, and H. Coelho, 'Agents that Collude to Evade Taxes', in *The 6. Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 07)*, 2007, pp. 1255–1257.

[20] K. M. Bloomquist and M. Koehler, 'A Large-Scale Agent-Based Model of Taxpayer Reporting Compliance', *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 2, p. 20, 2015.

[21] E. van Asselt, S. Osinga, and H. Bremmers, 'Simulating Compliance Behaviour for Effective Inspection Strategies Using Agent Based Modelling', *British Food Journal*, vol. 118, no. 4, pp. 809–823, 2016.

[22] T. Verwaart and N. I. Valeeva, 'An Agent-based Model of Food Safety Practices Adoption', in *Emergent Results of Artificial Economic*, 2011, pp. 103–114.

[23] S. McPhee-Knowles, 'Growing Food Safety from the Bottom Up: An Agent-Based Model of Food Safety Inspections', *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 2, p. 9, 2015.

[24] Y. Liu and H. Ye, 'The Dynamic Study on Firm's Environmental Behavior and Influencing Factors: An Adaptive Agent-Based Modeling Approach', *Journal of Cleaner Production*, vol. 37, pp. 278–287, 2012.

[25] N. Malleson, 'Using Agent-Based Models to Simulate Crime', in *Agent-based models of geographical systems*, Springer, 2012, pp. 411–434.

[26] N. Malleson, A. Heppenstall, and L. See, 'Crime Reduction Through Simulation: An Agent-Based Model of Burglary', *Computers, Environment and Urban Systems*, vol. 34, no. 3, pp. 236–250, 2010.

[27] S. Smojver, 'Inspecting Compliance to Many Rules: an Agent Based Model', *Interdisciplinary Description of Complex Systems*, vol. 14, no. 3, pp. 277–295, 2016.

[28] S. Smojver, 'Analysis of Banking Supervision via Inspection Game and Agent-Based Modeling', *Central European Conference on Information and Intelligent Systems*, 2012.

[29] M. Brogi, 'Once Bitten Twice Shy? A Study on the Effectiveness of Administrative Sanctions to Discipline Bank Board Members', *CAREFIN Centre for Applied Research in Finance Working Paper*, pp. 1–32, 2011.

[30] W. B. Gray and J. P. Shimshack, 'The Effectiveness of Environmental Monitoring and Enforcement: A Review of the Empirical Evidence', *Review of Environmental Economics and Policy*, vol. 5, no. 1, pp. 3–24, 2011.

[31] European Banking Authority, *Guidelines on Common Procedures and Methodologies for SREP*. London, 2014.

[32] K. Ko, J. Mendeloff, and W. Gray, 'The role of inspection sequence in compliance with the US Occupational Safety and Health Administration's (OSHA) standards: Interpretations and implications', *Regulation and Governance*, vol. 4, no. 1, pp. 48–70, 2010.

[33] E. Helland, 'The Enforcement of Pollution Control Laws: Inspections, Violations, and Self-Reporting', *Review of Economics and Statistics*, vol. 80, no. 1, pp. 141–153, 1998.

[34] M. D. Delis and P. K. Staikouras, 'Supervisory Effectiveness and Bank Risk', *Review of Finance*, vol. 15, no. 3, pp. 511–543, Mar. 2011.

[35] R. Avenhaus, B. von Stengel, and S. Zamir, 'Inspection games', *Handbook of Game Theory with Economic Applications*, vol. 3, pp. 1947–1987, 2001.

[36] J. P. Shimshack, 'The Economics of Environmental Monitoring and Enforcement', *Annual Review of Resource Economics*, vol. 6, no. 1, pp. 339–360, 2014.

[37] S. C. Winter and P. J. May, 'Motivation for compliance with environmental regulations', *Journal of Policy Analysis and Management*, vol. 20, no. 4, pp. 675–698, 2001.

[38] Hrvatska narodna banka, *Odluka o sadržaju revizije u kreditnim institucijama*. Narodne novine.

[39] Hrvatska narodna banka, *Odluka o sustavu unutarnjih kontrola*. Narodne novine, p. 1/2015.

[40] J. R. Barth, G. J. Caprio, and R. Levine, 'Bank Regulation and Supervision in 180 Countries from 1999 to 2011', *Journal of Financial Economic Policy*, vol. 5, no. 2, pp. 111–219, 2013.

