14th International Scientific Conference
Science and Higher Education in Function of Sustainable Development – SED 2025

HYPERBOLIC REPRESENTATION OF THE METHODOLOGICAL PROCEDURES OF MACHINE LEARNING MODELING IN SOLVING INVERSE TRAFFIC INSPECTION PROBLEMS

Milorad K. Banjanin[1], PhD; Mirko Stojčić[2], PhD; Milan Vasiljević[2], MSc; Aleksandar Stjepanović[2], PhD

[1]University of East Sarajevo, Department of Computer Science and Systems; University of Novi Sad, Department of Traffic, Serbia; email: banjanin@uns.ac.rs; milorad.banjanin@ffuis.edu.ba
[2]University of East Sarajevo, Faculty of Transport and Traffic Engineering Doboj, Department of Communication and Information systems, BiH, Republic of Srpska, mirko.stojcic@sf.ues.rs.ba, milan.vasiljevic@sf.ues.rs.ba, aleksandar.stjepanovic@sf.ues.rs.ba

Abstract: *The paper presents the methodological procedure in a substantive textual and graphic structure. The purpose of the presentation of the content textual structure is to explain the logical methodological procedure of machine learning modeling in fourteen points. Seven parallel steps each are oriented on the left ascending branch (from 1 to 7 steps) and on the right descending branch of the hyperbola (from 8 to 14 steps). For the representation of the graphic structure, seven basic characteristics of hyperbola are also used with the aim of visualizing creative thoughts in the thematic article in an original way. An applied example of the hyperbolic representation of the methodological procedures of machine learning modeling is used in solving inverse traffic inspection problems. The collection of research data is done in a real geo-area and is used to solve inverse problems of the first kind. The created machine learning models are oriented towards solving inverse problems of a second kind in traffic inspection jobs.*

Keywords*: hyperbolic representation, machine learning, Q-learning, deep learning, inverse problems, traffic inspection, agent decision optimizations*

1. INTRODUCTION

The topic of this paper is oriented toward the conceptual analysis of a selected number of compatible research entities through which a methodological procedure is originally designed for the creation of a hyperbolic representation based on the laws of learning and the modeling of solutions to inverse problems of traffic inspection (SbIn). SbIn is defined as "a specific module of the traffic system in which simple, complicated, and complex tasks of managing sustainable traffic safety within the geo-space of the Republic of Srpska (RS), BiH, are resolved"[1]. In the hyperbolic representation, seven research entities were selected from a set of textually compatible concepts or constructs with descriptions and objectives for generating analytical categories of business problem solutions or scientific research models. In the methodological procedure, they are visually and symmetrically arranged along the two branches of a hyperbola, as a geometric shape represented by "*a set of points in a plane with the property that the difference of distances from any point to two given points is a constant number*"[2].

The first seven compatible research entities are: conceptualization of the scientific research topic, development of the methodology, collection and connection of resources, implementation of formal analyses, model and software design, functional development of processes and actions, and compositional blueprint of the work structure. These are visually paired with seven notation points of the hyperbola, on its left ascending branch in an orientation from bottom to top. Symmetrically, on the right branch of the hyperbola, in an orientation from top to bottom, the next seven concepts are paired and arranged: final structuring of the paper's content, reviewing and editing of the paper's text, control supervision over the plan and implementation of the research, visualization of solutions and data, data curation, validation and verification of model solutions, data and results, project management and financial support of the research.

In the mathematical notation of the hyperbola [2], seven characteristic points are: a – the real semi-axis (2a is the real axis); b – the imaginary semi-axis (2b is the imaginary axis); r_1 and r_2 are vectors or radii for which $r_1 - r_2 = 2a$; $F_1(-c, 0)$, $F_2(c, 0)$ – are the foci of the hyperbola, where $c^2 = a^2 + b^2$; $e = c/a$ is the eccentricity (also, for a hyperbola it holds that

