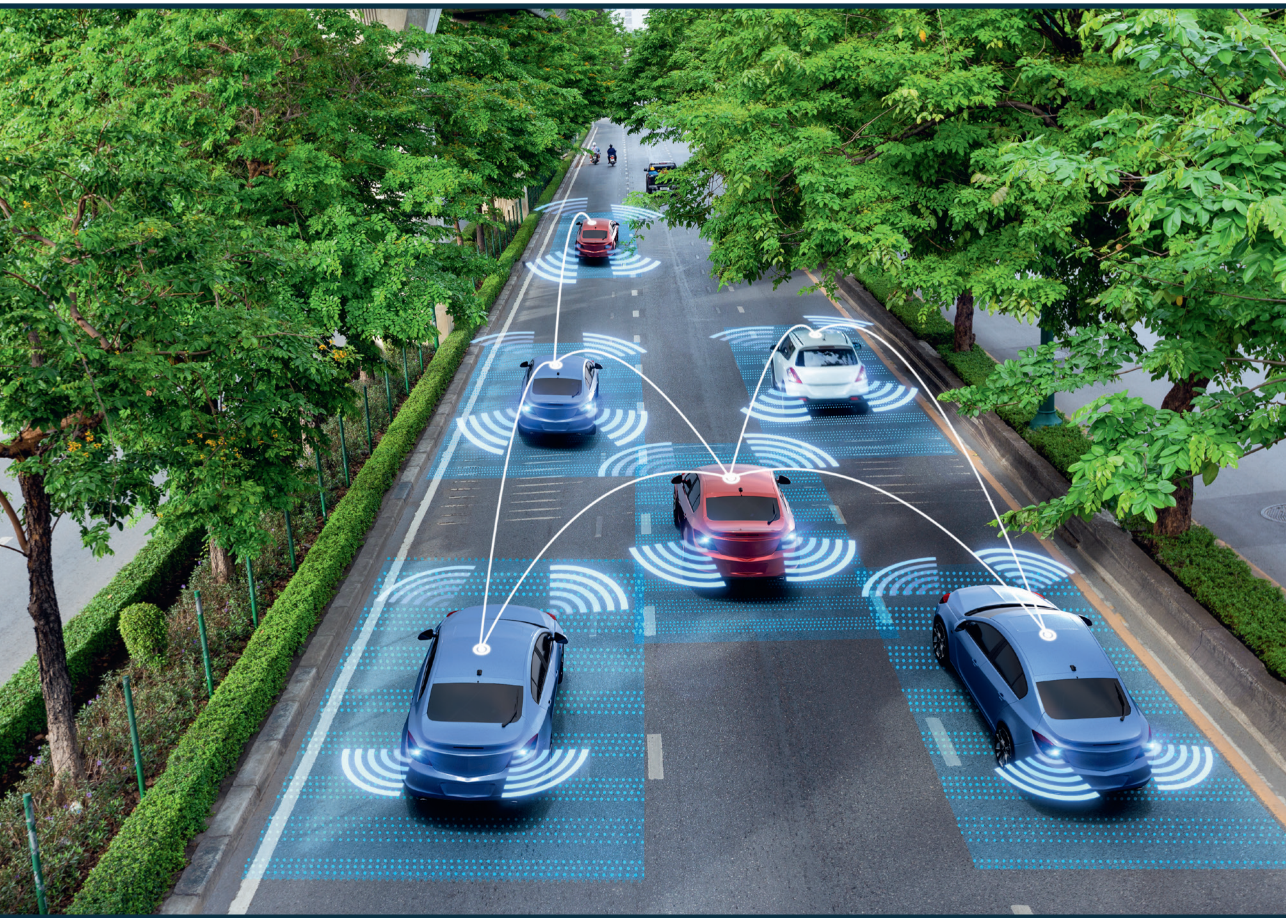
RIGA TECHNICAL UNIVERSITY

Anna Beinaroviča

RESEARCH AND DEVELOPMENT OF IMMUNE NEURAL NETWORK ALGORITHMS FOR ELECTRICAL TRANSPORT DANGEROUS SITUATION RECOGNITION AND PREVENTION

Summary of the Doctoral Thesis



RTU Press
Riga 2023

RIGA TECHNICAL UNIVERSITY
Faculty of Electrical and Environmental Engineering
Institute of Industrial Electronics and Electrical Engineering

Anna Beinaroviča
Doctoral Student of the Study Programme "Computerised Control of Electrical Technologies"

RESEARCH AND DEVELOPMENT OF IMMUNE NEURAL NETWORK ALGORITHMS FOR ELECTRICAL TRANSPORT DANGEROUS SITUATION RECOGNITION AND PREVENTION

Summary of the Doctoral Thesis

Scientific supervisor
Professor Dr. sc. ing. MIKHAIL GOROBETZ

RTU Press
Riga 2023

Beinaroviča, A. Research and Development of Immune Neural Network Algorithms for Electrical Transport Dangerous Situation Recognition and Prevention. Summary of the Doctoral Thesis. – Riga: RTU Press, 2023. – 40 p.

