ATLANTIS PRESS

International Journal of Computational Intelligence Systems
In Press, Uncorrected Proof
DOI: https://doi.org/10.2991/ijcis.d.191101.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

International Journal of Computational Intelligence Systems

II-Learn—A Novel Metric for Measuring the Intelligence Increase and Evolution of Artificial Learning Systems

László Barna Iantovics[1], Dimitris K. Iakovidis[2], Elena Nechita[3,*]

[1] *Department of Informatics, University of Medicine, Pharmacy, Sciences and Technology of Targu Mures, Gh. Marinescu 38, Targu Mures, Mures 540139, Romania*
[2] *Department of Computer Science and Biomedical Informatics, University of Thessaly, Papasiopoulou 2-4, Lamia, 35131, Greece*
[3] *Department of Mathematics and Informatics, Vasile Alecsandri University of Bacau, Calea Marasesti 157, Bacau, 600115, Romania*

ARTICLE INFO

Article History

Received 23 Aug 2018
Accepted 16 Oct 2019

Keywords

Machine learning
Machine intelligence
Intelligent system
Evolving system
Cooperative multiagent system
Machine intelligence measure

ABSTRACT

A novel accurate and robust metric called II-Learn for measuring the increase of intelligence of a system after a learning process is proposed. We define evolving learning systems, as systems that are able to make at least one measurable evolutionary step by learning. To prove the effectiveness of the metric we performed a case study, using a learning system. The universality of II-Learn is based on the fact that it does not depend on the architecture of the studied system.

1. INTRODUCTION

Intelligent systems have been successfully applied for the solution of a variety of difficult practical problems, such as medical diagnosis [1–6], intrusion detection [7], network traffic anomalies detection [8], multisensor battlefield reconnaissance simulation [9], local semantic indexing for resource discovery on overlay networks [10], distributed reasoning for context-aware service [11] and real-time water demand management [12]. Given a problem, its solving difficulty can be considered from different viewpoints, for example, from the human and the computational viewpoint. In this paper, we focus on difficult problems from the computational viewpoint, such as NP-hard problems.

Many intelligent systems are agent-based (ABSs) [13–20], consisting of intelligent agents (IAs) and intelligent cooperative multiagent systems (intelligent CMASs). There is no unanimous definition of intelligent agent-based systems (IABSs) [13,14,21–23]. Frequently, the characterization of a system as intelligent is based on its ability to learn autonomously a specific task. Learning can result in the modification of existing knowledge, or even in the construction or discovery of new knowledge.

The appreciation/definition of intelligence of a system is usually based on bio-inspired considerations [13–15,24–26] including autonomous learning, self-adaptation, and evolution. A variety of machine-learning approaches for solving practical problems have been proposed [27–38] varying from simple to complex techniques. *Rote learning* is an example of simple learning. For humans, rote learning consists in memorization based on repetition, while many early expert systems can be considered as rote learners since knowledge is retained in the form transmitted by a human supervisor without any modification. For example, MYCIN [39] could be considered as such a system capable of solving problems similarly with the human specialists in the context of decision support system for infectious disease management. More complex learning examples may include reasoning approaches, such as induction and abduction.

Must be mentioned that if a system learns, it does not necessarily mean that it has become more intelligent, in the sense that it has a measurable increase in the intelligence level. Learning in some cases could result even in a decrease of intelligence, for example, a system could learn misleading data or overfit on the training data thus losing its ability to generalize. Furthermore, we consider that the intelligence measure modification (increment or decrement in some cases), does not directly relate to the learning complexity. Even a very simple form of autonomous learning, such as rote learning, can result in a significant increase (e.g., of the performance) and sometimes even of the intelligence [40].

The main contribution of this paper is the introduction of a novel universal mathematically grounded metric, called *II-Learn* (*metric*

[*] *Corresponding author. Email: enechita@ub.ro*

for measuring the Intelligence Increase of artificial Learning systems), that is able to quantify the Central Intelligence Tendency (CIT) and provide additional characterization of a system's intelligence. The calculation of *II-Learn* metric considers the CIT of the intelligent system before and after the learning process and makes an accurate comparison by verifying if the two measured central intelligence tendencies are different from the statistical viewpoint, taking into consideration the variability in the practical problem-solving intelligence. The proposed approach provides both enhanced robustness, accuracy and universality over most current approaches.

In a previous study the universal and robust *MetrIntComp* metric [41] for the measurement of system's intelligence was proposed. Since both metrics *II-Learn* and *MetrIntComp* quantify the problem-solving intelligence of systems, they can be considered comparable. The main advantage of *II-Learn* over the *MetrIntComp* consists in its higher accuracy. *II-Learn* conserve the robustness property of the *MetrIntComp* metric by making some kind of experimental problem-solving intelligence evaluation data transformation. *II-Learn* consider the presence of possible extremes (measured problem-solving intelligence values that are very different from the other measured problem-solving intelligence values) and it applies a methodology to remove them.

Furthermore, we introduce definitions for a variety of concepts, which include the "Evolutionary Step" in the life cycle of an intelligent learning system, the "Involutionary Step" in the life cycle of an intelligent learning system, and respective system definitions that include the "Intelligent Evolving Learning System" and the "Intelligent Involving Learning System." *II-Learn* metric enables the verification if a studied system attained an *evolution in intelligence* by learning, in the sense of intelligence increase (such systems are referred as Evolving Intelligent Learning Systems) or intelligence decrease, called *involution in intelligence* (such systems are referred as Involving Intelligent Learning Systems). In nature the most frequent situation of what we call "involution" is when a species move into another environment or during a longer time evolution stops using some senses/structures for surviving; however, these senses/structures may decrease or even disappear. For example, we mention the wings of ostriches as they are remnants of their flying ancestors' wings. *II-Learn* enables accurate measurements, even in the case of a small modification of a system's intelligence, and it takes into consideration the variability in the problem-solving intelligence level. Thus it can be used in the context of a cooperative multiagent system that could manifest different levels of intelligence for different problems undertaken for solving.

To demonstrate the effectiveness of *II-Learn* we performed an experimental case study, using a cooperative multiagent system specialized in solving the *Travelling Salesman Problem* (TSP) [42,43] that is a well-known NP-hard problem. We investigate whether a simple rote learning of a system that exhibits a behavioral adaptation by learning results in an intelligence increase, and in the affirmative case if an evolutionary step in the intelligence is performed.

The rest of this paper is organized in 6 sections. Section 2 discusses the motivation of this work with respect to open issues identified in literature; presents the state-of-the-art metrics/methods proposed for measuring machine intelligence. In Section 3 the proposed *II-Learn* metric is presented. Section 4 presents an experimental study that demonstrates and validates *II-Learn* metric. Section 5 provides a discussion about the proposed metric, and finally, Section 6 summarizes the main conclusions of the presented research.

2. STATE-OF-THE-ART AND MOTIVATION

ABSs frequently enable intelligent solving of a large diversity of real-life difficult problems, like, optimization of coordination of human groups online [44], and collaborative multisensor agents for multitarget tracking and surveillance [45]. Although several studies address intelligent problem-solving using ABSs, the evidence provided for their intelligence is usually intuitive, and there are several still open issues with respect to measures used for the assessment of their learning ability.

2.1. Evidence of System Intelligence

In the case of cooperative multiagent systems generally, both learning and intelligence measurement could be even more complex than in the case of non-cooperative intelligent systems. Usually, cooperative multiagent systems are presented to be intelligent, based on the consideration that the cooperation between individual (even very simple) agents results in the emergence of intelligence at the level of the whole system [46,47]. This is the definition of intelligence that we adopt in our research. Such a simple consideration is just intuitive without offering evidence of intelligence, in terms of a measure that quantifies the level of intelligence.

For example, Yang *et al.* [48] presented an intelligent mobile multiagent system composed of simple reactive agents endowed with knowledge retained as a set of rules describing network administration tasks. This cooperative multiagent system could be considered as intelligent based on the intuitive motivation that it mimics the behavior of a human network administrator, who acts in an intelligent way as a human specialist. However, that study neither provided a quantitative measure to the intelligence of the cooperative multiagent system, nor an effective numerical comparison of a system's intelligence with the intelligence of another system.

Machine learning frequently is presented as a successful approach for different problems solving even in very recent studies and researches [49–51]. This proves that computing systems, including ABSs could use such learning techniques in order to solve efficiently problems. In the mentioned works and many others, the way how the learning leads to effective improvements is not effectively proved or even studied. Paper [49] presents a survey on supervised machine-learning techniques that lead to efficient automatic text classification. In [50] a smart adaptive run parameterization enhancement of user manual selection of running parameters in fluid dynamic simulations using machine-learning techniques is proposed. Paper [51] proposes a machine-learning approach to identify multimodal MRI patterns predictive of isocitrate dehydrogenase and 1p/19q status in diffuse low- and high-grade gliomas.

There are very few studies and researches that formulate the pertinent and important research question, if the learning does not have any kind of effect, or even worst it could lead as result decreases or errors. In the research [52] it is studied the problem of inclusion of machine-learning components (MLCs) in cyber-physical systems (CPSs). In case of such a CPS, the researchers state that the operation correctness could depends even in a high degree on the incorporated MLC(s). In this context, it was formulated the research question: "Can the output from learning components lead to a failure of the entire CPS?" As a practical application, the problem of falsifying signal temporal logic specifications for CPS endowed with MLC(s) is addressed. According to this requirement, it is proposed a compositional falsification framework. The effectiveness of the technique that is proposed is proved on an automatic emergency braking system that is endowed with a perception analysis based on deep neural networks component.

Another study [53] investigated the subject of inconsistent data, that appear as outliers, which can be learned by a diagnosis system if is not identified and deleted. Such data could have a negative effect to accuracy and performance of the diagnosis system. For outlier data identification a k-Nearest Neighbor (k-NN) method was applied.

Knowing the fact that the learning process does not have an effect or it could lead to a decrease or apparition of errors, it may be helpful to take a different solution in order to avoid such effects, like analyzing the result of this learning process, if there is a chance to lead to a malfunction for example.

2.2. Learning Ability Assessment

An important aspect in the study of learning systems is the assessment of their ability to learn. In this paper, we investigate an approach to measure if learning can lead to a measurable increase in the intelligence level. It should be noted that if a system learns, it does not necessarily mean that it has become more intelligent, in the sense that it has a measurable increase in the intelligence level. Learning in some cases could result even in a decrease of intelligence, for example, a system could learn misleading data or overfit on the training data thus losing its ability to generalize. Furthermore, we consider that the intelligence measure modification (increment or decrement in some cases), does not directly relate to the learning complexity. Even a very simple form of autonomous learning, such as rote learning, can result in a significant increase (e.g., of the performance), and sometimes even of the intelligence [40].

We consider intelligence level measuring of a CMAS based on difficult problem-solving ability. An intelligence increase can be quantified by a measurable increase of the problem-solving ability/intelligence, whereas an intelligence decrease can be quantified by a measurable decrease of the problem-solving ability/intelligence. In order to further explain the notion of difficulty and intelligence increase or decrease in problem-solving, as an example, let us consider a simplified scenario of an intelligent medical diagnosis system that is able to learn. The intelligent system is specialized in the solving of a very difficult diagnosis problem, for example, an illness for which the establishment of an effective treatment is very difficult. In this context, the variability of intelligence can be associated with the selection of a specific treatment that could be more or less effective. An intelligence increase can be associated with the elaboration of more effective treatments, whereas an intelligence decrease can be associated with the elaboration of less effective treatments.