$e > 1$); the lines $y = b/a \cdot x$ and $y = -b/a \cdot x$ – are the asymptotes of the hyperbola; the main equation of the hyperbola is $b^2x^2 - a^2y^2 = a^2b^2$.[2]
For solving problems in SbIn tasks, certain learning laws of human, software, and cyber-physical agents are suitable. Seven characteristic applications include: error correction learning, Hebbian learning, Competitive learning, Boltzmann learning, Thorndike learning, Q-learning, and Deep learning [3]. In the methodological procedure of hyperbolic representation, direct and inverse SbIn problems are analyzed *in the state space and action space* using the Q-learning technique, as one of the most well-known reinforcement learning algorithms. This approach is used to solve decision optimization problems in SbIn tasks, especially in situations where agents need to learn how to make decisions in a dynamic and uncertain environment. The agent decides which actions to choose based on the current state, and for the selected action, it receives a reward based on feedback information.
The agent's decision-making process [3] is modeled as an MDP (Markov Decision Process), represented as a quintuple (S, A, P, R, γ), where: S – state space; A – agent's action space; P – state transition probability function; R – reward to the agent for selected actions; γ – discount factor with a value between 0 and 1.
In this paper, alongside the MDP quintuple, two additional components of the solution model are included: π – the policy or set of decisions that defines the probability distribution for selecting specific actions in a given state, determining the agent's course of action; and α – the learning rate, which determines the magnitude of each update in order to obtain the optimal action-value function, which is Q-learning. The goal is to approximate the optimal Q-values through iterative updates during model training. In addition, deep learning, as a *modern concept in artificial intelligence, represents an advanced branch of machine learning that utilizes multilayer neural networks for analyzing and processing complex and unstructured data* [4]. The following section of the paper provides a review of previous research with a conceptual analysis. The third section contains the material and method of hyperbolic representation. In the fourth section, Q-learning is analyzed for solving inverse SbIn problems, and finally, the conclusion.

2. REVIEW OF PREVIOUS RESEARCH

Learning, from a formal perspective, can be viewed as an algorithmic problem-solving process in system identification and control. According to Helmut Moritz, [5] "learning is an inverse problem," which is applicable in the identification and management of SbIn tasks. Identification begins with the "box model," where distinctions are made between the "black box," "white box," and "gray box." The black box "delivers" a problem solution without insight into the procedures or algorithm. The white box corresponds to a problem-solving model conducted in a transparent manner, including all procedures and steps through which the target result is generated. The gray box is a combination of the black and white box approaches to problem solving [6].
For a more complete understanding of the concept of a problem, three deterministic parameters are used:

- A – represents the "weight matrix" acting as the operative technology that solves the problem task;
- f – represents the inputs or input data on which A performs operations according to a defined problem-solving approach;
- g – the direct output or solution of the problem task, i.e., the reaction to f influenced by A.

The operator A affects the identification of input f from the state space into the action space, so that through a certain number of operations, g is obtained as the result or solution of a properly posed problem.

$$g=Af \qquad (1)$$

A problem must be physically well or properly posed through the attributes of stability, uniqueness, and existence. Meeting these three requirements implies a context of the "constant influence of Laplace's thought experiment" or "intellect" (demon), which is considered "the first instance of articulating causal or scientific determinism." This problem was formulated in 1814 year by Pierre-Simon Laplace, [7] and for calculating certain physical values, the laws of classical mechanics are applied. When a problem is poorly posed or improperly defined, it means that at least one of the required conditions is violated. This implies that either a solution does not exist, or it is not unique, or it is not stable. Therefore, solving such problems requires deep learning by agents, i.e., acknowledging the fact that *"learning is an inverse problem."* In this case, we refer to inverse problems of the first kind, which are solved using the following equation:

$$f=A^{-1}g \qquad (2)$$

The goal is to, through the inversion of operator A, i.e., by determining A^{-1}, retrospectively identify the properties of the object f influenced by the operator, based on the observation g. In this sense, A is a search operator, which links what is observed with what actually represents a set of states in the state space. This defines a second inverse problem, which is expressed by the equation:

$A=f^{-1}g$ (3)

Based on the above, we conclude that SbIn tasks can be identified and managed by applying the theories of inverse problems. From the perspective of *Russell's theory*, the operator A functions as a *passive projection mechanism*, which gathers data from the "external world" f or from the state space, and then analyzes it through g, which represents the results of observation. According to *Popper's theory*, the operator A is an *active searcher*, which not only collects information but also actively participates in human perception. Human perception, of course, does not function as a purely mathematical problem, since the operators A and A^{-1} are realized *biologically and psychologically*, but with embedded mathematical patterns and neuro-computational mechanisms [4, 5, 6].