Published in accordance with the decision of Promotion Council “RTU-14” of 15 May 2023, Minutes No. 04030-9.12.2/3.

The Thesis has been supported by the European Social Fund within the Project No. 8.2.2.0/20/I/008 “Strengthening of PhD Students and Academic Personnel of Riga Technical University and BA School of Business and Finance in the Strategic Fields of Specialization” of the Specific Objective 8.2.2 “To Strengthen Academic Staff of Higher Education Institutions in Strategic Specialization Areas” of the Operational Programme “Growth and Employment” and by Riga Technical University Doctoral Grant Programme.

https://doi.org/10.7250/9789934229879
ISBN 978-9934-22-987-9 (pdf)

DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR THE PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF SCIENCE

To be granted the scientific degree of Doctor of Science (Ph. D.), the present Doctoral Thesis has been submitted for the defence at the open meeting of RTU Promotion Council on November 27, 2023 at the Faculty of Electrical and Environmental Engineering of Riga Technical University, Riga, Azenes street 12/1, Room 212.

OFFICIAL REVIEWERS

Professor Dr. sc. ing. Nadežda Kuņicina
Riga Technical University

Professor Dr. sc. ing. Carlos M. Travieso-González
University of Las Palmas de Gran Canaria, Spain

Professor Dr. sc. ing. Irina Jackiva (Yatskiv)
Transport and Telecommunication Institute, Latvia

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for the review to Riga Technical University for the promotion to the scientific degree of Doctor of Science (Ph. D.) is my own. I confirm that this Doctoral Thesis had not been submitted to any other university for the promotion to a scientific degree.

Anna Beinarovičа ……………………………. (signature)
Date: ………………………

The Doctoral Thesis has been written in English. It consists of Introduction, 5 chapters, Conclusions, 66 figures, 158 formulas, 81 tables; the total number of pages is 175. The Bibliography contains 92 titles.

CONTENTS

Topicality of the problem

It is not only Tesla manufacturers who are struggling to perfect self-driving technology. Honda manufacturers as well as Waymo owned by Google and Cruise owned by General Motors are working on it too. All these companies predicted that they would have full self-driving cars by 2020. But as yet, none of them has. Creating a full self-driving vehicle is much harder than vehicle makers initially thought. There is a wide range of possible risks and elements to consider.

This fact and completed reviews allow concluding that the problem of improving transport safety by the artificial intelligence systems is topical and requires scientific contribution for its solution.

The Doctoral Thesis is related to the safe motion of electric transport providing novel immune neural network-based algorithms for its control. The Thesis contributes to the safety improvement of multiple collaborating unmanned electric vehicles moving and performing own tasks in the same area by researching and developing immune neural network technology. The developed technology provides the ability for continuous unsupervised self-learning to avoid collisions by changing speed and trajectory maximizing the efficiency of task performance in real time.

Goal and tasks

The goal of the Thesis is to develop an immune neural network-based technology of machine learning for unsupervised safe vehicle control.

The main hypothesis is that an immune neural network can make control decisions to prevent vehicle collisions with better performance than a traditional neural network.

The following tasks were defined:

- To study the objects of electric transport traffic movement control and their interaction.
- To study the existing solutions, which are based on the algorithms of artificial neural network, for dangerous situation recognition and prevention of electric transport.
- To compare centralized, decentralized and distributed system structures, to choose the most suitable one for the proposed task, and to develop a novel system structure, which could help to make the proposed system cheaper, faster and easier to implement.
- To develop mathematical models and algorithms that could help to solve different types of transport safety and collision prevention tasks, such as object recognition, traffic light signal recognition, possible crossing point detection, collision probability evaluation, and collision prevention.
- To develop a novel immune neural network-based algorithm for dangerous situation recognition and prevention of unmanned electric vehicle.
- To develop an electric circuit diagram with an immune memory based on a single board computer for unmanned electric vehicle.
- To make computer simulations and to prove the efficiency of the proposed algorithms.

Scientific novelty

The most significant scientific novelty of the Doctoral Thesis is the immune neural network technology inspired by two biological systems – immune system and neural networks and their artificial analogs. The developed novel mathematical models and algorithms for this technology allows skipping the preliminary supervised training step and adapted for real-time continuous unsupervised self-learning of unmanned electric vehicle to recognize the dangerous situation and prevent the collision by making control decisions autonomously keeping the structures and weights of separate neural networks in the immune memory and retraining them to minimize the collision probability and maximizing the performance.

New mathematical models and algorithms for possible crossing point detection, for collision probability evaluation, and for collision probability minimization by the neural network are developed in the Thesis for this purpose.

Additional safety improvement mathematical models and methods for object recognition and traffic light signal recognition are developed in the Thesis and integrated in the proposed system.

Practical application

Algorithms developed in the Thesis can be implemented in intelligent electric vehicle control systems to avoid crashes and minimize the risk of collisions. The results of the developed algorithms offer solutions to the tasks of data collection from video surveillance cameras, sensors, cloud databases and other objects of intelligent transport infrastructure, information processing, identification of potentially dangerous situations, risk assessment and decision-making on measures to avoid an accident.