[41] P. J. May and S. C. Winter, 'Regulatory Enforcement and Compliance: Examining Danish Agro-Environmental Policy', *Journal of Policy Analysis and Management*, vol. 18, no. 4, pp. 625–651, 1999.

[42] A. Heyes and N. Rickman, 'Regulatory Dealing – Revisiting the Harrington Paradox', *Journal of Public Economics*, vol. 72, no. 3, pp. 361–378, Jun. 1999.

[43] W. A. Magat and W. K. Viscusi, 'Effectiveness of the EPA's Regulatory Enforcement: The Case of Industrial Effluent Standards', *The Journal of Law and Economics* , vol. 33, no. 2, pp. 331–360, 1990.

[44] H. Corman and N. H. Mocan, 'A Time-Series Analysis of Crime, Deterrence, and Drug Abuse in New York City', *The American Economic Review*, vol. 90, no. 3, pp. 584–604, 2000.

[45] J. Alm and J. P. Shimshack, 'Environmental Enforcement and Compliance: Lessons from Pollution, Safety, and Tax Settings', *Tulane Economics Working Paper Series*, pp. 1–46, 2014.

[46] J. T. Scholz and W. B. Gray, 'OSHA Enforcement and Workplace Injuries: A Behavioral Approach to Risk Assessment', *Journal of Risk and Uncertainty*, vol. 3, pp. 283–305, 1990.

[47] W. B. Gray and J. M. Mendeloff, 'The Declining Effects of OSHA Inspections on Manufacturing Injuries, 1979–1998.', *ILR Review*, vol. 58, no. 4, pp. 571-587., 2005.

[48] D. I. Levine, M. W. Toffel, and M. S. Johnson, 'Randomized Government Safety Inspections Reduce Worker Injuries with No Detectable Job Loss', *Science*, vol. 336, no. 6083, 2012.

[49] J. Grogger, 'Certainty vs. Severity of Punishment', *Economic Inquiry*, vol. 29, no. 2, pp. 297–309, 1991.

[50] C. Almer and T. Goeschl, 'Environmental Crime and Punishment: Empirical Evidence from the German Penal Code', *Land Economics*, vol. 86, no. 4, pp. 707–726, Nov. 2010.

[51] R. Pradiptyo, 'Does Punishment Matter? A Refinement of the Inspection Game', *Review of Law and Economics*, vol. 3, no. 2, pp. 197–219, 2007.

[52] B. Müller *et al.*, 'Describing Human Decisions in Agent-Based Models - ODD+D, An Extension of the ODD Protocol', *Environmental Modelling and Software*, vol. 48, pp. 37–48, 2013.

[53] V. Grimm *et al.*, 'A standard protocol for describing individual-based and agent-based models', *Ecological modelling*, vol. 198, no. 1, pp. 115–126, 2006.

[54] Y. Shoham and K. Leyton-brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[55] J. E. Baker, 'Reducing Bias and Inefficiency in the Selection Algorithm', in *Proceedings of the Second International Conference on Genetic Algorithms*, 1987, pp. 14–21.

[56] C. R. Reeves and J. E. Rowe, *Genetic Algorithms—Principles and Perspectives*, vol. 20. Boston: Kluwer Academic Publishers, 2002.

[57] H. Deguchi, *Agent-Based Modelling of Socio-Technical Systems*, vol. 9. Dordrecht: Springer Netherlands, 2013.

[58] L. Tesfatsion and K. L. Judd, *Handbook of Computational Economics*, vol. 2. 2006.

[59] L. W. Sherman, 'The Power Few: Experimental Criminology and the Reduction of Harm', *Journal of Experimental Criminology*, vol. 3, no. 4, pp. 299–321, Nov. 2007.

[60] U. Wilensky, 'NetLogo', *Center for Connected Learning and Computer-Based Modeling, Northwestern University,* 1999.

[61] H. Bersini, 'UML for ABM', *Journal of Artificial Societies and Social Simulation*, vol. 15, no. 1, p. 9, 2012.

[62] A. Lansdowne, 'Traffic Simulation Using Agent-Based Modelling', Bristol, 2006.

[63] T. R. Browning, 'Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions', *IEEE Transactions On Engineering Management*, vol. 48, no. 3, 2001.