Deep learning fosters critical thinking, data transfer with visualization and information presentation, the discovery of new knowledge by linking learning content with experience and real-life examples, and promotes active participation and motivation through interactive questioning, the communication of ideas, and the creation of new knowledge through collaborative work. Miller and Krajcnik [8]presented a methodology that connects project-based learning and the theory of useful knowledge.

3. MATERIALS AND METHOD

The functioning of SbIn involves more efficient decision-making and management of tasks that include the execution of various types of events (complex, hazardous, analytical, transactional, business), performing actions and activities related to supervision, control, and analysis of objects in the traffic infrastructure, at locations of physical and other objects in real time. Through collaboration among different agents, SbIn is aimed at improving the application of traffic regulations, ensuring technical standards in infrastructure, and ensuring the safe operation of transport units in road traffic.

3.1. Q-learning

Q-learning is an exceptionally powerful reinforcement learning technique that enables an agent to participate in optimizing its decisions through interaction with the environment. Although it has its challenges, such as complexity in large state spaces, it remains one of the most well-known algorithms in the field of artificial intelligence. Q-learning is one of the most recognized algorithms for reinforcement learning and is used to solve decision optimization problems, particularly in situations where an agent needs to learn how to make decisions in a dynamic and uncertain environment. It is a model-free algorithm, which means that the agent does not need a pre-defined model of the environment but learns from its experiences through interaction with the environment. In Q-learning, the agent needs to learn the best policy (set of decisions) that will maximize the total reward the agent can receive.

3.2. MDP - Markov Decision Processes

Markov Decision Processes (MDP), in the algorithm, include five basic components – a quintuple (S, A, R, P, **γ**):

- Space States (S): Represent all possible situations or circumstances in which the agent can be. Some states are terminal states, indicating the end of the task or episode when the agent achieves its goal. In contrast, there are non-terminal states or intermediate states within the task, where the agent can continue its actions.
- Space Actions (A): Represent all possible actions the agent can take in each state.
- Reward Function (R): After the agent takes an action, it receives a reward that depends on the state and action.
- State Transition Probability Function (P): Represents the probability that the agent will transition from state s to state s' after choosing action a, as shown in equation 4. It is often associated with random environmental influences, such as the dynamic nature of the problem or the impact of external factors.

$$P(s,a,s') = P\left\{S_{t+1} = s' \middle| S_t = s, A_t = a\right\} \tag{4}$$

The temporary reward value function (R) represents the total expected reward that the agent can receive from a given state or action.

$$P(S_{t+1} | s_0, a_0, s_1, ..., s_t, a_t) = P\left\{S_{t+1} \middle| s_t, a_t\right\} \tag{5}$$

- Discount factor (γ): It has a value between [0,1] and is used to control the impact of future rewards on current decisions. A higher discount factor indicates that the agent prioritizes long-term gains, while a lower discount factor implies a greater focus on immediate rewards. The discount factor helps balance the agent's consideration of both short-term and long-term rewards when determining the optimal policy.

$$G_t = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{T-1} R_{t+T} \tag{6}$$

3.3. Extended MDP algorithm with parameters π and α

In addition to the quintuple (S, A, R, P, γ), Q-learning also involves two particularly significant parameters: π and α.

● Policy (π) defines the probability distribution for selecting specific actions in a given state, determining the flow of the agent’s behavior. The policy guides the agent's decision-making process by specifying which actions to take in different states, aiming to maximize the cumulative reward over time.

$$\pi(a|s) = P\left[A_t = a \middle| S_t = a\right] \tag{7}$$

$$Q_\pi(s,a) = E_\pi\left[G_t \middle| S_t = s, A_t = a\right] \tag{8}$$

● The learning rate (α) determines the extent of each update:

$$Q_*(s,a) = \max_\pi Q_\pi(s,a) \tag{9}$$

The essence of achieving the optimal action value function is *Q*-learning, which aims to approximate the optimal *Q*-values through iterative updates. A specific update rule is shown in Equation (10), where **α** represents the learning rate and determines the extent of each update. As training progresses, the *Q*-values gradually converge to the optimal *Q*∗, enabling the identification of the optimal policy.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\,[R_{t+1} + gmaxQ(S_{t+1}, a) - Q(S_t, A_t)] \tag{10}$$