The developed algorithms allow implementing the computer modeling and simulation of optimal control system for electric transport to recognize and prevent dangerous situations. The proposed algorithms are multifunctional and can be implemented into different types of vehicles without mandatory changes and improvement of the infrastructure objects. However, the intelligent infrastructure may provide additional inputs to the developed system.

Approbation

1. International conference "2020 IEEE 61th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)", report "Unsupervised Transport Vehicle Control: Simulation Study and Performance Results", 2020, Riga, Latvia.
2. International conference "2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)", report "Modelling and Simulation of Transport Collision Probability Recognition Algorithm for Traffic Safety", 2019, Riga, Latvia.

3. International conference “Applications of Intelligent Systems (APPIS 2019)”, report “Machine Learning Algorithm of Immune Neuro-Fuzzy Anti-collision Embedded System for Autonomous Unmanned Aerial Vehicles’ Team”, 2019, Las Palmas de Gran Canaria, Spain.
4. International conference “20th European Conference on Power Electronics and Applications, EPE’18 ECCE Europe”, report “Algorithm for Immune Neural Network in Transport Collision Prevention Control System of Unmanned Electrical Vehicle”, 2018, Riga, Latvia.
5. International conference “2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Algorithm of Signal Recognition for Railway Embedded Control Devices”, 2018, Riga, Latvia.
6. International conference “22nd International Scientific Conference. Transport Means 2018”, report “Self-Organized Learning Algorithm for Immune Neuro-Fuzzy Anti-collision System of Autonomous Unmanned Aerial Vehicles’ Team”, 2018, Trakai, Lithuania.
7. International conference “12th International Conference Intelligent Technologies in Logistics and Mechatronics Systems”, report “Control Algorithm of Multiple Unmanned Electrical Aerial Vehicles for Their Collision Prevention”, 2018, Panevėžys, Lithuania.
8. International conference “2017 IEEE 58th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Convolutional Neural Network in Turn Recognition Tasks for Electric Transport Safety”, 2017, Riga, Latvia.
9. International symposium “25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017)”, report “Convolutional Neural Networks of Active Railway Safety System with Braking Dynamics Prediction. Dynamics of Vehicles on Roads and Tracks”, 2017, Rockhampton, Queensland, Australia.
10. International conference “31st European Conference on Modelling and Simulation”, report “Modeling and Simulation of Public Transport Safety and Scheduling Algorithm”, 2017, Budapest, Hungary.
11. International conference “Building up Efficient and Sustainable Transport Infrastructure (BESTInfra)”, report “Innovative Neuro-fuzzy System of Smart Transport Infrastructure for Road Traffic”, 2017, Prague, Czech Republic.
12. International conference “57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Immune Algorithm and Intelligent Devices for Schedule Overlap Prevention in Electric Transport”, 2016, Riga, Latvia.

Publications

1. Beinarovica, A., Gorobetz, M., Ribickis, L. Immune Neuro-Fuzzy Network Based System for Collision Free Motion Control of Unmanned Electrical Vehicles. In: 25^{th} European

Conference on Power Electronics and Applications (EPE 2023): Proceedings, Aalborg, Denmark, 4.-8. September 2023.

2. Gorobetz, M., Ribickis, L., Beinarovica, A., Kornejevs A. Immune Neural Network Machine Learning of Autonomous Drones for Energy Efficiency and Collision Prevention. In: Open access chapter “Drones - Various Applications”, September 2023.
3. Beinarovica, A., Gorobetz, M., Alps, I. Unsupervised Transport Vehicle Control: Simulation Study and Performance Results. In: 2020 IEEE 61st International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2020): Proceedings, Riga, Latvia, 5–7 November 2020, pp. 303–308.
4. Beinarovica, A., Gorobetz, M., Alps, I. Modelling and Simulation of Transport Collision Probability Recognition Algorithm for Traffic Safety. In: 2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2019): Proceedings, Riga, Latvia, 7–9 October 2019, pp. 1–6.
5. Beinarovica, A., Gorobetz, M., Ribickis, L., Levcenkov, A. Machine Learning Algorithm of Immune Neuro-Fuzzy Anticollision Embedded System for Autonomous Unmanned Aerial Vehicles Team. In: 2nd International Conference on Applications of Intelligent Systems (APPIS 2019): Proceedings. ACM International Conference Proceedings Series, Las Palmas de Gran Canaria, Spain, 7–9 January 2019, pp. 1–8.
6. Beinarovica, A., Gorobetz, M., Alps, I., Levcenkov, A. Algorithm for Immune Neural Network in Transport Collision Prevention Control System of Unmanned Electrical Vehicle. In: 2018 20th European Conference on Power Electronics and Applications (EPE'18 ECCE Europe): Proceedings, Riga, Latvia, 17–21 September 2018, pp. 1–8.
7. Beinarovica, A., Gorobetz, M., Alps, I., Levcenkov, A. Algorithm of Signal Recognition for Railway Embedded Control Devices. In: 2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2018): Proceedings, Riga, Latvia, 12–14 November 2018, pp. 1–5.
8. Beinarovica, A., Gorobetz, M., Levcenkov, A. Control Algorithm of Multiple Unmanned Electrical Aerial Vehicles for Their Collision Prevention. In: 12th International Conference Intelligent Technologies in Logistics and Mechatronics Systems: Proceedings, Panevėžys, Lithuania, 26–27 April 2018, pp. 37–43.
9. Beinarovica, A., Gorobetz, M., Levcenkov, A. Self-Organized Learning Algorithm for Immune Neuro-Fuzzy Anti-collision System of Autonomous Unmanned Aerial Vehicles’ Team. In: Transport Means 2018: Proceedings of the 22nd International Scientific Conference. Part 3, Trakai, Lithuania, 3–5 October 2018, pp. 1334–1341.
10. Beinarovica, A., Gorobetz, M., Levcenkov, A. Convolutional Neural Networks of Active Railway Safety System with Braking Dynamics Prediction. In: Dynamics of Vehicles on Roads and Tracks: Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017): Proceedings, Rockhampton, Australia, 14–18 August 2017, pp. 953–958.
11. Beinarovica, A., Gorobetz, M., Levcenkov, A. Modeling and Simulation of Public Transport Safety and Scheduling Algorithm. In: 31st European Conference on Modelling