A problem with a specific type could be solved by a variety of CMASs with different architectures. Even if we consider a particular domain of knowledge there are a lot of solvable problems by computing systems with a wide diversity of types and complexity. This motivates the necessity to design universal metrics for measuring machine intelligence. However to date, as it is also pinpointed in the review study presented in the next sections, current metrics have limitations in their universality.

2.3. Metrics for Measuring Machine Intelligence

In 1950, Alan Turing [54], considered a computing system as intelligent, if a human assessor could not decide the nature of the system (being human or otherwise) based on questions asked from a hidden room. We consider that the biological and Artificial Intelligence (AI) are different in nature; therefore, they should not be compared directly. Although Turing's test was an early theoretical proposal in AI; we consider it as a good starting point for the design of metrics/methods enabling the comparison of intelligence of two systems by the same type (biological or artificial). Some milestones of AI could be clearly set this way, for example, using metrics to assess intelligence in competitions between systems and humans, such as the well-known competition between the chess machine named Deep Blue and the chess master Garry Kasparov [55], and between the IBM Watson computing system and human experts in the game named Jeopardy [56]. Schreiner [57] accentuated the necessity of creating metrics for measuring the systems' intelligence in a study performed for the US National Institute of Standards and Technology and proposed relevant measurement and comparison approaches.

Legg and Hutter [58] studied a number of well-known definitions of human intelligence and extracted their essential features. These were then mathematically formalized in order to produce a general measure of machine intelligence. The authors showed that this formally defined measure is related to the theory called universal optimal learning agents. Hibbard [59] proposed a novel metric for measuring an agents' intelligence, which is based on the theory/principle of the hierarchy of sets of increasingly difficult environments. Hibbard considers an agent's intelligence measurement according to the ordinal of the most difficult set of environments that may occur. The metric proposed in that work includes the number of time steps required for an agent to pass the test.

Hernandez-Orallo and Dowe [60] proposed the idea of universal anytime intelligence test. They considered that such a test should be universal in the sense that could be able to measure intelligence, in a wide variety of situations that include even very low or very high intelligence. Their proposal was based on the so-called C-tests and compression-enhanced Turing tests that were designed in the late 1990s. In that study, a synthesis of different state-of-the-art tests was performed, highlighting their limitations.

Sometimes the agents' intelligence is considered based on the complexity of the problems that they are able to solve. Anthon and Jannett [61] investigated the intelligence measuring appropriateness of the agent-based systems based on the ability to compare different alternatives considering their complexity. In their experimental setup, a distributed sensor network system was considered. This approach was tested by comparing an intelligence measure in different agent-based scenarios.

Winklerova [62] assessed the collective intelligence of a particle swarm optimization system according to a novel Maturity Model. The proposed approach was based on the Maturity Model of

Command and Control operational space and the model of Collaborating Software. The main aim of that study was to obtain a more thorough explanation of how the intelligent behavior of the particle swarm emerges.

Franklin and Abrao [63] proposed an introductory methodology for the agent's intelligence testing. This methodology is based on the calculation of a proposed general intelligence factor and the theory of multiple human intelligences [64]. A set of tests assessed the multiple intelligences of the agents, by analyzing their problem-solving behavior in different situations. The proposed approach was intended for both qualitative and quantitative evaluations of intelligence.

In [65] we proposed an innovative metric called MetrIntPair (Metric for Pairwise Intelligence Comparison of Agent-Based Systems) for comparison of two cooperative multiagent systems problem-solving intelligence. MetrIntPair is able to make an accurate comparison by taking into consideration the variability in the problem-solving intelligence of systems. Two intelligent systems with the same intelligence can be included in the same class of intelligence. The design of metrics that are able to make differentiation of systems in problem-solving with respect to their intelligence is a very important subject since they can be used to select the system that is able to solve a particular problem in the most intelligent way.

In [66] we proposed a metric called MetrIntMeas (Metric for the Intelligence Measuring) for measurement of the intelligence of a swarm system for difficult problem-solving. MetrIntMeas is an accurate and robust metric enabling at an application the classification based on the intelligence of a studied swam system. It is able to verify if a studied swarm system belongs to the same class with the systems which have a specific reference intelligence value. In that paper, the intelligent evolving systems were defined stating that the evolution in the intelligence of a swarm system should be measured by using the MetrIntMeas metric.

In [67] we highlighted the fact that the identification of intelligent systems (not particularly intelligent learning systems) with extremely-low or extremely-high, outlier, intelligence is an important subject, and we proposed a method called OutIntSys for the detection of systems with high and low outlier machine intelligence from a set of studied intelligent systems. We consider that the treating of outliers is a very important subject even in the case of metrics for measuring the intelligence of learning systems. Outlier values could have a negative influence on the accuracy of the intelligence measurements.

In our previous studies [65,66] we highlighted the fact that a metric for intelligence measurement needs to treat the aspect of variability in intelligence. The metrics presented in [65,66] are illustrative to the situation where the treatment of the variability by a metric can result in advantages to the accuracy and robustness in intelligence level comparison and classification of the systems based on their intelligence. A disadvantage of many previously proposed intelligence metrics in the literature consists of limited universality. Universality provides advantages including independence of the metric from the measured agent/cooperative multiagent system architecture, the environment in which the system operates, and other influencing factors. Metrics proposed in the literature are designed based on different kind of considerations/principles; therefore, most of them cannot be effectively compared with each other.

The literature review performed in this section reveals that there is no unanimous viewpoint related to what a metric for measuring the intelligence should measure. There is no accepted standardization related to intelligence measurement. Most intelligent systems are able to learn. The design of effective metrics for artificial learning systems is necessary. We consider that a metric for measuring machine intelligence must conserve the very important properties by universality, accuracy and robustness. In the case of an intelligent learning system under investigation, such a metric should be useful at least in the following processing/analysis tasks: 1) to measure the intelligence of the system before a learning process; 2) to measure the intelligence of the system after a learning process; 3) to verify if the learning process results in any modification in intelligence level; 4) to be able to verify accurately even small changes in intelligence.

We would like to outline that a measurable improvement by learning does not necessary means an increase in the intelligence. There are very few studies focusing on the effect of modification of the intelligence as a consequence of learning [68,69]. There are no studies explicitly focusing on the design of metrics able to verify accurately (taking into consideration the variability in the intelligence) that even small changes, as result of learning, can have as an effect an evolutionary or involutionary step in intelligence.

3. THE PROPOSED II-LEARN METRIC

In order to cope with the afore-mentioned open issues, we propose II-Learn metric, which can be used to provide quantifiable evidence of intelligence, suitable for assessing the intelligence increase of a cooperative multiagent system by learning, while conserving universality and providing enhanced accuracy and robustness. In the following we describe this metric after the introduction of the explanation of the principle and the formalism on which it is based on.

3.1. Principle

There are many studies and researches [46–48] that define CMASs composed of extremely simple agents as intelligent based on the efficient, flexible and robust cooperative problem-solving. It is considered the intelligence as emergent at the system's level. Frequently the autonomous learning capacity is associated with the intelligence of a system. In our approach we are focused on CMASs where intelligence emerges at the system's level, that are able to learn. The rationale of the metric that we propose is to measure if by learning a studied CMAS evolved in intelligence. In our approach, the intelligence of such a considered CMAS must allow the solving of very difficult problems. We consider the principle of cooperative multiagent system's intelligence level measuring based on the advanced ability to solve difficult problems. The difficult problem-solving ability is considered as the intelligence level measure.

3.2. Formalism

In our approach, it is considered the measuring of a specific type of intelligence considered by interest to a human evaluator (HE) who is a specialist in machine intelligence science.

Different studies discuss human specialist who acts as evaluators. Paper [70] presents a methodology for analysis of different kind of human errors. The US Patent [71] proposes a general approach to some methods for presentation and evaluation of constructed responses assessed by the HEs. In [72] is investigated the HE in image enhancement tasks with the purpose to make visual improvements of images. Paper [73] study the perceived value of two designed sentiment analysis tools for understanding the Finnish language, in contrast to HEs. In [74] the principal role of HEs is analyzed in the applied automatic semantic technologies.

In our approach HE is responsible to measure if a CMAS system's intelligence level increased as a result of learning by using the *II-Learn* metric. A problem-solving intelligence evaluation result by the considered type is expressed as a so-called intelligence indicator value. An intelligence indicator value is based on a specific calculus performed on an experimental problem-solving intelligence evaluation. In order to illustrate the previously introduced notions, we provide the following a scenario.

Let us consider an intelligent CMAS denoted *CoopRob* composed of robotic agents able to pilot cooperative transport cars. The agents must cooperatively perform different missions established by a human specialist(s) denoted *HE*. The first type of mission consists of the transportation of a set of objects to established destinations. The task of distribution of a larger set of objects to different destinations is undertaken for solving by the transport cars. A type of problems to cope with this situation can be the TSP. The latter type of problems consists in the collection of a set of objects from distributed sources that are established or they should be found/discovered. Based on the efficient cooperative solving of difficult problems the intelligence can be considered at the cooperative system's level. The intelligence of such a system cannot be unanimously defined. The human specialist(s) who would like to measure the intelligence of *CoopRob* must define what type of intelligence he/she would like to measure. The intelligence can be considered based on different particular or general considerations/principles (or their combination), such as advanced ability to collect objects or advanced ability to distribute objects. *HE* should establish the corresponding calculus of problem-solving intelligence indicator value that corresponds to the type of intelligence by interest.

Our approach includes a calculus of the so-called CIT before the learning process and after the learning process. The CITs of a learning system before and after learning are not absolute measure; they are based on the calculated intelligence indicator values obtained as the result of the problem-solving intelligence evaluations. Along with the CITs some other calculated indicators, which characterize the intelligence level are computed. For example, we mention the indicator that measures the homogeneity-heterogeneity of problem-solving intelligence level. We consider that similarly with the humans, the systems can have a variability of intelligence in problem-solving.

3.3. The II-Learn Metric Description

Evolving systems presented in the literature often evolve gradually based on methods such as autonomous learning, inheritance, self-adaptation, or changing of the structure [75–77]. We consider that an important aspect, which to our knowledge is still untreated, is the study of evolution in intelligence during a system's learning process. Based on this consideration, in the following, we define the notions: "Evolutionary Step," "Involutionary Step," "Intelligent Evolving Learning System" and "Intelligent Involving Learning System."

Definition 1. Evolutionary step made by an intelligent learning system. *We call evolutionary step made by an intelligent learning system, a measurable increase in intelligence by using the II-Learn metric as a result of a learning process.*

Definition 2. Involutionary step made by an intelligent learning system. *We call involutionary step made by an intelligent system, a measurable decrease in intelligence by using the II-Learn metric as a result of a learning process.*

Definition 3. *Intelligent Evolving Learning System.* *We define an Intelligent Evolving Learning System, as a learning system able to make at least one evolutionary step in intelligence, measurable by the II-Learn metric. An intelligent evolving learning system could make more evolutionary steps during its life cycle. Each step should be measurable by using the II-Learn metric.*

Definition 4. *Intelligent Involving Learning System.* *We define an Intelligent Involving Learning System, as a learning system that makes at least one involutionary step in intelligence, measurable by the II-Learn metric.*

Let us consider a cooperative multiagent system denoted as ILS^{BL}, $ILS^{\mathrm{BL}} = \{ILS_1, ILS_2, \dots.., ILS_{\mathrm{n}}\}$, that is able to learn. The members of this set, $ILS_1, ILS_2, \dots.., ILS_{\mathrm{n}}$ represent different agents, where, $n = |ILS^{\mathrm{BL}}|$ is the cardinality (the number of member agents) of ILS^{BL}. We denote with ILS^{AL} the cooperative multiagent system that results after learning. ILS^{BL} may have any architecture, and it may learn autonomously by using any learning technique. The learning in ILS^{BL} may result in agent-level (new acquired data/information/knowledge etc.) or system-level (duplication of efficient agents, modification of the structure of the system etc.) modifications.