[64] S. Railsback *et al.*, 'Improving Execution Speed of Models Implemented in NetLogo', *Journal of Artificial Societies and Social Simulation*, vol. 20, no. 1, 2017.

[65] K. Kravari and N. Bassiliades, 'A Survey of Agent Platforms', *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 1, 2015.

[66] J. C. Thiele, 'R Marries NetLogo: Introduction to the RNetLogo Package', *Journal of Statistical Software*, vol. 58, no. 2, pp. 1–41, 2014.

[67] U. Wilensky and W. Rand, *An Introduction to Agent-Based Modeling Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. The MIT Press, 2015.

[68] F. Stonedahl and U. Wilensky, 'BehaviorSearch.org', 2017. [Online]. Available: http://www.behaviorsearch.org/about.html. [Accessed: 30-Aug-2017].

[69] M. Richiardi, R. Leombruni, N. J. Saam, and M. Sonnessa, 'A Common Protocol for Agent-Based Social Simulation', *Journal of Artificial Societies and Social Simulation*, vol. 9, no. 1, p. 15, 2006.

[70] J. M. Galán *et al.*, 'Errors and Artefacts in Agent-Based Modelling', *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 1, p. 1, 2009.

[71] P. Windrum, G. Fagiolo, and A. Moneta, 'Empirical Validation of Agent-Based Models: Alternatives and Prospects', *Journal of Artificial Societies and Social Simulation*, vol. 10, no. 2, p. 8, 2007.

[72] Ö. Gürcan, O. Dikenelli, and C. Bernon, 'A Generic Testing Framework for Agent-Based Simulation Models', *Journal of Simulation*, vol. 7, no. 3, pp. 183–201, 2013.

[73] J. C. Thiele, W. Kurth, and V. Grimm, 'Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and "R"', *Journal of Artificial Societies and Social Simulation*, vol. 17, no. 3, p. 11, 2014.

[74] A. Saltelli, S. Tarantola, and F. Campolongo, 'Sensitivity Analysis as an Ingredient of Modeling', *Statistical Science*, vol. 15, no. 4, pp. 377–395, 2000.

[75] A. Saltelli *et al.*, *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, Ltd, 2008.

[76] M. D. Morris, ‘Factorial Sampling Plans for Preliminary Computational Experiments’, *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.

[77] M. A. Louie and K. M. Carley, ‘Balancing the Criticisms: Validating Multi-Agent Models of Social Systems’, *Simulation Modelling Practice and Theory*, vol. 16, no. 2, pp. 242–256, Feb. 2008.

[78] S. Moss, ‘Alternative Approaches to the Empirical Validation of Agent-Based Models’, *Journal of Artificial Societies and Social Simulation*, vol. 11, no. 1, p. 5, 2008.

[79] P. Murè and V. Pesic, ‘Are Sanctions Effective in Improving Bank Performance? A Study on Supervision and Administrative Sanctions Upon Italian Banks During the 1998- 2009 Period’, *CAREFIN Centre for Applied Research in Finance Working Paper*, no. 23, 2010.

[80] J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard, *Metaheuristics for Hard Optimization: Methods and Case Studies*. Berlin Heidelberg: Springer-Verlag, 2005.

[81] R Core Team, ‘R: A Language and Environment for Statistical Computing’. Vienna, Austria, 2017.

[82] RStudio Team, ‘RStudio: Integrated Development Environment for R’. Boston, MA, 2015.

[83] H. Wickham, ‘Package “plyr” - Tools for Splitting, Applying and Combining Data’, 2016. [Online]. Available: http://had.co.nz/plyr. [Accessed: 27-Aug-2017].

[84] J. Fox, ‘Package “polycor” - Polychoric and Polyserial Correlations’, 2016. [Online]. Available: https://cran.r-project.org/package=polycor.

[85] H. Wickham, ‘Package “tidyverse”’, 2017. [Online]. Available: https://cran.r-project.org/web/packages/tidyverse/tidyverse.pdf. [Accessed: 27-Aug-2017].

[86] H. Wickham, ‘Reshaping Data with the {reshape} Package’, *Journal of Statistical Software*, vol. 21, no. 12, pp. 1–20, 2007.

[87] B. Auguie and A. Antonov, ‘Package “gridExtra” - Miscellaneous Functions for “Grid” Graphics’, 2016. [Online]. Available: https://cran.r-project.org/web/packages/gridExtra/gridExtra.pdf. [Accessed: 27-Aug-2017].