and Simulation (ECMS 2017): Proceedings, Budapest, Hungary, 23–26 May 2017, pp. 215–221.
12. Beinarovica, A., Gorobetz, M., Levcenkov, A. Convolutional Neural Network in Turn Recognition Tasks for Electric Transport Safety. In: 2017 IEEE 58th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2017): Proceedings, Riga, Latvia, 12–13 October 2017, pp. 231–236.
13. Beinarovica, A., Gorobetz, M., Levcenkov, A. Innovative nauro-fuzzy system of smart transport infrastructure for road traffic safety. In: Building up Efficient and Sustainable Transport Infrastructure (BESTInfra 2017): Proceedings, Prague, Czechia, 21–22 September 2017, pp. 1–8.
14. Alps, I., Gorobetz, M., Beinarovica, A., Levcenkov, A. Immune Algorithm and Intelligent Devices for Schedule Overlap Prevention in Electric Transport. In: 2016 57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2016): Conference Proceedings, Riga, Latvia, 13–14 October 2016, pp. 191–197.
15. Beinarovica, A., Gorobetz, M., Levcenkov, A. Algorithm of Energy Efficiency Improvement for Intelligent Devices in Railway Transport. In: Electrical Control and Communication Engineering (ECCE journal), Riga, Latvia, 2016, pp. 29–34.
16. Gorobetz, M., Ribickis, L., Alps, I., Beinarovica, A. Patent application “Bezpilota Transporta Līdzekļa Sadursmju Novēršanas Iekārta ar Pašapmācošo Imūno Atmiņu”.

Content of the Doctoral Thesis

Introduction

The introduction of the Thesis describes the industrial and scientific research review on the topic of the Thesis, as well as assesses the topicality and scientific novelty of the topic, defines the goal and tasks of the work, and provides information about the approbation and practical application. The works closest to the topic of the Thesis are [19], [23], which offer methods of artificial intelligence – neural network strengthened by immune memory. In difference to [19], the main idea of the Doctoral Thesis is that input data are stored in the immune memory together with weights that were used previously for solving this situation, which helps to reduce calculation time that is very important for real time systems. In turn, the method presented in [23] was not based on a quality function to evaluate the solutions.

1. Problem formulation of electric transport safety control task

Chapter 1 of the Doctoral Thesis is devoted to comparing centralized, decentralized, and distributed system models and developing the novel system structure, which could help to make the proposed system cheaper, faster, and easier to implement.

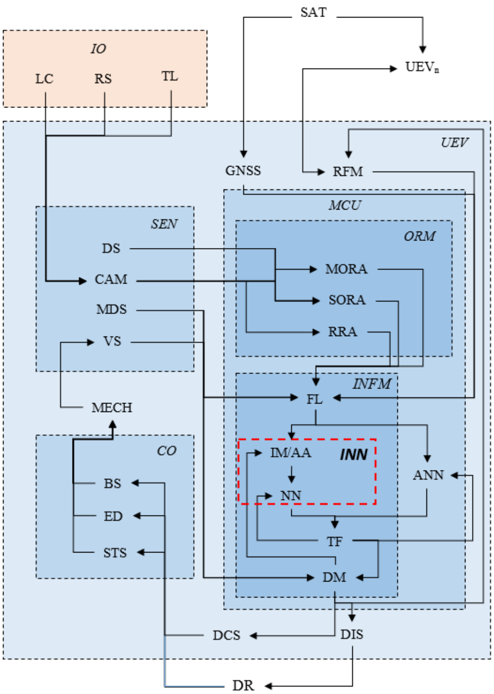


Fig. 1.1. Scheme of the distributed system structure with reduced number of components.