Let us denote as $IntA = \{IntA_1, IntA_2, \dots, IntA_{\mathrm{r}}\}$ the measured problem-solving intelligence indicators obtained during the intelligence evaluation of ILS^{BL} in solving a set of test problems *ProblA*. In this notation, $r = |IntA|$ represents the sample size of the measured intelligence indicators in *IntA*. Let $IntB = \{IntB_1, IntB_2, \dots, IntB_{\mathrm{k}}\}$ denote the measured problem-solving intelligence indicators obtained during the intelligence evaluation of ILS^{AL} in solving a set of test problems *ProblB*. $k = |IntB|$ represents the sample size of the measured intelligence indicators in *IntB*.

The establishment of the numbers of problems of the sets *ProblA* and *ProblB* is based on a mathematically grounded calculus. *HE* is responsible for the effective choosing of the problems of the sets *ProblA* and *ProblB*. This must be made by taking into consideration the type of intelligence that is intended to be measured, which corresponds to a certain type of problem-solving by a specific dimensionality/complexity.

The following *Normality Verification and Extraction* (*VerExtr*) algorithm describes the intelligence characterization made before (ILS^{BL}) and after (ILS^{AL}) a learning process.

The notation "@" used in the algorithm indicates a set of processing and/or analyses that are executed. For example "@Verify the *IntA* and *IntB* data normality," indicates the performing of a statistical analysis of the data normality. The intelligence comparison of ILS^{BL} and ILS^{AL} is based on the specific mathematical calculus of *II-Learn* algorithm, which invokes the *VerExtr*. *II-Learn* algorithm is not restrictive to the equality of the number of problem-solving intelligence evaluations ($|IntB|$ and $|IntA|$) for ILS^{BL} and ILS^{AL}.

```
**VerExtr:**
**Normality Verification and Extraction Algorithm**
**Input:** $IntA = \{IntA_1, IntA_2, \ldots, IntA_r\}$;
$IntB = \{IntB_1, IntB_2, \ldots, IntB_k\}$;
**Output:** ($CentrInd_A$, $LCIm_A$, $HCIm_A$, $SD_A$, $CV_A$);
($CentrInd_B$, $LCIm_B$, $HCIm_B$, $SD_B$, $CV_B$);
*Begin*
@Set $\alpha N$ the normality test significance level;
GotoLabel;
@Verify the IntA and IntB data normality;
If ((IntA and IntB) are normally distributed) then
    Norm: = "YES";
    Else
    @Opting for a transformation or elimination
     of outlier values. Update of IntA and IntB.
    Goto GotoLabel;
EndIF
@Calculate the principal indicators of the Central Intelligence
Tendency: $CentrInd_A$ and $CentrInd_B$.
@Set the CL value;
@Calculate the additional indicators.
$LCIm_A$, $HCIm_A$, $SD_A$, $CV_A$; $LCIm_B$, $HCIm_B$, $SD_B$, $CV_B$;
EndNormalityVerificationExtraction
```

In the case of ILS^{BL} and ILS^{AL} the *VerExtr* algorithm calculates the principal CIT indicators denoted as $CentrInd_A$ and $CentrInd_B$. Additionally, there are calculated some other additional indicators that characterize the intelligence. *CentrInd* is calculated as the mean of the intelligence indicator data that result from the experimental intelligence evaluations. The *Standard Deviation* (*SD*) [78] of the intelligence indicators, denoted as SD_A and SD_B, are calculated to quantify the amount of variation of the respective samples. The *Coefficient of Variation* (*CV*) calculated as $CV = 100 \times (SD/Mean)$, is considered also as such a quantifier, normalized by the mean of the total number of samples. In *VerExtr* algorithm, CV_A indicates the CV of the *IntA* data, CV_B indicates the CV of the *IntB* data. We consider necessary the establishment of the confidence interval of the mean at a specific *Confidence Level* (CL). In most cases, a CL of 95% can be considered as appropriate. As examples of other values of CL that can be chosen we mention 90% and 99%. For the ILS^{BL}, $LCIm_A$ and $HCIm_A$ represent the low and high bounds of the confidence interval of the mean. For the ILS^{AL}, $LCIm_B$ and $HCIm_B$ represent the low and high bounds of the confidence interval of the mean.

The One-Sample Kolmogorov–Smirnov test (*K–S test*) [79], *Lilliefors test* [80,81] and *Shapiro–Wilk test* [82,83] are among the most frequently applied goodness-of-fit tests used for data normality verification. For the normality testing, if *IntA* and *IntB* are normally distributed, we suggest the application of the *One-Sample Kolmogorov–Smirnov*, at significance level $\alpha N = 0.05$. As examples of other significance levels that can be chosen we mention 0.01 and 0.1. If considered necessary, as an alternative, a powerful verification of the normality assumptions we suggest the application of the *Shapiro–Wilk test* also. Some studies [82,83] prove that the *Shapiro–Wilk test* has the best statistical power, considering the most frequently used tests for the verification of the normality: One-Sample Kolmogorov–Smirnov, Lilliefors, Shapiro–Wilk and Anderson–Darling, for a specific significance level.

The Quantile–Quantile plot (Q–Q plot) is a scatterplot appropriate for the normality visual appreciation. A Q–Q plot is created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, the points form a line that is roughly straight. The joint use of Q–Q plot with the Shapiro–Wilk test is suggested for checking the normality.

II-Learn metric verifies whether the intelligence of ILS^{BL} has changed (increased or decreased) as a result of learning or remained the same. In the framework of *Verify Evolution in Intelligence by Learning* algorithm is verified if both data sets *IntA* and *IntB* pass the normality assumption (approximately normal distribution, in case of real-life data is not expectable perfect normality). If the sample intelligence data is not normally distributed, a solution to obtain normally distributed data consists in the application of a transformation that should corresponds to the type of data distribution. The transformation should be applied to both ILS^{BL} and ILS^{AL}. This is required for the proposed metric to provide a trustworthy result. Some of the most common normalizing transformations are indicated in Table 1 [84].

We consider as an outlier intelligence indicator value, an extremely high or extremely low intelligence value, i.e., a value that is statistically different from those other intelligence indicator values. A further from the rest intelligence indicator value is not an outlier but it is statistically further from the rest. An intelligence indicator dataset could contain no outlier (same for the further from the rest values) values, or in alternative cases, it may contain one or more outlier intelligence values (same for the further from the rest values). The number of outlier values (same for the further from the rest values) usually cannot be established in advance. If an intelligence indicator dataset does not pass the normality assumption, then an alternative option to the transformation consists in the removal of the outliers, and/or the further from the rest intelligence indicator values. If it is requested or expected that the intelligence indicator data can be reasonably approximated by a normal distribution (e.g., in case there exist a problem specific knowledge, that in some similar cases, the obtained data is normally distributed), then we suggest the Grubbs test [85,86] for the identification of outlier values. An approach to formulate the fact that is expectable data normality can be concluded by visual analyzing of Q–Q plot. We suggests the application of the Grubbs test with the significance level $\alpha G = 0.05$ in most of the cases. As examples of other significance level values that can be chosen we mention 0.01, 0.1. The *Grubbs test* can be applied recursively more times until no outliers

Table 1 Normalizing transformations.

Type of Distribution	Normalizing Transformation
IND is Lognormal	Log(***IND***)
IND is Binomial	Arcsine(SquareRoot(***IND***))
IND is Poisson	SquareRoot(***IND***)

(or suspicious values) can be detected. At each application, it is able to identify at most an outlier/suspicious value if there exists such a value.

We call *Null Hypothesis* and denote it as $H0_{ei}$, the statement that the $CentrInd_A$ of ILS^{BL} is equal from the statistical point of view with the $CentrInd_B$ of ILS^{AL}. We call *Alternative Hypothesis* and denote it with $H1_{ei}$ the hypothesis that the $CentrInd_A$ of ILS^{BL} is different from a statistical point of view from $CentrInd_B$ of ILS^{AL}. The testing of $H0_{ei}$ and $H1_{ei}$ is realized with the significance level denoted αMet. αMet represents the probability of rejecting the *Null Hypothesis* when it is true. αMet is a parameter of the algorithm. The most frequently used values are 0.05, 0.01 and 0.1. We suggest a value of 0.05 for the αMet. This value indicates the probability of apparition of a type I error. A type I error is the incorrect rejection of a true null hypothesis. Many studies [87] related to statistics prove that in most of the cases the significance level by 0.05 is the most appropriate to be selected. Our decision is based also on the relation between the type I and type II errors. A type II error is incorrectly retaining a false null hypothesis. Decreasing the type I error rate from 0.05, have as result the increase of the type II error rate probability of apparition.

II-Learn – Algorithm

Verify Evolution in Intelligence by Learning

```
Input: $IntA = \{IntA_1, IntA_2, \ldots, IntA_r\}$;
$IntB = \{IntB_1, IntB_2, \ldots, IntB_k\}$;
Output: IntelligenceComparisonDecision;
Step 1. Preprocessing and analyzing.
@Apply VerExtr algorithm;
@Set $\alpha Met$; //Significance level of hypothesis testing.
Step 2. Verify if IntA and IntB have equal SD.
@Set $\alpha F$; //Significance level of the F-test.
@Verifies the standard deviations equality using F-test;
If ($SD_A = SD_B$ from the statistical point of view) then
    SDInd: = ”YES”;
    Else SDInd: = ”NO”;
EndIF
Step 3. Verification. of making an evolutionary step.
@Formulate $H0_{ei}$ (the Null Hypothesis) and $H1_{ei}$ (the Alternative Hypothesis);
If (SDInd = ”YES”) then
    @Apply the Unpaired Two-Sample T-test;
    @Calculates Pval (the P-value of the test);
    Else
    @Apply the Welch corrected Unpaired Two-Sample
    T-test; Calculates Pval (the P-value of the test);
EndIF
Step 4. Interpretation of the evaluation results.
If ($Pval > \alpha Met$) then
    @Accept $H0_{ei}$. “$ILS^{BL}$ intelligence has not changed.”
    else
      @Accept $H1_{ei}$. “$ILS^{BL}$ intelligence has not changed.”
      If ($CentrInd_A > CentrInd_B$) then
        “$ILS^{BL}$ intelligence increased; $ILS^{BL}$ evolved”.
        Else
          “$ILS^{BL}$ intelligence decreased; $ILS^{BL}$
          involved”.
      EndIf
EndIF
EndII-Learn
```

In the algorithm *Verify Evolution in Intelligence by Learning*, for the $H0_{ei}$ testing, we considered the application of the *Two-Sample Unpaired T-test* [88–90] as most appropriate in the case of equality from the statistical point of view between the standard deviations SD_A and SD_B. In addition, the *Welch Corrected Two-Sample Unpaired T-test* (Welch test) [90–92] was considered, in the case the standard deviations SD_A and SD_B are not equal from the statistical point of view. We propose the use of two-tailed tests in both cases (*Two-Sample Unpaired T-test* and *Welch Corrected Two-Sample Unpaired T-test*). For the verification of the equality of SDs, we choose the *F-test* [93], applied at the significance level $\alpha F = 0.05$.