If the agent selects only actions that maximize *Q(s, a)* during training, actions that have not been explored before might be neglected, potentially causing the action value function to become "stuck" in a local optimum. To avoid this, the epsilon-greedy strategy (ϵ-greedy strategy) is typically used, as shown in Equation (11). This strategy allows the agent to explore new actions by occasionally choosing random actions (with probability ϵ), while most of the time it exploits the best-known actions by choosing the ones that maximize *Q(s, a)*.

$$a = \begin{cases} \arg\max Q(s,a), 1-\grave{o} \\ randomaciton, \grave{o} \end{cases} \tag{11}$$

3.4. Advantages, Disadvantages, and Timeline of the Q-learning Process

The advantages of Q-learning can be presented with three statements: model-free, simplicity and efficiency, and optimality.

● Model-free: This qualifier is given because "Q-learning does not require prior knowledge about the environment’s dynamics." The exclusion of the need for an environmental model is very useful in situations where agents must learn directly from their experiences.

● Simplicity and Efficiency: Q-learning is easy to implement and efficient in many reinforcement learning problems, especially when the state and action spaces are relatively small.

● Optimality: Given sufficient time for learning and correct parameter settings (such as learning rate and discount factor), Q-learning leads to an optimal policy or decision set.

The authors [9] point out that the disadvantages of Q-learning are particularly controversial when learning in large state and action spaces, which results in many interactions with the environment. Especially in the case of solving complex and complicated problems in SbIn tasks, the Q-table often becomes very large when the number of possible states and actions increases.

The learning process can be algorithmically broken down into the following four steps [9] :

1. Initial Step: At the beginning, Q-values are usually initialized to random values or zero.
2. Action Selection: At every moment, the agent selects an action from the current state based on the exploration-exploitation trade-off. This means the agent can:
   - o Exploit (use the currently best action) with probability $1-\epsilon$
   - o Explore (choose a random action) with probability ϵ. This is often referred to as the epsilon-greedy strategy, where ϵ controls the level of randomness.
3. Q-value Update: After the agent selects an action, it receives a reward and transitions to a new state. The Q-values are updated according to the formula above, allowing the agent to adjust its estimate of the value of the actions that led to positive outcomes.

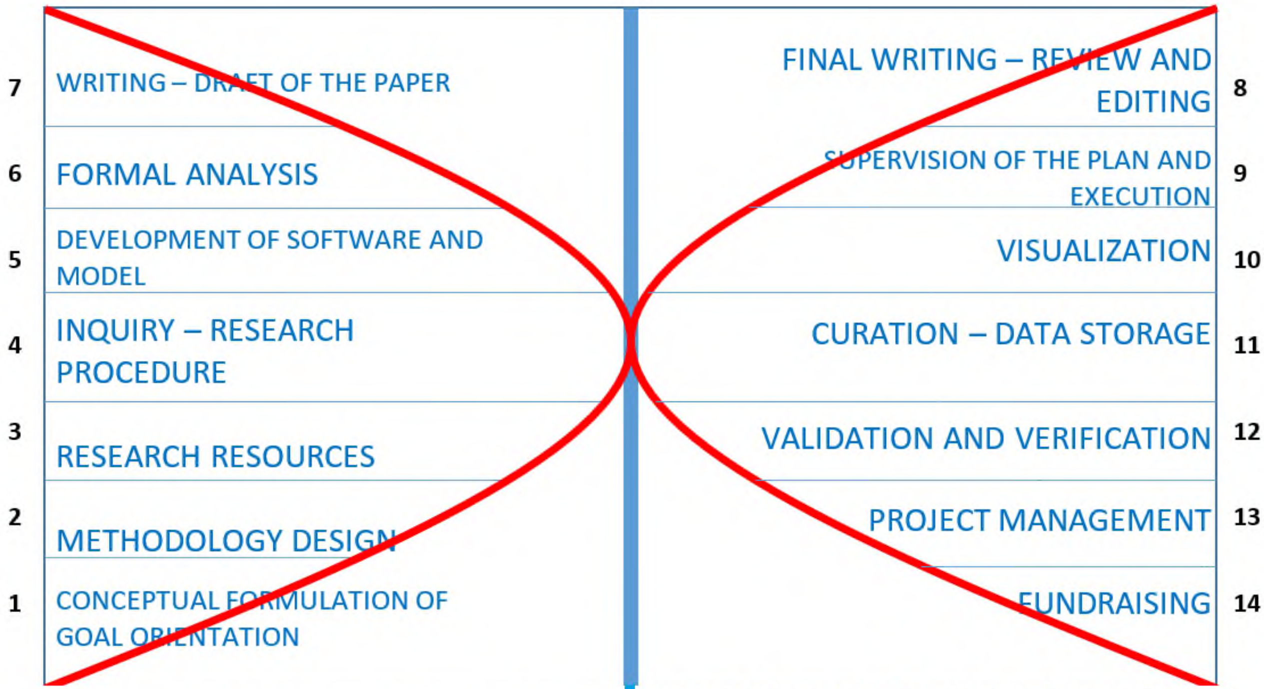
4. Re-learning: This process repeats until the agent learns the optimal policy, i.e., until the Q-values converge, and the agent starts to recognize the best possible actions in each state.