The results of comparison show that the distributed system is more preferable than the centralized or decentralized. Distributed models are easier to implement, they have less components, they are cheaper for infrastructure owner, and they are not connected to the specific area; they have also decreased time for reaction and decreased risk at system failure. That is why the distributed system structure was used in this research.

A scheme of the distributed system structure is shown in Fig. 1.1, where SAT – satellite; UEV – unmanned electric vehicle; UEVn – other unmanned electric vehicle; LC – level crossings; RS – road signs; TL – traffic lights; GNSSR – GNSS receiver; RFM – radio frequency module; DS – distance sensor; CAM – videocamera; MDS – movement direction sensor; VS – speed sensor; MORA – moving object recognition algorithm; SORA – static object recognition algorithm; RRA – road recognition algorithm; FL – fuzzy logic; IM – immune memory; AA – affinity algorithm; ANN – traditional neural network; NN – neural network as a part of the INN; INN – immune neural network; TF – target function; DM – decision making module; DIS – driver informing system; DCS – driver control system; BS – electric vehicle braking system; ED – electric vehicle's electric drive; STS – electric vehicle's steering system; MECH – mechanical part; IO – infrastructure object; CO – control object; SEN – sensor; MIC – microcontroller; ORM – object recognition module; INFM – immune neuro-fuzzy module; DR – electric vehicle's driver.

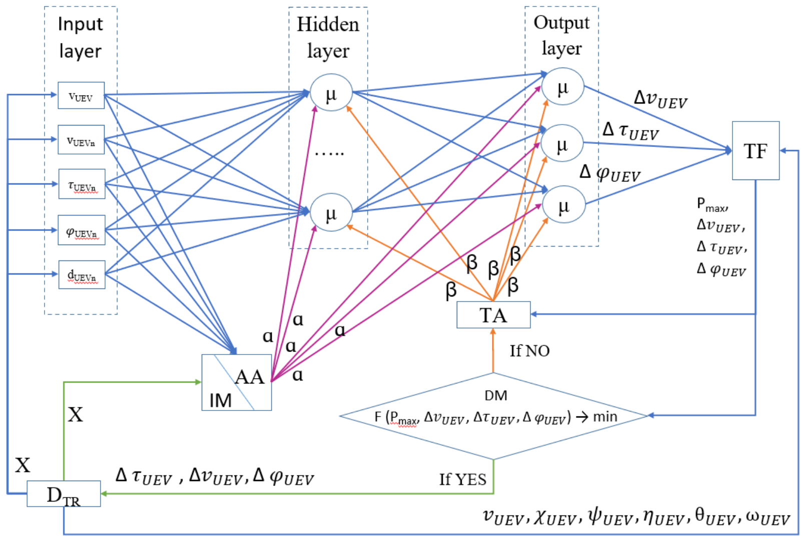


Fig. 1.2. Immune neural network system structure of UEV.

Different parts of the proposed system are used for different tasks. ORM object recognition module, based on convolutional neural network (CNN), is used for the moving and static object recognition tasks as well as for road recognition task. INFM immune neuro fuzzy module, based on the novel immune neural network (INN), is used for unsupervised collision probability evaluation and minimization task. Artificial neural network (ANN) is used for evaluating and minimizing collision probability, along with fuzzy logic. In this research, ANN is included to compare its results with those of the proposed novel INN. The objective is to draw conclusions on whether the novel network is better or worse than the traditional one.

Three subsystem structures were developed and described in the Doctoral Thesis: subsystem structure for object recognition, based on traditional CNN; subsystem structure for electric vehicle collision probability evaluation and minimization, based on traditional ANN; novel INN based technology of machine learning for unsupervised safe vehicle control (Fig. 1.2.). Description of the traditional neural networks is provided in the Doctoral Thesis.

The novel INN-based technology can be used in distributed systems. It obtains data, makes calculations, and provides necessary solutions how to avoid collision in the context of one particular UEV. It does not provide solutions for other participants.

<u>Input data (X).</u> Data: v_{UEV} – speed of the own electric vehicle; v_{UEVn} – speed of other electric vehicles; τ_{UEVn} – horizontal movement direction of the other electric vehicle relative to one's own direction; φ_{UEVn} – vertical movement direction of the other electric vehicle relative to one's own direction; d_{UEVn} – distance till the possible crossing point with the other electric vehicle. Number of parameters in the input data (X) depends on the situation – the number of other UEVs in the control area of own UEV. There is one input parameter for own UEV: speed. There are 4 input parameters for other UEVs: speed, horizontal movement direction in relation to the own UEV, vertical movement direction in relation to the own UEV, and distance till possible crossing point. Data is received from the UEV embedded electronic device and is sent to the input layer of the immune neural network INN.