3.3.1. Definition of the II-learn metric

We define II-Learn metric as being able to measure the statistically significant increase or decrease in the intelligence level of a studied ICMAS as result of a learning process, which leads to evolution or involution in intelligence. The processing and analyses performed in the frame of the metric are described by the Verify Evolution in Intelligence by Learning algorithm.

If $H0_{ei}$ is verified then it can be concluded that ILS^{BL} intelligence has not changed from the statistical point of view by learning. The numerical difference is the result of the variability. It should be noted that by repeating the experimental conditions slightly different experimental results will be obtained (heuristic problem-solving behavior) but the formulated conclusion will be the same.

If $H1_{ei}$ is accepted, and $CentrInd_A > CentrInd_B$ then it can be derived that ILS^{BL} intelligence has increased, and that ILS^{BL} has evolved in intelligence by making an evolutionary step. If $H1_{ei}$ is accepted, and $CentrInd_A < CentrInd_B$ then it can be derived that ILS^{BL} intelligence has decreased, and that ILS^{BL} has involved in intelligence.

For the establishment of the sample sizes ($|IntA|$ and $|IntB|$) is performed an a priori calculus [94,95]. The notion a priori is used with the significance that the precise necessary sample size is established at the beginning of the experimental problem-solving intelligence evaluation. It is not based on the methodology of verification after finishing the experimental evaluations what the influence of the chosen sample size has to the formulated conclusions. The calculation [94,95] of the |*IntA*| and |*IntB*| is based on: *αMet, PowerMet* (*PowerMet* = 1-*βMet*), *number of tails* (we considered the application in most of the cases the test with two tails), *allocation ratio* (*allocation ratio* = |*IntA*| / |*IntB*|; for example, *allocation ratio* = 1 indicates that the sample sizes |*IntA*| and |*IntB*| are equal) and the *Effect size d* (*Cohen's d*) [96]. As previously mentioned, *αMet* denotes the probability of a type I error. *βMet* denotes the probability to make a type II error. Can be formulated that a type I error is detecting an effect that is not present, while a type II error is failing to detect an effect that is present. An effect size is a quantitative measure of the strength of a phenomenon. The importance of calculation of the effect size is discussed in the papers [96,97]. The *Effect size d* (*Cohen's d*) [96] is given by the formula $d = (CentrInd_{\mathrm{A}} - CentrInd_{\mathrm{B}})/S$. Where: $CentrInd_{\mathrm{A}}$ and $CentrInd_{\mathrm{B}}$ denote the means of the two samples; *S* denotes the SD of either group. $d = 0.2$ indicates a small effect size; $d = 0.5$ indicates a medium effect size; $d = 0.8$ indicates a large effect size; $d = 1.3$ indicates a very large effect size.

3.4. Components of the Intelligence Measure

In our research, we considered the cooperative multiagent systems intelligence measured at the level of the whole system. The proposed *II-Learn* metric is appropriate for systems, where the collective intelligence indicator can be expressed as a single evaluated measure. If necessary, for different types of systems, based on their specificity, it can be calculated as a weighted sum of some other intelligence components that characterize different aspects of the considered system's intelligence. Eq. (1) presents the general case when an intelligence indicator *IntInd*, is calculated as the weighted sum of *q* types of intelligence components measure at a problem-solving, where: 1,2,...*q* represent the identifiers of the intelligence components, mas_1, mas_2,..., mas_{q} represent the considered intelligence components of measure, and wgh_1, wgh_2,...,wgh_{q} represent their weights.

$$\begin{aligned} IntInd = {} & wgh_1 \times mas_1 + \ldots + wgh_{\mathrm{q}} \times mas_{\mathrm{q}}; \\ & wgh_1 + \ldots.. + wgh_{\mathrm{q}} = 1 \end{aligned} \tag{1}$$

The weights indicate the importance in the calculus of the problem-solving intelligence. For example, $wgh_{\mathrm{i}} = wgh_{\mathrm{k}}$ means that mas_{i} and mas_{k}, the measures of components *i* and *k*, have the same weight in the calculus of the intelligence; if $wgh_{\mathrm{i}} < wgh_{\mathrm{k}}$ means that mas_{i} is less important (has a lower weight) than mas_{k} in the calculus of the intelligence.

4. MEASURING THE INTELLIGENCE INCREASE OF A LEARNING COOPERATIVE MULTIAGENT SYSTEM USING II-LEARN. AN EXPERIMENTAL CASE STUDY

There are many studies related to the intelligence of biological swarms/colonies, like the ant colonies, composed of simple living creatures, which at the swarm level have an amazing surviving capacity that could be associated with some kind of intelligence [98].

Marco Dorigo in his Ph.D. thesis [99,100] proposed for the first time a generic problem-solving methodology based on simple computing agents (artificial ants) that mimic the behavior of a natural ant colony in search for food. Similarly to the biological ants, the artificial ants (agents) wander randomly, and upon finding a solution of the problem (in case of biological ants finding foods) they return to their home (colony in case of biological ants) while laying down signs understandable to other agents called pheromone. The name pheromone is established based on the analogy with biological ants. In the case of biological ants, the pheromone trails represent semiochemicals secreted by the body of the ants. In case of the artificial agents, this is a numerical value that represents intensity. If a natural ant finds a path with a high pheromone amount, it will likely follow that trail, returning and reinforcing it if eventually find food. This is a specific communication and collaboration of the biological ants as a result of a long-term evolution. Such communication in case of artificial ants allows an efficient, robust (if some agents fail to operate during a problem-solving, the problem-solving can continue even this case) and scalable (the number of operating ants could be increased even during a problem-solving) cooperation. Over time, the pheromone trail starts to evaporate (its intensity decreases), thus reducing its attractiveness to the ants. The more time it takes for an ant to travel down the path and back again, more time the pheromones have to evaporate. A short path gets marched more frequently, and thus the pheromone density becomes higher on shorter paths than longer ones.

Ant Colony Optimization algorithms (*ACOs*) have been seen as a suitable model for distributed reinforcement learning [101,102]. Many ACO implementations belong to the *Ant-Q* family [103].

Ant algorithms have many applications, like the emergency management using geographic information systems [42]. One of the most important applications includes transportation [42,43,104,105], with the *Vehicle Routing Problem* (*VRP*) representing an important generalization of the TSP [106]. *VRP* is an NP-hard problem as TSP. It searches for the optimal set of routes for a fleet of vehicles to traverse in order to make a delivery to a given set of customers [106]. Examples of important applications of *VRP* in healthcare include medical emergency management [107], and medical supplies insurance in large-scale emergencies [108].

The *Ant System* (AS) is the first ACO, which proved to be a viable method for solving hard combinatorial optimization problems. Over time, different variants of this algorithm have been designed. In an *AS*, initially, each agent (artificial ant) is placed on some randomly chosen node. An agent *k* currently at a node *i* choose to move to node *j* by applying the probabilistic transition rule (2). After each agent completes its tour, the pheromone amount on each path will be adjusted according to Eqs. (3–5).

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha} \times \left[\eta_{ij}\right]^{\beta}}{\sum\limits_{l \in J_k(i)} \left[\tau_{il}(t)\right]^{\alpha} \times \left[\eta_{il}\right]^{\beta}} & \text{if } j \in J_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{3}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{4}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L_k, & \text{if } (i,j) \in \text{ tour done by agent } k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

In Eqs. (2–5) the following notations are used: Q denotes an arbitrary constant; α and β are adjustable parameters that control the relative weights of the heuristic visibility and the pheromone trail. In the parameters establishment, a trade-off between edge length and pheromone intensity appears to be necessary. Also, $h_{kh}, h_{kh} = 1/d_{k,h}$ represents the heuristic visibility of edge (k, h), with $d_{\mathrm{k,h}}$ to be the distance between the cities (k and h), and $0 < \rho < 1$ to represent the trail evaporation. m denotes the number of ants. L_k denotes the length of the tour performed by agent k.

The *Best-Worst Ant System* (BWAS) [109,110] model tries to improve the performance of *ACO* models applying some modifications in the specific way how the artificial ants search for food. *BWAS* achieves exploitation of the search by allowing both the iteration-best agent and the iteration-worst agents to update the pheromone on the traversed trail. It makes use of the positive feedback of iteration-best agent and use of the negative feedback of iteration-worst agent. This property has been proved efficient in different problem-solving. The use of this simple mechanism to limit the strengths of the pheromone trails, effectively avoids premature convergence of the search.

In a *BWAS* the applied probabilistic transition rule is defined by Eq. (2). After each agent completes its tour, the evaporation is applied according to Eq. (6) on all the edges (i, j). The iteration-best agent and iteration-worst agent updates pheromones.

$$\tau_{ij}(t) = (1-\rho) \times \tau_{ij}(t) \tag{6}$$

In the following, k denotes the iteration-best agent; L_{k} is the length of the tour performed by the agent k. The iteration-best agent update of pheromones is indicated in Eqs. (7) and (8):

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L_k, & \text{if } (i,j) \in \text{ tour done by agent } k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\tau_{ij}(t) = \tau_{ij}(t) + \Delta\tau_{ij}^{k}(t) \tag{8}$$

On the paths of the round trip of the iteration-worst agent for the current iteration that are not in the best-to-date solution has additional evaporation as indicated in the following:

$$\tau_{rs}(t) = (1-\rho_w) \times \tau_{rs}(t) \tag{9}$$

where ρ_w is an additional factor for all $L_{rs} \in T_w$ and $L_{rs} \notin T_w \cap T_{BS}$, T_w is the worst solution for the given iteration, and T_{bs} is the best-to-date solution. Eq. (10) establishes the final pheromone update at the end of iteration t on all the edges (i, j).

$$\tau_{ij}(t+1) = \tau_{ij}(t) \tag{10}$$

A *Min–Max Ant System* (*MMAS*) [97,111,112] is an ACO, a variant of the AS. *MMAS* have some differences from the AS in some aspects. The *MMAS* can be seen as an Interactive Machine Learning implementation, with external intervention [113]. An *MMAS* give dynamically evolving bounds on the pheromone trail intensities, this is done in such a way that the pheromone intensity on all the paths is always within a specified limit of the path with the greatest pheromone intensity. All the paths will have permanently a non-trivial probability of being selected. This way a wider exploration of the search space is assured. *MMAS* uses lower and upper pheromone bounds to ensure that all of the pheromone intensities are between this two bounds.

In an *MMAS* the applied probabilistic transition rule is defined by Eq. (2). There are allowed to update pheromones: the best-for-current-iteration or best-to-date agent or the best-after-latest-reset agent or the best- to-date agent for even (or odd) iterations. There are minimal and maximal pheromone limits to the quantity of pheromone on the paths between cities, denoted as τ_{min} and $\tau_{max.}$ The evaporation can be expressed by Eq. (11), $\rho \in (0,1)$ represents the trail evaporation. Eq. (12) expresses the pheromone update based on the selected agent's round trip

$$\tau_{ij}(t) = \max\left((1-\rho)\cdot\tau_{ij}(t), \tau_{\min}\right) \tag{11}$$

$$\tau_{ij}(t+1) = \min\left(\tau_{ij}(t) + \Delta\tau_{ij}^{bs}(t), \tau_{\max}\right) \tag{12}$$

where $\Delta t_{ij}^{bs}(t) = Q/L^{bs}$ if the path $ij \in T^{bs}$, T^{bs} is the selected best-to-date agent's round trip L^{bs} is the length of the trip. It was used for initiation $\tau_0 = 1/nr$, where nr denotes the *number of cities*.