3.5. Textual and graphical representation of the hyperbolic representation of septuples in the methodological approach of *Q*-learning and Deep Learning.

The hyperbolic representation of the methodological procedure for modeling machine learning techniques (*Q*-learning and Deep Learning) is used to solve complex inverse problems in traffic inspection tasks. The symmetric geometric shape of the hyperbola with two branches (left and right) and seven characteristics, given by mathematical notation, is paired with seven process instances to solve inverse problems in two phases of representation: seven instances of the creative phase on the left side (ascending branch) in a down-top orientation, and seven instances of the implementation and evaluation phase on the right side (descending branch) in a top-down orientation (Figure 1). In the tabular part of the hyperbolic representation, the definitional descriptions and objectives of each instance are provided.

4. MODELING OF DEEP LEARNING AGENTS IN Sbln TASKS IN THE SELECTED GEO-RESEARCH AREA

From a technological perspective, "deep learning" is the most advanced form of the broader field of artificial intelligence. The representative power of machines to "learn," especially supercomputers that process large amounts of data, is increasingly resembling the functioning of the human brain. For this reason, there is also deep learning in human education, seen as the best method of learning, which requires clear and consistent understanding, connecting new knowledge with old, and its meaningful application in the real context of life. The context of deep learning is content-rich, dynamic, interactive, and creatively motivational, followed by the guidance of one's own learning process towards collaborative work and problem-solving.
In this paper, the goal of creating a deep learning model is the classification of the Importance Rank of event types in managing SbIn tasks (Y) based on the following predictors or independent variables: 1) Importance Rank of task acceptance (X1); 2) Importance Rank of task quality (X2); 3) Importance Rank of research expectations (X3); 4) Importance Rank of cognitive alertness and capacity of human agents (X4); 5) Importance Rank of awareness of bias and data shifting (X5); 6) Importance Rank of data quality identification in evaluation (X6). All of the listed variables are categorical in nature with coded values ranked in categories from 1 to 5, and the dataset consists of 32 instances (Table 1). Using the Data Augmentation method, with the help of an Autoencoder artificial neural network, the dataset was expanded with synthetically reconstructed "copies" of the data to 1024 instances, i.e., 1024 input-output vectors. This was achieved by inputting the 32 available vectors into the network's input layer, which results in the same number of modified copies being produced at the output as interpretations of the input. Therefore, in one described iteration, the input dataset is doubled. In each subsequent iteration, the input layer of the autoencoder network receives the sum of the input vectors and the corresponding number of reconstructed vectors from the previous iteration. In total, five iterations were performed, increasing the original dataset 32 times.
The deep neural network model was created in the R programming language, using the RStudio integrated development environment (IDE), with the libraries keras, readxl, dplyr, haven, and caret. The model's hyperparameters were set to the following values:
-Data Split. Training set: 90% of the data; Test set: 10% of the data; Set seed: set.seed(123) for result reproducibility.
-Network Architecture. Input layer with 64 artificial neurons, activation function: Rectified Linear Unit (ReLU); Four hidden layers with 32 neurons and ReLU activation function; Output layer with the number of units corresponding to the number of classes (ncol(y_cat)), activation function: softmax.
-Model Preparation for Training: Loss function: categorical_crossentropy; Optimizer: Adam with a learning rate of 0.001; Metric: Accuracy.
-Model Training: Number of epochs: 100; Batch size: 32; Validation: 10% of the data used for validation during training (validation_split = 0.1).
The results of training the model with the described hyperparameters are shown in Figure 2. The loss graph (Figure 2.a) illustrates how the loss function value changes over the epochs of training. The loss decreases during the training for both data sets (Validation and Train), indicating that the model is becoming better fitted to the data. In contrast to the loss function, the accuracy increases with the number of epochs, with the largest growth occurring at the beginning of the training, and in the final phase, the accuracy curve begins to flatten out (Figure 2.b).