<u>Input layer.</u> Input layer receives input data (X). Each UEV considers only those UEVs which are in his control area in order to minimize number of necessary calculations. Input data (X) is ordered for a more accurate recognition of the situation. The goal is to order multiple UEVs in relation to the own UEV to better understand their position and relative movement. Three parameters are used for ordering the UEVs: the horizontal movement direction τ_{UEVn}, the vertical movement direction φ_{UEVn}, and the distance till the crossing point d_{UEVn}. Ordering of other UEVs is done according to the slope τ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of τ_{UEVn}, then ordering of these UEVs is done according to the slope φ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of φ_{UEVn}, then ordering of these UEVs is done according to the distance to the crossing point with these UEV d_{UEVn}. This method helps to describe the situation accurately. Input data (X) are sent from the input layer to the affinity algorithm AA and hidden layer.

<u>Affinity algorithm (AA).</u> AA checks all the similar situations stored in the IM and calculates discrepancies $\mathcal{E}$. Situation with a smallest discrepancy $\mathcal{E}$ is chosen and its number α is sent to all μ neurons of the INN. If there is no similar situation stored in IM, then situation number α = 0 is sent to the μ neurons.

<u>Immune memory (IM).</u> Database that contains input data about previous situations that were solved. Each situation has its number α. All the data in IM is stored in clusters for easier and faster match finding processes. For example, if 3 vehicles are participating in the possible collision situation, there is no need to find a similar situation in the group of situations with 2 participants. The method of clustering is used for data storage in IM and faster AA work.

<u>Hidden layer.</u> Hidden layer consists of specialized μ neurons. Input data of each μ neuron of the hidden layer are: input data (X); situation number α received from the affinity algorithm AA; signal β that indicates the need to recalculate the weights of μ neurons and is received from

the training algorithm TA. In the μ neuron, number of situation α is stored together with set of weights W_{μ} which were used while solving the exact problem, i.e. processing the similar input data. After number of the situation α is received, weights W_{μ} are chosen and training can be started. If there is no similar situation and $\alpha = 0$, then $W_{\mu} = 0$.

Output layer. Output layer consists of specialized μ neurons. Input data of each μ neuron of the output layer are: output data of the μ neurons of the hidden layer; situation number α received from the AA; signal β that indicates the need to recalculate the weights of μ neurons and is received from the TA. In the μ neuron, number of situation α is stored together with the set of weights W_{μ} that were used while solving the exact problem, similar as in the μ neuron of the hidden layer. After number of the situation α is received, weights W_{μ} are chosen and training can be started. If there is no similar situation and α = 0, then $W_{\mu} = 0$. Output data of the output layer: necessary horizontal movement direction change of the own UEV – $\Delta\tau_{UEV}$; necessary vertical movement direction change of the own UEV – $\Delta\varphi_{UEV}$; necessary speed change of the own UEV – Δv_{UEV}.

Target function (TF). Input data of TF: necessary horizontal movement direction change of the own UEV – $\Delta\tau_{UEV}$; necessary vertical movement direction change of the own UEV – $\Delta\varphi_{UEV}$; necessary speed change of the own UEV – Δv_{UEV}; input data obtained directly from UEV embedded electronic device (v_{UEV} – actual speed of the UEV; χ_{UEV} – latitude of the UEV actual position; ψ_{UEV} – longitude of the UEV actual position; η_{UEV} – altitude of the UEV actual position; θ_{UEV} – actual horizontal movement direction of the UEV; ω_{UEV} – actual vertical movement direction of the UEV). In current research, the location of the crossing points is a variable value, which makes the solution more complicated, as the found solution $< \Delta v_{UEV}, \Delta\tau_{UEV}, \Delta\varphi_{UEV} >$ has an influence on the distance to the crossing point. Thus, the evaluation of the TF requires the additional inputs $< v_{UEV}, \chi_{UEV}, \psi_{UEV}, \eta_{UEV}, \theta_{UEV}, \omega_{UEV} >$ obtained directly from the UEV to re-calculate the crossing point, the distance and time to it. TF calculates the collision probability P_{max}. Output data of TF: collision probability P_{max}; necessary horizontal movement direction change $\Delta\tau_{UEV}$; necessary vertical movement direction change $\Delta\varphi_{UEV}$; necessary speed change Δv_{UEV}.

Decision module (DM). Input data of DM: collision probability P_{max}, received from the TF; necessary horizontal movement direction change $\Delta\tau_{UEV}$, received from the TF; necessary vertical movement direction change $\Delta\varphi_{UEV}$, received from the TF; necessary speed change Δv_{UEV}, received from the TF. DM evaluates the found solution. If collision probability P_{max} is greater than acceptable (safe) collision probability P_{safe}, then the number of training iterations t is checked. If the number of training iterations t is less than maximal possible number of iterations T_{max}, it means the solution is not found yet and training must be repeated. DM sends signal to the training algorithm (TA). If the number of training iterations t is bigger or equal to maximal possible number of iterations T_{max}, it means that the situation cannot be solved in the defined time, so speed reduction is done. DM sends signal to the UEV embedded electronic device to stop the UEV, $v_{UEV} = 0$. If collision probability P_{max} is less or equal to acceptable (safe) collision probability P_{safe}, then the found solution $< \Delta v_{UEV}, \Delta\tau_{UEV}, \Delta\varphi_{UEV} >$ is sent to the UEV embedded electronic device and the match error ε_a between the current situation and situation chosen from the IM in the beginning of training is calculated. If the match error ε_a is bigger than the maximal possible match error ε_{lim}, responsible for creation of a new record in the IM or replacing the existing, then IM saves the situation as a new record and each μ neuron of the hidden and output layers saves the set of weights W_μ that was used for solving this situation together with a number of this situation α. If the match error ε_a is less or equal to the maximal possible match error ε_{lim}, then the record of the situation α is updated in the IM and values of weights W_μ of μ neurons of the hidden and output layers are updated according to the last used.