In our experimental setup, we considered ILS^{BL} operating as a *BWAS* [109,110]. ILS^{BL} focuses on the solution of the *TSP* [42,43]. There are many applications, adaptations and similar problems with the TSP. For example, Borsani *et al.* [114] proposed a human resource scheduling model of home health service, and Kergosien *et al.* [115] proposed a home health care problem that resembles an extended, multiple TSP.

We considered a simple rote-learning approach, where ILS^{BL} simply copies the behavior of a *MMAS* [97,111,112]. After learning, the obtained multiagent system is denoted as ILS^{AL}. The effect of learning is the adaptation of the cooperative multiagent problem-solving behavior. In our case study, we aim to verify if the adaptation has as result a statistical significant change (increase or decrease) of the ILS^{BL} intelligence using the proposed metric.

We applied an a priori calculus in order to establish the necessary experimental evaluations, $|IntA|$ and $|IntB|$.

Calculation Input: $\alpha Met = 0.05$; $PowerMet(1-\beta Met) = 0.8$; $|IntA| / |IntB| = 0.96$ (is not requested to sample sizes to be the same, $|IntA|$ to be equal $|IntB|$); Tails = *two; Effect size d* = 0.575 (*medium effect size*).

Calculation Output: $|IntA| = 48$, $|IntB| = 50$.

In the experimental setup, we considered maps with $nr = 35$ randomly placed cities on the map. The problems considered in the experimental problem-solving intelligence evaluations are by the same type and dimensionality/complexity in the case of both ILS^{BL} and ILS^{AL}. The parameters were considered: $No = 1000$ (*number of iterations*); α, $\alpha = 1$ (power of the pheromone); β, $\beta = 1$ (distance/edge weight) and $\rho = 0.1$ (the evaporation factor). Figure 1 and Table 2 present the obtained experimental evaluation results. In the experimental evaluations, it was considered as the intelligence indicator the obtained best-to-date travel value from the end of the problem-solving.

In Table 2, symbol "#" indicates an intelligence indicator value that is not identified as an outlier, but it is statistically further from the rest. It can be noticed that no outlier values were detected. The superscripts indices indicate at which application of the outliers' detection test those value is identified as further from the rest. It can be noticed that the outlier detection test was applied two times on *IntA* (at the second application it does not detect any value further from the rest), and recursively three times on *IntB* (at the third application it does not detect any value further from the rest). Table 2

The first two columns of Table 3, labeled as "IntA" and "IntB," present the results of the *Normality Verification and Extraction Algorithm*. The CIT is calculated as the *mean* of intelligence indicators. *SD* denotes the standard deviation.

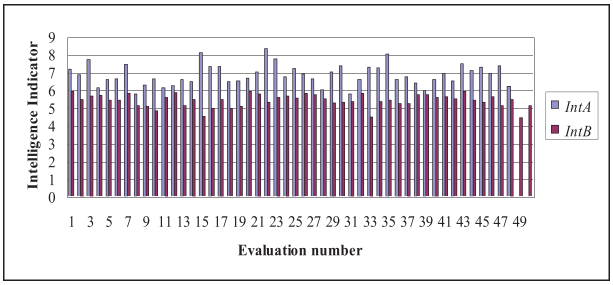

Figure 1 Graphical representation of *IntA* and *IntB*.

Table 2 *IntA* and *IntB* problem-solving intelligence evaluations results.

IntA				*IntB*			
7.172;	6.864;	7.691;	6.12;	5.932;	5.454;	5.657;	5.706;
6.572;	6.612;	7.413;	5.786;	5.409;	5.442;	5.826;	5.123;
6.262;	6.626;	6.13;	6.217;	5.065;	4.81;	5.579;	5.853;
6.586;	6.467;	8.084;	7.313;	5.121;	5.459;	4.492;	4.944;
7.295;	6.473;	6.516;	6.657;	5.466;	4.978;	5.095;	5.917;
7.009;	$8.297^{\#1}$;		7.714;	5.76;	5.315;	5.558;	5.661;
6.729;	7.177;	6.887;	6.612;	5.546;	5.809;	5.729;	5.519;
5.99;	7.007;	7.333;	5.78;	5.288;	5.293;	5.365;	5.806;
6.585;	7.257;	7.225;	8.005;	$4.465^{\#2}$;	5.345;	5.427;	5.217;
6.592;	6.741;	6.37;	5.944;	5.244;	5.741;	5.724;	5.579;
6.573;	6.911;	6.513;	7.447;	5.599;	5.506;	5.907;	5.421;
7.066;	7.277;	6.924;	7.343;	5.312;	5.632;	5.101;	5.476;
6.204				$4.405^{\#1}$;	5.111		

Table 3 Results of VerExtr algorithm.

	First Application		*Second Application*	
	IntA	***IntB***	***IntA****	***IntB****
CentrInd	6.841	5.40378	6.81	5.444
[*LCIm, HCIm*]	[6.671, 7.011]	[5.3, 5.508]	[6.648, 6.972]	[5.354, 5.535]
INVCentrInd	0.1462	0.1851	0.1468	0.1837
SD	0.5861	0.3647	0.5515	0.3115
Sample size	48	50	47	48
CV/homogeneity	8.567/hom.	6.749/hom.	8.1/hom.	5.72 CV < 10/hom.

CV, Coefficient of Variation; SD, Standard Deviation.

As a first approach for the normality verification, the K–S *test* of normality was applied, at $\alpha N = 0.05$ significance level. The obtained results, presented in Table 4 shows that both samples, *IntA* and *IntB*, passed the normality assumption. Based on this fact, the application of a transformation or opting for the elimination of outlier values is not required.

Furthermore, the forthcoming processing and analysis of the *Verify Evolution in Intelligence by Learning* algorithm was applied. For the verification of statistical equality of standard deviations SD_A and SD_B, the *F-test* was applied. The calculation results where $F = 2.583$ and $Pfval = 0.0013$ (*P-value* of the *F-test*). The obtained result, $Pfval \leq \alpha F$ with significance level $\alpha F = 0.05$, proves that the difference between the two considered SDs is statistically significant. Based on this result, according to *Step 3* of the *Verify Evolution in Intelligence by Learning* algorithm, it can be deduced that it should be applied the *Welch Corrected Unpaired Two-Sample T-test*. By applying the *Welch's Corrected Unpaired Two-Sample T-test* we obtained $T = 14.507$ and $Pval \approx 0.0001$. Based on the fact that the $Pval \leq \alpha Met$, it can be concluded that the ILS^{BL} intelligence is changed, it performed an evolutionary or involutionary step in intelligence. We calculated the $INVCentrInd = 1/CentrInd$, based on the fact that in the experimental study it was considered as intelligence indicator the global-best found solution. Smaller global-best value is better, suggests higher intelligence than a higher global-best. Given that $INVCentrIndA < INVCentrIndB$, based on Step 4 of the *II-Learn* metric algorithm, it can be deduced that the intelligence of ILS^{BL} evolved, in intelligence, by making an evolutionary step in intelligence.

The *CL* is established as 95%. *LCIm, HCIm* denotes the low and high bounds of the confidence interval of the mean.

In order to obtain a precise conclusion related to the fact that ILS^{BL} has made an evolutionary step, as a second analysis of the normality using the *Shapiro–Wilk test*. By analyzing the results of the *Shapiro–Wilk test* for *IntA* and *IntB* presented in Table 5, can be formulated the conclusion that the data did not pass the normality assumption. For an alternative visual analysis, we created the histograms corresponding to *IntA*, (Figure 2) and *IntB* (Figure 3), and *Q-Q Plots* corresponding to *IntA* (Figure 4) and *IntB* (Figure 5). The provided visual representation suggests also the violation of the normality assumption.

To obtain normally distributed data, we opted for the application of outlier intelligence detection using the Grubbs test. The

Table 4 Results of K–S test applied to *IntA, IntB.*

	IntA	***IntB***
K–S statistic	0.1023	0.1057
Pn *(P-value of normality test)*	>0.10	>0.10
Normality passed ($Pn > \alpha N$)	Yes	Yes

K–S, Kolmogorov–Smirnov.

Table 5 Results of *S–W test* for *IntA* and *IntB.*

	IntA	***IntB***
S–W statistic	0.977	0.931
Pn *(P-value of normality test)*	0.457	0.006
Normality passed ($Pn > \alpha N$)	No	No

S–W test, Shapiro–Wilk test.

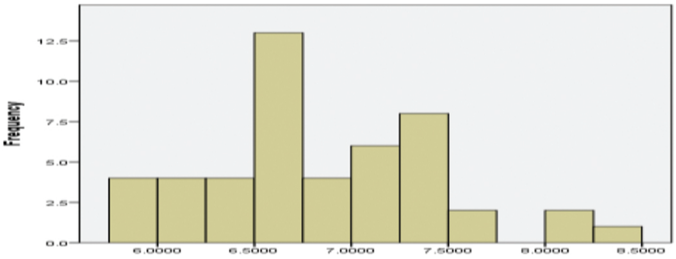

Figure 2 Histogram of the *IntA* data for the visual analysis of the normality.

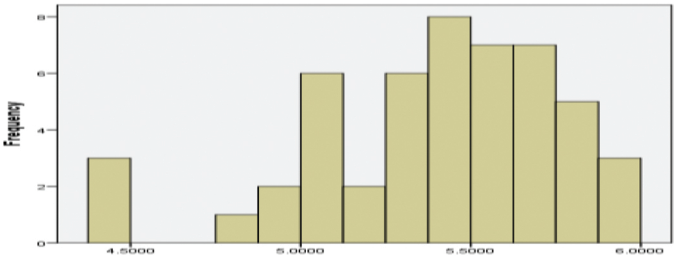

Figure 3 Histogram of the *IntB* data for the visual analysis of the normality.

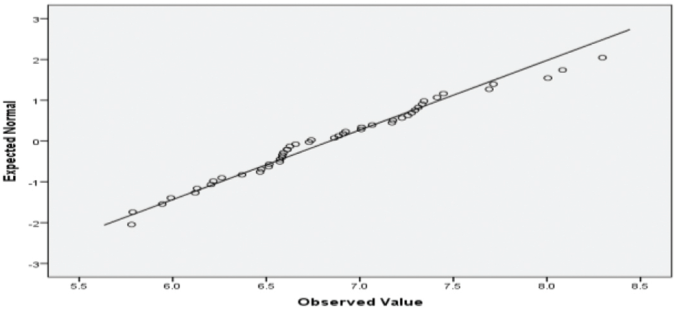

Figure 4 Normal Quantile–Quantile (Q–Q) Plot of *IntA*.

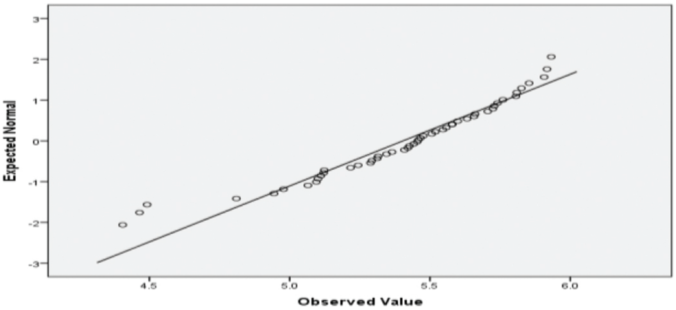

Figure 5 Normal Quantile–Quantile (Q–Q) Plot *IntB*.

decision was made also taking into consideration the expectable data normality by analyzing the drawn Q–Q plots. The *Grubbs test* was applied at a significance level of $aG = 0.05$. The two-sided *Grubbs* test was applied in order to detect at the same time low and high outlier values and low and high values that are not outliers but are statistically further from the rest.