[88] B. Ripley, ‘Package “MASS” - Support Functions and Datasets for Venables and Ripley’s MASS’, 2017. [Online]. Available: http://www.stats.ox.ac.uk/pub/MASS4/. [Accessed: 27-Aug-2017].

[89] G. Pujol *et al.*, ‘Package “sensitivity” - Global Sensitivity Analysis of Model Outputs’, 2017. [Online]. Available: https://cran.r-project.org/web/packages/sensitivity/sensitivity.pdf. [Accessed: 27-Aug-2017].

[90] E. Neuwirth, ‘Package “RColorBrewer”’, 2014. [Online]. Available: https://cran.r-project.org/web/packages/RColorBrewer/RColorBrewer.pdf. [Accessed: 27-Aug-2017].

[91] F. J. Miguel Quesada, J. a. Noguera, T. Llacer, and E. Tapia Tejada, ‘Exploring Tax Compliance: An Agent-Based Simulation’, in *ECMS 2012 Proceedings edited by: K. G. Troitzsch, M. Moehring, U. Lotzmann*, 2012, pp. 638–643.

[92] L. R. Izquierdo, S. S. Izquierdo, and F. Vega-Redondo, ‘Learning and Evolutionary Game Theory’, *Encyclopedia of the Sciences of Learning*. Springer US, pp. 1782–1788, 2012.

[93] K. Tuyls and A. Nowe, ‘Evolutionary Game Theory and Multi-agent Reinforcement Learning’, *The Knowledge Engineering Review*, vol. 20, no. 1, pp. 63–90, 2005.

[94] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press, 2003.

[95] G. R. Norman and D. L. Streiner, *PDQ statistics*, 3rd ed., vol. 1. London: BC Decker Inc, 2003.

[96] T. Pohlert, ‘Package “PMCMR” - Calculate Pairwise Multiple Comparisons of Mean Rank Sums’, 2016. [Online]. Available: https://cran.r-project.org/web/packages/PMCMR/PMCMR.pdf. [Accessed: 27-Aug-2017].

[97] M. D. Delis and P. K. Staikouras, ‘On-Site Audits, Sanctions, and Bank Risk-Taking: An Empirical Overture Towards a Novel Regulatory and Supervisory Philosophy’, *MPRA Paper No. 16836*, 2009.

[98] R. Paternoster, ‘Deterrence And Rational Choice Theories’, *21st Century Criminology A Reference Handbook*. SAGE Publications, Inc., pp. 236–244, 2009.

[99] G. Tsebelis, ‘Penalty Has No Impact on Crime: A Game-theoretic Analysis’, *Rationality and Society*, vol. 2, no. 3, pp. 255–286, 1990.

[100] G. Tsebelis, ‘Are Sanctions Effective?: A Game-Theoretic Analysis’, *Journal of Conflict Resolution*, vol. 34, no.

1. pp. 3–28, 1990.

[101] J. Black, 'The Rise, Fall and Fate of Principles Based Regulation', in *Law Reform and Financial Markets*, Elgar Fina., K. Alexander and N. Moloney, Eds. 2010, pp. 3–34.

[102] R. Hohzaki, J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith, 'Inspection Games', in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley & Sons, Inc., 2011.

[103] V. Kolokoltsov, H. Passi, and W. Yang, 'Inspection and Crime Prevention: An Evolutionary Perspective', *University of Strathclyde Working Paper*, pp. 1–33, Jun. 2013.

[104] S. Walklate, *Understanding Criminology Current Theoretical Debates*, 3rd editio. Open University Press, 2007.

[105] N. Feltovich, 'Belief-Based Learning Models', *Encyclopedia of the Sciences of Learning*. Springer US, pp. 444–447, 2012.

[106] T. Brenner, 'Agent Learning Representation: Advice on Modelling Economic Learning', *Handbook of Computational Economics*, vol. 2, pp. 895–947, 2006.

[107] T. S. Critchfield and S. H. Kollins, 'Temporal Discounting: Basic Research and the Analysis od Socially Important Behavior', *Journal Of Applied Behavior Analysis*, vol. 34, no. 1, pp. 101–122, 2001.

[108] D. E. Knuth, 'Two notes on notation', *The American Mathematical Monthly*, vol. 99, no. 5, pp. 403–422, 1992.