CREATIVE PHASES OF REPRESENTATION	IMPLEMENTATION AND EVALUATION PHASES OF REPRESENTATION
7) The master equation of pave *xa*. *Draft text of the original creative representation.* Preparation, creation and/or presentation of work, especially writing initial drafts (including substantial translation).	**8) The main equation of the line yb** *Reviewing and editing the final text of the representation.* Preparation, creation and/or review of published work by those in the original research group, particularly review, commentary or revision – including pre- or post-publication stages.
6) Asymptote of the ascending limb of the hyperbola H1 *Statistical, mathematical and computer elements of the form (conceptual, visual, constructional, relational) representation.* Application of statistical, mathematical, computer or other formal techniques for analysis or synthesis of data from studies.	**9) Asymptote of Descending Arm H** *Realization of oversight and responsibility in leadership representation* for planning and execution of research, activities, including mentoring outside the core team.
5) Eccentricity of the ascending arm of the hyperbola C1. *Functional and non-functional specifications of software solutions* representing code components and algorithms; design of computer programs; implementation of computer code and accompanying algorithms; testing existing code components.	**10) Eccentricity of the downward arm of the hyperbola C2** *Visualization of the content and graphic structure of the representation.* Preparation, creation and/or presentation of published work, visualization/presentation of data.
4) Location of Imaginary focal point of F1 *representation.* Conducting research and investigation processes, conducting experiments or gathering data/evidence.	**11) Locating the real focal point F2** *in the realization of the representation.* Management activities for metadata production, data cleansing and maintenance of research data (including software code, where necessary for data interpretation) for initial use and subsequent reuse.
3) Radius vector HR1 *associated research resources/data representation.* Provision of study material, participants in the research process, formation of samples or research instruments, computer resources or other tools for analysis.	**12) Radius vector HR2.** *Validation of experimental planned and realized results of representation.* Verification, either as part of the activity or separately, of replication/reproducibility of results/experiments and other research results
2) Imaginary axis 2b *Holistic design of representation methodology*; creating a "black box" model and algorithm.	**13) Real axis 2a.** *Management of the administration of the representation procedure.* Responsibility for the management and coordination of the planning of the execution of the research activity.
1) Imaginary semi-axis of hyperbola *Interpretable conceptualization of the theme of representation*; formulation or evolution of comprehensive research goals - specific statements of intentions and detailed description of the target destination (goals) and the goals of the desired outcomes and vision of the development of the topic with an action plan (aims).	**14) The real semi-axis of the hyperbola** *of curation and digital treatment of the results of representation.* Obtaining financial support for the project that led to this publication.