Table 6 Results of normality tests applied to *IntA** and *IntB**.

	IntA*	***IntB****	
K–S statistic	0.09878	0.08	**K–S test**
Pn *(calculated P-value)*	>0.01	>0.1	
Normality passed $(Pn > \alpha N)$	Yes	Yes	
S–W statistic	0.98	0.965	**S–W test**
Pn *(calculated P-value)*	0.609	0.161	
Normality passed $(Pn > \alpha N)$	Yes	Yes	

K–S, Kolmogorov–Smirnov, S–W test, Shapiro–Wilk.

In the case of *IntA*, at the first application of the outlier detection test, outliers were not identified, only the value 8.297 that was identified as further from the rest (calculated $Z = 2.484$ and the *Critical Value Of Z*: 3.112). The decision was the elimination of this value, $IntA^* = IntA - \{8.297\}$. The second column of Table 6 labeled "ILS^{BL}/IntA*" presents the results obtained by analyzing *IntA**. It can be noticed that all the normality tests were passed.

In the case of *IntB* at the first application of the outlier detection test, outliers were not identified, only the value 4.405 that was identified as further from the rest (the calculated $Z = 2.739$ and the *Critical Value Of Z*: 3.128). Thus, we decided on the elimination of this value, $IntB^* = IntB - \{4.405\}$. The obtained result of the *Shapiro–Wilk test* indicated that *IntB** did not pass the normality assumption. We applied the outlier intelligence detection test again. No outliers were identified, only a value 4.465 that was further from the rest (the calculated $Z = 2.834$ and the *Critical Value Of Z*: 3.12). Thus we decided to eliminate this value, $IntB^* = IntB - \{4.465\}$. Table 6 shows that in the case of *IntB** all the normality tests were passed. The second column of Table 6 presents the results obtained by analyzing *IntB**.

The final obtained *IntA** and *IntB** passed the normality assumption according to K–S and S–W tests, thus allowing the application of the specific data analysis in order of verification if the studied system performed an evolutionary step in intelligence.

For the verification of statistical equality of standard deviations $SD_A{}^*$ and $SD_B{}^*$, the *F-test* was applied. The obtained calculation results where $F = 3.134$ and $Pfval = 0.0002$ (*P-value* of the *F-test*). The obtained *Pfval*, $Pfval \leq \alpha F$ with $\alpha F = 0.05$, proves that the difference between the SD of $SD_A{}^*$ and $SD_B{}^*$, is statistically significant. Based on this result, according to *Step 2* of the *Verify Evolution in Intelligence by Learning* algorithm, it can be deduced that the *Welch's Corrected Unpaired Two-Sample T-test* should be applied. Following this, $T = 14.822$, and $Pval < 0.0001$ were obtained. Based on the fact that $Pval \leq \alpha Mmet$, it can be concluded that the ILS^{BL} intelligence is changed significantly. This proves that ILS^{BL} made an evolutionary step in intelligence or involutionary step in intelligence. Based on the fact that $INVCentrIndA < INVCentrIndB$, it can be concluded that an evolutionary step in intelligence was made.

Summarizing, in the performed experimental study, first we applied the K–S test. The data passed the normality test, and based on this consideration we used the proposed metric for measuring the machine intelligence of the studied learning system. At a second approach, we applied the *S–W test* that is considered more powerful than the *K–S test*. None of *IntA* and *IntB* passed the normality assumption by using the *Shapiro–Wilk* test. Based on this fact it was opted for the elimination of outliers, finally obtaining normally distributed intelligence indicator data, followed by the application of the metric. The conclusion in both cases with the results presented in the previous paragraphs, was the same, that LS^{BL} made a measurable evolutionary step in intelligence.

5. DISCUSSION

Intelligent CMASs are challenged to solve difficult practical problems in an efficient and effective way. In our study, we consider that machine intelligence measuring based on the ability of a CMAS to solve difficult problems. If learning results in a statistically significant improvement of machine intelligence then this improvement should be measurable by a metric. There are very few designed metrics that can be applied for measuring machine intelligence. Usual drawbacks of such metrics include limitations in universality, accuracy and robustness. Another important aspect that a metric should treat consists in the variability of intelligence.

Considering the previously mentioned limitations, it was proposed a novel accurate and robust metric called *II-Learn* (*metric for measuring Intelligence Increase of artificial Learning systems*) for measuring the increase of intelligence of a CMAS after a learning process.

The intelligence measuring criteria in our approach is based on some kind of difficult problem-solving ability. The fact that a specific type of problem could be solved with more or less intelligence by different CMASs is well-known. In case of a CMAS if there are some changes (changing of the problem-solving knowledge or details detained by a problem whose solving is in progress), which could be the result, for example, of performed autonomous learning, the problem-solving ability could change also (it could increases, remains unmodified or it could even decreases).

In a previous study, the *MetrIntComp* metric [41] with the purpose of measurement of machine intelligence of CMASs is proposed. The intelligence measuring criteria was also based on the principle of difficult problem-solving ability. *MetrIntComp* was designed to be robust, which is based on the fact that in the process of classification it uses the *Two-Sample Unpaired Mann–Whitney test* that is known as a non-parametric robust test [116]. Since both metrics quantify the problem-solving intelligence of learning systems, both *II-Learn* and *MetrIntComp* can be considered comparable.

The novelty of *II-Learn versus MetrIntComp* is based on different performed processing and analyses. One of the significant differences from the performed computations point of view between the two metrics is that *II-Learn* uses *Two-Sample Unpaired T-test* [88–90] and *Welch Two-Sample Unpaired T-test* (*Welch's test*) [91,92] for intelligence comparison as the parametric analog of the *Mann–Whitney test. Two-Sample Unpaired T-test* is appropriate in the case of equality from the statistical point of view between the SDs of the two intelligence indicators samples. One representing the problem-solving intelligence evaluation result of the system before learning, the other one representing the problem-solving intelligence evaluation result after the learning. The *Welch's test* [90] is more appropriate when the SDs are not equal from the statistical point of view. *Welch's test* represents a generalization of the *Two-Sample Unpaired T-test,* in the sense that it can be used in case of unequal SDs [91]. In [92] the *Welch's test* was proved more reliable than *Two-Sample Unpaired T-test* when the two samples have unequal SDs and unequal sample sizes. The statistical power of *Welch's t-test* is very close to that of *Two-Sample Unpaired T-test*, when the population SDs are equal and sample sizes are balanced [92]. In [117] a generalization to the *Welch's t-test* was presented, in order to be applied to any number of samples (even more than two samples). That study showed that this generalization is more robust than *One-Way Analysis of Variance* (ANOVA) that can be applied to any number of samples; however, it requires the assumption of normality and equality of sample sizes.

The main advantage of *II-Learn* over the *MetrIntComp* is its accuracy for smaller sample sizes. It takes into consideration the sample intelligence indicator data property to be normally distributed. Based on this fact, a mathematically grounded calculus is applied for verification if the studied CMAS made an evolutionary step, which is the most appropriate for normally distributed data. In the case of non-Gaussian intelligence indicator data, a transformation should be applied (see Table 1), resulting in enhanced robustness. Another advantage in the calculation of *II-Learn* over *MetrIntComp* is that it considers the presence of possible extremes and applies a methodology to remove them from the samples.

Similarly to the humans, intelligent computing systems have a variability of intelligence in problem-solving. The elaborated *II-Learn* metric takes into consideration this variability in problem-solving intelligence. In a specific situation, the studied system's reaction could be more or less intelligent. As a measure of the CIT, we established as the most representative the mean based on the consideration that the data were sampled from a Gaussian population.

The Central Limit Theorem can be enounced as follows [118]: given independent random samples of M observations, the distribution of sample means approaches normality as the size of M increases, regardless of the shape of the population distribution. In general, many studies consider that M should be at least 30 or higher. In [119] it is proved that in the case of samples consisting of hundreds of data, the distribution of the data can be ignored. In our case study making hundreds of problem-solving intelligence, evaluations could be very expensive or even impossible. As a good practice, we recommend using intelligence indicators sets larger than 30. Thus, we suggest the application of *II-Learn* metric with at least 30 experimental evaluation of the learning system before the learning process and with at least 30 experimental evaluations after the learning process.

The number of necessary experimental evaluations is also an important issue that should be analyzed in order to formulate scientifically correct conclusions. Conclusions formulated based on too few experimental evaluations could be inaccurate or even incorrect. Sometimes the realization of an experiment could be expensive or time-consuming, thus suggesting the limitation of the number of experiments. Based on this aspect we consider that, in some cases, a compromise must be made with respect to the number of experiments that will be realized, and the chance of an error occurrence (in our approach the probability of occurrence of a type I error or a type II error). In this paper, we proposed a calculus for the exact estimation of the number of experimental evaluations.

Different metrics presented in the literature like those based on analytical models are limited in universality. *II-Learn* metric is universal, it can be used for intelligent learning systems generally; it does not depend on aspects like the intelligent system's architecture. *II-Learn* can be applied even in the case of individual learning agents. The universality is assured based on our specific approach on that a system's intelligence is evaluated on some difficult problems-solving evaluations before a learning process and problem-solving intelligence evaluations after a learning process. On these two obtained intelligence indicators data sets is applied some specific calculus in order to verify if the intelligence as result of learning is changed, and if this has as an effect that the system has made an evolutionary or involutionary step in intelligence. The method of the metric provides both accuracy and robustness, addressing the variability in problem-solving intelligence.

We would like to mention that the significance that we give to the notion universality is different from the significance given to universality in some other studies, like that presented by Hernandez-Orallo and Dowe [60]. The researchers proposed the idea of universal anytime intelligence test able to measure any kind of natural/biological and AI. In our approach, the notion of universality in intelligence measuring is considered independent of the studied system architecture. We do not intend to build metrics able to measure both biological (including the human) and AI.

6. CONCLUSIONS

In this paper, we proposed a novel metric called *II-Learn*, for measuring a cooperative multiagent system (able to learn) intelligence, and the verification if the learning resulted in an intelligence level/measure modification by statistically significant increasing or decreasing. *II-Learn* metric takes into consideration the variability in the problem-solving intelligence (lower and higher intelligence in different situations). Advantages of the *II-Learn* metric consist in universality, accuracy and robustness.

The proposed *II-Learn* metric was compared with the state-of-the-art *MetrIntComp* metric that was also designed to be robust. For the *II-Learn* metric robustness increase we proposed solutions based on the experimental intelligence evaluations results transformation and detection of experimentally obtained outlier intelligence values. The main advantage of *II-Learn* over the *MetrIntComp* is its increased accuracy. Another advantage of *II-Learn* over *MetrIntComp* is that it considers the presence of possible outliers and applies a methodology to remove them from the measured problem-solving intelligence samples.

For proving the effectiveness of the metric, we performed a case study, which showed that the learning process had as an effect the making of an evolutionary step in the intelligence (the studied system evolved).

II-Learn is an original metric, and it will represent the basis for the intelligence measurement of learning systems, and intelligence increase based on learning, in many future researches related to different intelligent systems, which operate individually or cooperate with each other. Examples of practical applications, may include intelligent robotic transportation systems; swarm of robots performing different tasks in different environments; agents specialized in solving different tasks in the healthcare, agents and cooperative multiagent systems able to solve different tasks in transportation.