Training algorithm (TA). Input data of the TA: collision probability P_{max}, received from the TF; necessary horizontal movement direction change $\Delta\tau_{UEV}$, received from the TF; necessary vertical movement direction change $\Delta\varphi_{UEV}$, received from the TF; necessary speed change Δv_{UEV}, received from the TF; signal to repeat training, received from the DM. Training algorithm is used instead of traditional backpropagation algorithm. Backpropagation is typically used in supervised learning where the network is trained using labeled data, but the proposed novel INN is based on unsupervised learning. TA stores the last value of the P_{max}, which was received while solving this situation, and compares this value to the new one. TA sends signal β to all μ neurons, which means that training must be repeated. Signal β differs according to the result of the P_{max} comparison. If it is the first training iteration, TA does not have information about previous P_{max}, so TA sends signal β_1 to all μ neurons of the hidden and output layer. Signal β_1 means that the found solution does not solve the situation and training must continue. The same happens if the result of the found solution is better or equal to previous $P_{max2} \leq P_{max1}$. TA sends signal β_1 to all μ neurons of the hidden and output layers, which means that the found solution is not worse than previous one and training must continue. If the result of the found solution is worse than previous $P_{max2} > P_{max1}$, then TA sends signal β_2 to all μ neurons of the hidden and output layers. Signal β_2 means that the found solution does not solve the situation and the result of last iteration is worse than the result of the previous one. The values of the weights must be returned to the previous before training continues.

Training of μ neurons. When receiving β_1, new values of weights $W_{\mu j}$ are randomly chosen from the range ($W_{\mu j} - z \leq W_{\mu j+1} \leq W_{\mu j} + z$), where z is a predefined range parameter (may be adjustable). When receiving β_2, new values of weights $W_{\mu j}$ rollback to the previous values $W_{\mu j-1}$ and then are randomly chosen from the range ($W_{\mu j-1} - z \leq W_{\mu j} \leq W_{\mu j-1} + z$), where z is a predefined range parameter (may be adjustable).

2. Developed mathematical models

Mathematical models were developed and are presented in Chapter 2 of the Thesis.