Figure 1. Textual and graphical representation of the hyperbolic representation of septuples in the methodological approach for modeling machine learning is used to solve complex inverse problems in traffic inspection tasks

Table 1. Part of the coded dataset with research variables

Serial number	Importance Rank of event types in managing SbIn tasks (Y)	Code	Importance Rank of task acceptance (X1)	Importance Rank of task quality (X2)	Importance Rank of research expectations (X3)	Importance Rank of cognitive alertness and capacity of human agents (X4)	Importance Rank of awareness of bias and data shifting (X5)	Importance Rank of data quality identification in evaluation (X6)
1	5	TI 01	5	5	5	5	5	3
2	5	TI 02	5	5	4	4	4	5
3	1	TI 03	3	2	2	2	2	3
...	...	...	...	...	...	...	...	...
32	1	TI 32	3	2	3	3	2	4

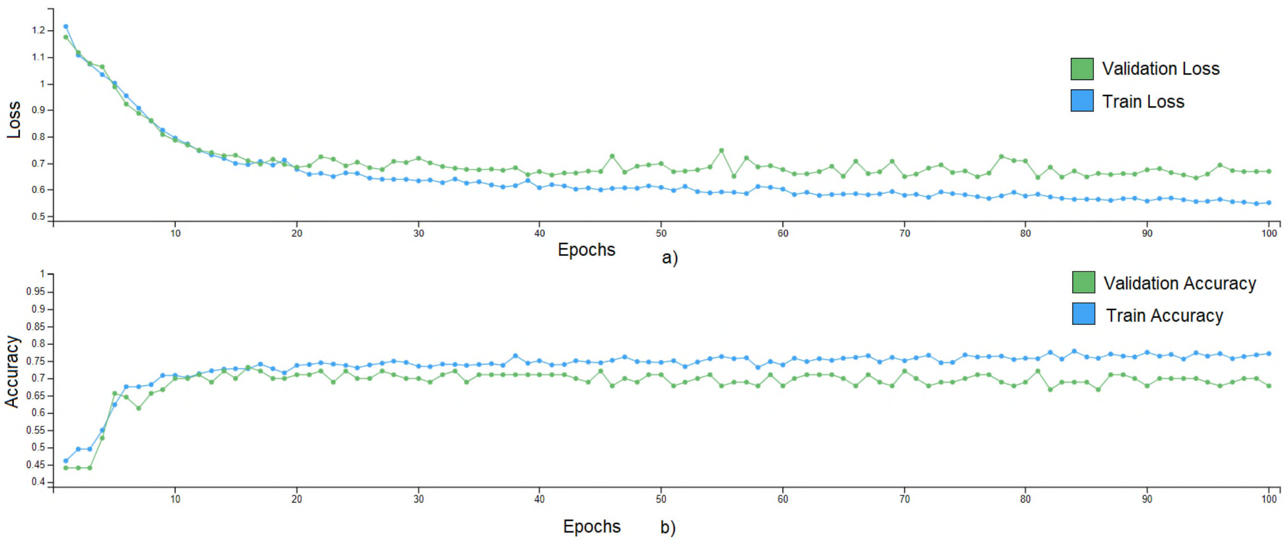


Figure 2. Model training history: a) loss and b) classification accuracy during training epochs.

The overall accuracy of the model on the test set is 80.58%, and a more detailed insight into the prediction results by individual classes is possible based on the confusion matrix shown in Table 2. The confusion matrix shows that the model classifies the majority of examples well, with the most accurate predictions for classes 0 and 1. For class 0, the model correctly predicted 13 times, while it made a mistake by predicting class 1 only once. Class 1 was correctly predicted 2 times without errors. Class 2 has 15 correct predictions, but also a few errors, especially in the classification as class 3. Class 3 was mostly accurately classified with 53 correct predictions, while the errors were mostly in classes 2 and 4. Although the errors are minimal, especially for classes 0 and 1, the model could be improved in the predictions for class 2, which is often replaced by class 3.

Table 2. Confusion matrix for a deep neural network model

		Actual				
		0	**1**	**2**	**3**	**4**
Predicted	**0**	13	1	0	0	0
	1	0	2	0	0	0
	2	1	4	15	3	0
	3	0	1	7	53	3