ACKNOWLEDGMENTS

This work was supported by the project CNFIS-FDI-2019-0453: Support actions for excellence in research, innovation and technological transfer at "Vasile Alecsandri" University of Bacău (ACTIS-Bacău), financed by the National Council for Higher Education, Romania.

REFERENCES

[1] Y. Zhang, S. Liu, Z. Zhu, S. Si, Agent-based intelligent medical diagnosis system for patients, Technol. Health Care. 23 (2015), S397–S410.

[2] L.B. Iantovics, Agent-based medical diagnosis systems, Comput. Informat. 27 (2008), 593–625.

[3] Y. Chen, X. Yue, H. Fujita, S. Fu, Three-way decision support for diagnosis on focal liver lesions, Knowl. Based Syst. 127 (2017), 85–99.

[4] D. Sodkomkham, D. Ciliberti, M.A. Wilson, K.I. Fukui, K. Moriyama, M. Numao, F. Kloosterman, Kernel density compression for real-time Bayesian encoding/decoding of unsorted hippocampal spikes, Knowl. Based Syst. 94 (2016), 1–12.

[5] D.K. Iakovidis, E. Papageorgiou, Intuitionistic fuzzy cognitive maps for medical decision making, IEEE Trans. Inf. Technol. Biomed. 15 (2011), 100–107.

[6] R. Chandwani, R. De, Doctor-patient interaction in telemedicine: logic of choice and logic of care perspectives, Inf. Syst. Front. 19 (2017), 955–968.

[7] S. Raja, M. Jaiganesh, S. Ramaiah, An efficient fuzzy self-classifying clustering based framework for cloud security, Int. J. Comput. Intell. Syst. 10 (2017), 495–506.

[8] G. Kolaczek, K. Juszczyszyn, Attack pattern analysis framework for a multiagent intrusion detection system, Int. J. Comput. Intell. Syst. 1 (2008), 215–224.

[9] X. Li, Z. Dong, H. Wang, Q. Meng, Z. Wang, Y. Zhang, A novel approach to selecting contractor in agent-based multi-sensor battlefield reconnaissance simulation, Int. J. Comput. Intell. Syst. 5 (2012), 985–995.

[10] M. Singh, X. Cheng, R. Belavkin, Local semantic indexing for resource discovery on overlay network using mobile agents, Int. J. Comput. Intell. Syst. 7 (2014), 432–455.

[11] S. Han, H.Y. Youn, Reflecting the perspectives of multiple agents in distributed reasoning for context-aware service, Int. J. Comput. Intell. Syst. 6 (2013), 700–711.

[12] B. Ponte, D. de la Fuente, J. Parreño, R. Pino, Intelligent decision support system for real-time water demand management, Int. J. Comput. Intell. Syst. 9 (2016), 168–183.

[13] L.B. Iantovics, C.B. Zamfirescu, ERMS: an evolutionary reorganizing multiagent system, Int. J. Innov. Comput. Inf. Control. 9 (2013), 1171–1188.

[14] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, second ed., Prentice Hall, Upper Saddle River, 2003.

[15] S. Mascarenhas, N. Degens, A. Paiva, R. Prada, G.J. Hofstede, A. Beulens, R. Aylett, Modeling culture in intelligent virtual agents, Auton. Agents Multi-Agent Syst. 30 (2016), 931–962.

[16] G. Weiss (Eds.), Multiagent systems, second ed., MIT Press, Cambridge, 2013.

[17] D.I. Inan, G. Beydoun, S. Opper, Agent-based knowledge analysis framework in disaster management, Inf. Syst. Front. 20 (2018), 783–802.

[18] M. Kazemifard, N. Ghasem-Aghaee, B.L. Koenig, T.I. Ören, An emotion understanding framework for intelligent agents based on episodic and semantic memories, Auton. Agents Multi-Agent Syst. 28 (2014), 126–153.

[19] P. Herrero, C. Greenhalgh, A. de Antonio, Modelling the sensory abilities of intelligent virtual agents, Auton. Agents Multi-Agent Syst. 11 (2005), 361–385.

[20] K. Dautenhahn, Book Review: Swarm Intelligence by James Kennedy, Russell C. Eberhart, with Yuhui Shi, Genet. Prog. Evolvable Mach. 3 (2002), 93–97.

[21] L.S. Sterling (Ed.) Intelligent Systems, Concepts and Applications, Springer, New York, 1993.

[22] F. Cena, L. Console, A. Matassa, I. Torre, Multi-dimensional intelligence in smart physical objects, Inf. Syst. Front. 21 (2019), 383–404.

[23] C. Zarges, H. Lipson, M. Kurman, Driverless: intelligent cars and the road ahead, Genet. Program. Evol. Mach. 19 (2018), 301–303.

[24] D. Floreano, C. Mattiussi, Bio-Inspired Artificial Intelligence, Theories, Methods, and Technologies, MIT Press, Cambridge, Massachusetts, London, England, 2008.

[25] V.C. Müller, N. Bostrom, Future progress in artificial intelligence: a survey of expert opinion, in: V.C. Müller (Ed.), Fundamental Issues of Artificial Intelligence, Springer, Berlin, 2016, pp. 553–571.

[26] L. Argotte, G. Arroyo-Figueroa, J. Noguez, SI-APRENDE: an intelligent learning system based on SCORM learning objects for training power systems operators, developing concepts in applied intelligence, Stud. Comput. Intell. 363 (2011), 33–38.

[27] I. Zuckerman, S. Kraus, J.S. Rosenschein, Using focal point learning to improve human-machine tacit coordination, Auton. Agents Multi-Agent Syst. 22 (2011), 289–316.

[28] L. Panait, S. Luke, Cooperative multi-agent learning: the state of the art, Auton. Agents Multi-Agent Syst. 11 (2005), 387–434.

[29] A.B. Williams, Learning to share meaning in a multi-agent system, Auton. Agents Multi-Agent Syst. 8 (2004), 165–193.

[30] M.V.N. Prasad, V.R. Lesser, Learning situation-specific coordination in cooperative multi-agent systems, Auton. Agents Multi-Agent Syst. 2 (1999), 173–207.

[31] C. Meek, B. Thiesson, D. Heckerman, The learning-curve sampling method applied to model-based clustering, J. Mach. Learn. Res. 2 (2002), 397–418.

[32] A.L. Madsen, F. Jensen, A. Salmerón, H. Langseth, T.D. Nielsen, A parallel algorithm for Bayesian network structure learning from large data sets, Knowl. Based Syst. 117 (2017), 46–55.

[33] M. Liu, W. Pan, M. Liu, Y. Chen, X. Peng, Z. Ming, Mixed similarity learning for recommendation with implicit feedback, Knowl. Based Syst. 119 (2017), 178–185.

[34] D. Yu, N. Chen, F. Jiang, B. Fu, A. Qin, Constrained NMF-based semi-supervised learning for social media spammer detection, Knowl. Based Syst. 125 (2017), 64–73.

[35] T. Jaakkola, S.P. Singh, M.I. Jordan, Reinforcement learning algorithm for partially observable Markov decision problems, in: Advances in Neural Information Processing Systems 7, NIPS'94, MIT Press, Cambridge, 1995, pp. 345–352.

[36] L.P. Kaelbling, M.L. Littman, A.W. Moore, Reinforcement learning: a survey, J. Artif. Intell. Res. 4 (1996), 237–285.

[37] R. Sutton, A. Barto, Reinforcement Learning, An Introduction, MIT Press, Cambridge, 1998.

[38] C. Sammut, G.I. Webb, (Eds.) Encyclopedia of Machine Learning and Data Mining, Springer Science & Business Media, New York, 2017.

[39] B.G. Buchanan, E.H. Shortliffe, Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, 1984.

[40] L. Orseau, Teleporting universal intelligent agents, international conference on artificial general intelligence, AGI 2014: artificial general intelligence, LNCS. 8598 (2014), 109–120.

[41] L.B. Iantovics, C. Rotar, E. Nechita, A novel robust metric for comparing the intelligence of two cooperative multiagent systems, Procedia. Comp. Sci. 96 (2016), 637–644.

[42] G.C. Crisan, C.M. Pintea, V. Palade, Emergency management using geographic information systems, application to the first romanian traveling salesman problem instance, Knowl. Inf. Syst. 50 (2017), 265–285.

[43] C. Rego, D. Gamboa, F. Glover, C. Osterman, Traveling salesman problem heuristics: leading methods, implementations and latest advances, Eur. J. Oper. Res. 211 (2011), 427–441.

[44] H. Shirado, N.A. Christakis, Locally noisy autonomous agents improve global human coordination in network experiments, Nature. 545 (2017), 370–374.

[45] E. Adamey, A.E. Oğuz, Ü. Özgüner, Collaborative multi-MSA multi-target tracking and surveillance: a divide & conquer method using region allocation trees, J. Intell. Robot. Syst. 87 (2017), 471–485.

[46] A.J.C. Sharkey, Robots, insects and swarm intelligence, Artif. Int. Rev. 26 (2006), 255–268.

[47] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. Part B. 26 (1996), 29–41.

[48] K. Yang, A. Galis, X. Guo, D. Liu,Rule-driven mobile intelligent agents for real-time configuration of IP networks, in: V. Palade *et al.* (Eds.), International Conference on Knowledge-Based Intelligent Information and Engineering Systems, LNCS, 2003, pp. 921–928.

[49] A.I. Kadhim, Survey on supervised machine learning techniques for automatic text classification, Artif. Intell. Rev. 52 (2019), 273–292.

[50] H. Ghorbel, N. Zannini, S. Cherif, F. Sauser, D. Grunenwald, W. Droz, M. Baradji, D. Lakehal, Smart adaptive run parameterization (SArP): enhancement of user manual selection of running parameters in fluid dynamic simulations using bio-inspired and machine-learning techniques, Soft Comput. 23 (2019), 12031–12047.

[51] H. Zhou, K. Chang, H.X. Bai, B. Xiao, C. Su, W.L. Bi, P.J. Zhang, J.T. Senders, M. Vallières, V.K. Kavouridis, A. Boaro, O. Arnaout, L. Yang, R.Y. Huang, Machine learning reveals multimodal MRI patterns predictive of isocitrate dehydrogenase and 1p/19q status in diffuse low- and high-grade gliomas, J. Neurooncol. 142 (2019), 299–307.

[52] T. Dreossi, A. Donzé, S.A. Seshia, Compositional falsification of cyber-physical systems with machine learning components, J. Autom. Reason. 63 (2019), 1031–1053.

[53] Y.S. Kim, T.B. Yoon, H.J. Cha, Y.M. Jung, E. Wang, J.H. Lee, A outliers analysis of learner's data based on user interface behaviors, in Proceedings of the Seventh IEEE International

Conference on Advanced Learning Technologies, ICALT 2007, IEEE Computer Society Press, Niigata, 2007.

[54] A.M. Turing, Computing machinery and intelligence, Mind. 59 (1950), 433–460.

[55] M. Newborn, Kasparov vs. DEEP BLUE: Computer Chess Comes of Age, Springer-Verlag, New York, 1997.

[56] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, E.T. Mueller, Watson: beyond jeopardy, Artif. Intell. 199–200 (2013), 93–105.

[57] K. Schreiner, Measuring IS: toward a US standard, IEEE Intell. Syst. Appl. 15 (2000), 19–21.

[58] S. Legg, M. Hutter, Universal intelligence: a definition of machine intelligence, Minds Mach. 17 (2007), 391–444.

[59] B. Hibbard, Measuring agent intelligence via hierarchies of environments, in: J. Schmidhuber, K.R. Thórisson, M. Looks (Eds.), Artificial General Intelligence, LNCS, vol. 6830, Springer, Berlin, Heidelberg, 2011, pp. 303–308.

[60] J. Hernandez-Orallo, D.L. Dowe, Measuring universal intelligence: towards an anytime intelligence test, Artif. Intell. 174 (2010), 1508–1539.

[61] A. Anthon, T.C. Jannett, Measuring machine intelligence of an agent-based distributed sensor network system, in: K. Elleithy (Eds.), Advances and Innovations in Systems, Computing Sciences and Software Engineering, Springer, Dordrecht, 2007, pp. 531–535.

[62] Z. Winklerova, Maturity of the particle swarm as a metric for measuring the collective intelligence of the swarm, in: Y. Tan, Y. Shi, H. Mo (Eds.), Advances in Swarm Intelligence, LNCS, vol. 7928, Springer, Berlin, Heidelberg, 2013, pp. 40–54.

[63] D. Franklin, A. Abrao, Measuring software agent's intelligence, in Proceeding International Conference: Advances in Infrastructure for Electronical Business, Science and Education on the Internet, L'Aquila, 2000.

[64] H. Gardner, Multiple Intelligences: The Theory in Practice, A Reader, Basic Books, New York, 1993.

[65] L.B. Iantovics, C. Rotar, M.A. Niazi, MetrIntPair - a novel accurate metric for the comparison of two cooperative multiagent systems intelligence based on paired intelligence measurements, Int. J. Intell. Syst. 33 (2018), 463–486.

[66] L.B. Iantovics, F. Emmert-Streib, S. Arik, MetrIntMeas a novel metric for measuring the intelligence of a swarm of cooperating agents, Cogn. Syst. Res. 45 (2017), 17–29.

[67] S. Arik, L.B. Iantovics, S.M. Szilagyi, OutIntSys - a novel method for the detection of the most intelligent cooperative multiagent systems, in: D. Liu, S. Xie, Y. Li, D. Zhao, E.S. El-Alfy (Eds.), Neural Information Processing, Lecture Notes in Computer Science, vol. 10637, Springer, Cham, 2017, pp. 31–40.

[68] E.R. Messina, A.M. Meystel (Eds.), Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 PerMIS Workshop August 13–15, 2002, National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce, 2002.

[69] A.M. Meystel, E.R. Messina (Eds.), Measuring the Performance and Intelligence of Systems: Proceedings of the PerMIS W. August 14–16, 2000, National Institute of Standards and Technology, U.S. Department, 2000.

[70] W.B. Rouse, H. Sandra, Rouse analysis and classification of human error, IEEE Trans. Syst. Man Cybern. 13 (1983), 539–549.

[71] I.I. Bejar, S.J. Whalen, Methods and systems for presentation and evaluation of constructed responses assessed by human evaluators, US Patent. 6,526,258, (2003).

[72] C. Munteanu, A. Rosa, Gray-scale image enhancement as an automatic process driven by evolution, IEEE Trans. Syst. Man Cybern. Part B. 34 (2004), 1292–1298.

[73] J. Jussila, V. Vuori, J. Okkonen, N. Helander, Reliability and perceived value of sentiment analysis for Twitter data, in: A. Kavoura, D. Sakas, P. Tomaras (Eds.), Strategic Innovative Marketing, Springer Proceedings in Business and Economics, Springer, Cham, 2017, pp. 43–48.

[74] K. Siorpaes, E. Simperl, Human intelligence in the process of semantic content creation, World Wide Web. 13 (2010), 33–59.

[75] E. Lughofer, C. Cernuda, S. Kindermann, M. Pratama, Generalized smart evolving fuzzy systems. Evol. Syst. 6 (2015), 269–292.

[76] W. Liang, Y. Hu, N.K. Kasabov, Evolving personalized modeling system for integrated feature, neighborhood and parameter optimization utilizing gravitational search algorithm, Evol. Syst. 6 (2015), 1–14.

[77] L. Altenberg, Evolvability and robustness in artificial evolving systems: three perturbations, Genet. Prog. Evol. Mach. 15 (2014), 275–280.

[78] J.M. Bland, D.G. Altman, Statistics notes: measurement error, BMJ. 312 (1996), 1654.

[79] I.M. Chakravarti, R.G. Laha, J. Roy, Handbook of Methods of Applied Statistics, vol. I, John Wiley and Sons, Hoboken, 1967, pp. 392–394.

[80] H. Lilliefors, On the Kolmogorov-Smirnov test for normality with mean and variance unknown, J. Am. Stat. Assoc. 62 (1967), 399–402.

[81] H. Lilliefors, On the Kolmogorov-Smirnov test for the exponential distribution with mean unknown, J. Am. Stat. Assoc. 698 (1969), 387–389.

[82] N. Razali, Y.B. Wah, Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests, J. Stat. Model. Anal. 2 (2011), 21–33.

[83] M.A. Stephens, EDF statistics for goodness of fit and some comparisons, J. Am. Stat. Assoc. 69 (1974), 730–737.

[84] H. Motulsky, GraphPad InStat version 3, The InStat Guide to Choosing and Interpreting Statistical Tests, GraphPad Software Inc., San Diego, 2003.

[85] V. Barnett, T. Lewis, Outliers in Statistical Data, third ed., Wiley, Chichester, 1994.

[86] F.E. Grubbs, Sample criteria for testing outlying observations, Ann. Math. Stat. 21 (1950), 27–58.

[87] D.S. Moore, G.P. McCabe, Introduction to the Practice of Statistics, second ed., W.H. Freeman and Company, New York, 1993.

[88] D.G. Altman, Practical Statistics for Medical Research, Chapman and Hall, London, 1991.

[89] P. Armitage, G. Berry, Statistical Methods in Medical Research, third ed., Blackwell, England, Oxford, 1994.

[90] M., Marusteri, V. Bacarea, Comparing groups for statistical differences: how to choose the right statistical test?, Biochem. Medica. 20 (2010), 15–32.

[91] B.L. Welch, The generalization of Student's problem when several different population variances are involved, Biometrika. 34 (1947), 28–35.

[92] G.D. Ruxton, The unequal variance t-test is an underused alternative to Student's t-test and the Mann-Whitney U test, Behav. Ecol. 17 (2006), 688–690.

[93] C.A. Markowski, E.P. Markowski, Conditions for the Effectiveness of a Preliminary Test of Variance, Am. Stat. 44 (1990), 322–326.

[94] J. Cohen, Statistical Power Analysis for the Behavioral Sciences, Lawrence Erlbaum Associate, Hillsdale, New Jersey, 1988.

[95] J.B. Cunningham, E. McCrum-Gardner, Power, effect and sample size using GPower: practical issues for researchers and members of research ethics committees, Evid. Based Midwifery. 5 (2007), 132–136.

[96] K. Kelley, K.J. Preacher, On effect size, Psychol. Methods. 17 (2012), 137–152.

[97] T. Stutzle, H.H. Hoos, The MAX-MIN ant system and local search for the traveling salesman problem, in Proceeding of ICEC97, IEEE Press,Piscataway, 1997, pp. 309–314.

[98] B.R. Johnson, M.L. Borowiec, J.C. Chiu, E.K. Lee, J. Atallah, P.S. Ward, Phylogenomics resolves evolutionary relationships among ants, bees, and wasps, Curr. Biol. 23 (2013), 2058–2062.

[99] A. Colorni, M. Dorigo, V. Maniezzo, Distributed Optimization by Ant Colonies, Actes de la première conférence européenne sur la vie artificielle, Elsevier, Paris, 1991, pp. 134–142.

[100] M. Dorigo, Optimization, Learning and Natural Algorithms, Ph.D thesis, Politecnico di Milano, Italy, 1992.

[101] D.L. Poole, A.K. Mackworth, Artificial Intelligence, Foundations of Computational Agents, Cambridge University Press, Cambridge, 2010.

[102] L.M. Gambardella, M. Dorigo, Ant-Q: a reinforcement learning approach to the traveling salesman problem, in Proceeding Twelfth International Conference on Machine Learning (ML-95), Morgan Kaufmann, 1995, pp. 252–260.

[103] M. Dorigo, L.M. Gambardella, A study of some properties of Ant-Q, in Proceeding of PPSN-IV, 1996, pp. 656–665.

[104] C. Comtois, B. Slack, J.P. Rodrigue, The Geography of Transport Systems, third ed., Taylor & Francis, Routledge, London, 2013.

[105] G. Hasle, K.A. Lie, E. Quak, O. Kloster, (Ed.), Geometric Modelling, Numerical Simulation, and Optimization Applied Mathematics at SINTEF, Springer-Verlag, Berlin, 2007.

[106] G.B. Dantzig, J.H. Ramser, The truck dispatching problem, Manage. Sci. 6 (1959), 80–91.

[107] J.C. Créput, A. Hajjam, A. Koukam, O. Kuhn, Dynamic vehicle routing problem for medical emergency management, in: J.I. Mwasiagi (Ed.), Self Organizing Maps - Applications and Novel Algorithm Design, INTECH, 2011, pp. 233–250.

[108] J.Y. Luo, J.Y. Wang, H. Yu, A dynamic vehicle routing problem for medical supplies in large-scale emergencies, in 6th IEEE Joint International Conference on Information Technology and Artificial Intelligence (ITAIC 2011), Chongqing, 2011.

[109] O. Cordon, I.F. de Viana, F. Herrera, Analysis of the best-worst ant system and its variants on the QAP, in: M. Dorigo *et al.* (Eds.), Ant Algorithms, LNCS, vol. 2463, Springer, Berlin, Heidelberg, 2002, pp. 228–234.

[110] Y. Zhang, H. Wang, Y. Zhang, Y. Chen, BEST-WORST ant system, in Proceeding of the 3rd International Conference on Advanced Computer Control, IEEE CS Press, Harbin, 2011, pp. 392–395.

[111] A. Prakasam, N. Savarimuthu, Metaheuristic algorithms and probabilistic behaviour:a comprehensive analysis of ant colony optimization and its variants, Artif. Int. Rev. 45 (2016), 97–130.

[112] T. Stützle, H.H. Hoos, MAX-MIN ant system, Future Gener. Comput. Syst. 16 (2000), 889–914.

[113] A. Holzinger, M. Plass, K. Holzinger, G. Crisan, C. Pintea, V. Palade, Towards interactive machine learning: applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach, in IFIP International Cross Domain Conference, Springer, Cham, 2016, pp. 81–95.

[114] V. Borsani, A. Matta, G. Beschi, F. Sommaruga, A home care scheduling model for human resources, in 2006 International Conference on Service Systems and Service Management, IEEE Computational Society Press, Troyes, 2006.

[115] Y. Kergosien, C. Lenté, J.C. Billaut, Home health care problem an extended multiple traveling salesman problem, in Proceedings of the Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009), Dublin, 2009, pp. 85–92.

[116] M.P. Fay, M.A. Proschan, Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules, Stat. Surv. 4 (2010), 1–39.

[117] B.L. Welch, On the comparison of several mean values: an alternative approach, Biometrika. 38 (1951), 330–336.

[118] J. Avigad, J. Hölzl, L. Serafin, A formally verified proof of the central limit theorem, J. Autom. Reason. 59 (2017), 389–423.

[119] D.G. Altman, J.M. Bland, Statistics notes: the normal distribution, BMJ. 310 (1995), 298.