5. CONCLUSION

This paper explores *Q*-learning as an extremely powerful tool in reinforcement machine learning, allowing agents to participate in optimizing their decisions through interaction with the environment. Although it has its challenges, such as complexity in large state spaces, it is still one of the most famous algorithms in the field of artificial intelligence. The second line of research is focused on solving complex and complex inverse problems in the context of traffic, referring to situations in which SbIn should achieve certain goals such as improving traffic safety). Certain theories of inverse

problems applicable in response to the questions of what actions, strategies or parameters should be applied in order to achieve the set goal functions are analyzed. In the hyperbolic representation of the methodological approach, Q-learning is identified as extremely useful for solving inverse problems in traffic, because it enables the optimization of decision-making policies in the management of SbIn and the entire structure of traffic systems. Minimizing the number of accident events, reducing the risk of accidents and improving overall road traffic safety are highlighted as objective functions and value functions. In this context, *Q*-learning does not mean only attempts to experientially "guess" what the best actions are, but proactive learning from joint actions and collaborative experience of SbIn ageats in order to find the optimal policy (set of decisions) that maximizes traffic safety. To solve inverse problems in SbIn, individual components and parameters should be analytically focused. In the state space S, the agents orient themselves to the post-temporal situational factors, among which are the density of traffic dynamics, the number of vehicles on the traffic lanes, the current speeds of movement in certain states, estimating the probability time until the next accident, etc. Actions (a) can be included in the area of actions (A) for regulating the duration of individual traffic lights, changing speed limit parameters or situational decisions on traffic flow diversion during peak hours during the day. The value of rewards (R) is related to the reduction of the total number of accidents or accidents caused in situations of traffic congestion or the occurrence of hazardous events. It is important to point out that the reward policy (π) includes not only positive results but also negative ones. Through Q-learning, the agent chooses actions by iteratively learning which actions to decide how to minimize accidents and their consequences.
The paper specifically analyzes "deep learning" as the most advanced form of the wider field of artificial intelligence, which, in terms of its functioning, very closely approaches the human brain. That is why deep learning is the best method of learning agents in SbIn jobs that require a clear consistent understanding, connecting new with old functional knowledge and meaningful application in the real unstable context of business events. The multidimensional context of deep learning is meaningful, dynamic, interactive and creatively motivating, followed by directing the SbIn agents' own learning process towards joint work and solving complex and complex problems.
In the ordered research steps, a more detailed modeling of solving inverse problems based on multi-agent communication in SbIn affairs in the selected area is foreseen, which includes the interaction between human agents (HA), software agents (SA) and cyber-physical agents (CPA), which together perform tasks through the exchange of structured data and messages.

REFERENCES

[1] A. Đukić, M. K. Banjanin, M. Stojčić, T. Đurić, R. Đekić, and D. Anđelković, "An ensemble of machine learning models for the classification and selection of categorical variables in traffic inspection work of importance for the sustainable execution of events," *Sustainability*, vol. 16, no. 22, p. 9720, Nov. 2024, doi: 10.3390/su16229720.

[2] Matematički fakultet Univerziteta u Beogradu, "Hiperbola," [Online]. Available: https://www.maf.bg.ac.rs. [Accessed: Jan. 21, 2025].

[3] M. K. Banjanin, A. Đukić, M. Stojčić, and M. Vasiljević, "Machine learning models in the classification and evaluation of traffic inspection jobs in road traffic and transport," in *Proc. 13th Int. Sci. Conf. SED 2023*, Vrnjačka Banja, Serbia, Jan. 5–8, 2023.ouci.dntb.gov.ua

[4] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/science.aaa8415.

[5] H. Moritz, *Znanost, um i svemir: uvod u prirodnu filozofiju*. Zagreb, Croatia: Školska knjiga, 1998.

[6] W. Houghton, *Engineering Subject Centre Guide: Learning and Teaching Theory for Engineering Academics*. Loughborough, UK: Higher Education Academy, 2004.

[7] P. S. Laplace, *A Philosophical Essay on Probabilities*, 6th ed., F. W. Truscott and F. L. Emory, Trans. New York, NY, USA: Dover Publications, 1951.

[8] E. C. Miller and J. S. Krajcik, "Promoting deep learning through project-based learning: A design problem," *Disciplinary and Interdisciplinary Science Education Research*, vol. 1, no. 1, pp. 1–10, 2019, doi: 10.1186/s43031-019-0009-6.

[9] T.-J. Zhao, K. Gao, X. Li, and H. Sun, "Deep learning-based design image management," presented at the *Int. Conf. Environmental Science and Sustainable Energy*, Guangzhou, China, 2023.