Mathematical sets of system objects. Electrialc transport control system is defined by the following object classes: $UEVS = \{IO; SAT; UEV; DR\}$, where $UEVS$ – unmanned electric vehicle system structure; IO – infrastructure objects; SAT – satellite, component to obtain the coordinates of the electric vehicle position in real time; UEV – unmanned electric vehicles; DR – driver of the electric vehicle. Infrastructure objects: $IO = \{LC; RS; TL\}$, where LC – level crossings; RS – road signs; TL – traffic lights. Set of level crossings: $LC = \{LC_1; LC_2; \dots; LC_n\}$. Set of road signs: $RS = \{RS_1; RS_2; \dots; RS_n\}$. Set of traffic lights: $TL = \{TL_1; TL_2; \dots; TL_n\}$. Set of satellites: $SAT = \{SAT_1; SAT_2; \dots; SAT_n\}$. Set of unmanned electric vehicles: $UEV = \{UEV_1; UEV_2; \dots; UEV_n\}$. Set of drivers of the electric vehicles: $DR = \{DR_1; DR_2; \dots; DR_n\}$. UEV consists of: $UEV = \{SEN; GNSSR; RFM; ORM; INFN; DIS; DCS; CO; MECH\}$, where SEN – sensors to obtain the input data; $GNSSR$ – $GNSS$ signal receiver to obtain the $DATA_{GNSS}$; RFM – radio frequency module to obtain the $DATA_{RFM}$; ORM – object recognition module; $INFN$ – immune neuro-fuzzy module; DIS – driver informing system; DCS – driver control system; CO – electric vehicle control system; $MECH$ – mechanical part of the electric vehicle. Set of sensors: $SEN = \{DS; CAM; MDS; VS\}$, where DS – distance sensor to obtain the $DATA_{DS}$; CAM – videocamera to obtain the $DATA_{CAM}$; MDS – movement direction sensor to obtain the $DATA_{MDS}$; VS – speed sensor to obtain the $DATA_{VS}$. Data obtained by GNSS receiver: $DATA_{GNSS} = \{\eta_{tr}; \chi_{tr}; \psi_{tr}\}$, where η_{tr} – altitude ηη of the electric vehicles position; χ_{tr} – latitude χχ of the electric vehicles position; ψ_{tr} – longitude ψψ of the electric vehicles position. Data obtained by RFM: $DATA_{RFM} = \{\eta_{trn}; \chi_{trn}; \psi_{trn};\ \theta_{trn};\ \omega_{trn};\ v_{trn}\}$, where η_{trn} – altitude ηη of the other electric vehicles position; χ_{trn} – latitude χχ of the other electric vehicles position; ψ_{trn} – longitude ψψ of the other electric vehicles position; θ_{trn} – yaw angle of the flight of the other electric vehicle; ω_{trn} – pitch angle of the flight of the other electric vehicle; v_{trn} – other electric vehicles speed. Data obtained by DS: $DATA_{DS} = \{d_{tr}\}$, where d_{tr} – distance till the object. Data obtained by videocamera: $DATA_{CAM} = \{RGB;\ XY\}$, where RGB – red, green, blue pixels code; XY – position of the pixel. Data obtained by MDS: $DATA_{MDS} = \{\theta_{tr};\ \omega_{tr}\}$, where θ_{tr} – yaw angle of the flight of the electric vehicle; ω_{tr} – pitch angle of the flight of the electric vehicle. Data obtained by VS: $DATA_{VS} = \{v_{tr}\}$, where v_{tr} – electric vehicles speed. Object recognition module consists of: $ORM = \{MORA; SORA; RRA\}$, where $MORA$ – moving object recognition algorithm; $SORA$ – static object recognition algorithm; RRA – road recognition algorithm. MORA consists of: $MORA = \{DATA_{DS};\ DATA_{CAM}; CNN\}$, where $DATA_{DS}$ – data taken by distance sensor; $DATA_{CAM}$ – data taken by video camera; CNN – convolutional neural network. SORA consists of: $SORA = \{DATA_{DS};\ DATA_{CAM}; CNN\}$, where $DATA_{DS}$ – data taken by distance sensor; $DATA_{CAM}$ – data taken by video camera; CNN – convolutional neural network. RRA consists of: $RRA = \{DATA_{CAM}; CNN\}$, where $DATA_{CAM}$ – data taken by video camera; CNN – convolutional neural network. CNN consists of: $CNN = \{CONV_{CNN};\ POOL_{CNN};\ FC_{CNN}\}$, where $CONV_{CNN}$ – convolutional layer of the convolutional neural network; $POOL_{CNN}$ – pooling layer of the convolutional neural network; FC_{CNN} – fully-connected layer of the convolutional neural network. $CONV_{CNN}$ consists of: $CONV_{CNN} = \{K_{CNN}; F_{CNN}; S_{CNN}; P_{CNN}\}$, where K_{CNN} – number of convolutional neural network filters; F_{CNN} – convolutional neural network filter's spatial extent; S_{CNN} – stride; P_{CNN} – the amount of zero padding. $POOL_{CNN}$ consists of: $POOL_{CNN} = \{F_{CNN}; S_{CNN}\}$, where F_{CNN} – convolutional neural network filter's spatial extent; S_{CNN} – stride. FC_{CNN} consists of: $FC_{CNN} = \{HID_{CNN}; CL_{CNN}\}$, where HID_{CNN} – hidden layer of the convolutional neural network; CL_{CNN} – number of output classes. Immune neuro-fuzzy network consists of the following elements: $INFN = \{FL; IM; AA; NN; TF; DM\}$, where FL – fuzzy logic; IM – immune memory; AA – affinity algorithm; NN – neural network; TF – target function; DM – decision making algorithm. Control objects consist of: $CO = \{BS; ED; STS\}$, where BS – braking system; ED – electric drive; STS – steering system.

Mathematical model for traffic light red signal recognition task. This mathematical model was developed to distinguish the red colour signal from other colour signals. When the red colour is detected, it is assumed that the UEV will automatically reduce its speed to 0 km/h. This method is essential in ensuring the safety of electric vehicles. A description of the model can be found in Chapter 2 of the Doctoral Thesis.

Mathematical model for object recognition task. The proposed system structure works as follows: CAM videocamera receives the data about other objects (LC – level crossings, RS – road signs, TL – traffic lights, UEVn – other electric vehicles) by taking pictures. After these data are sent to the ORM object recognition module, based on the CNN convolutional neural network, where depending on the object type, object is recognized by MORA – moving object regognition algorithm, SORA – static object recognition algorithm, or RRA – road recognition algorithm.

Convolutional layer $CONV_{CNN}$ computes the output of neurons n that are connected to local regions in the input. Each neuron n computes a dot product between their weights W and a small region they are connected to in the input volume. The convolutional layer $CONV_{CNN}$ parameters consist of a set of learnable filters F_{CNN}. This layer requires four hyper-parameters: K_{CNN} – number of filters; F_{CNN} – filter's spatial extent; S_{CNN} – the stride; P_{CNN} – the amount of zero padding. Functions of the pool layer $POOL_{CNN}$ are: to reduce progressively the spatial size of the representation; to reduce the amount of parameters and computation in the network; and hence to also control overfitting. This layer requires two hyper-parameters: F_{CNN} – filter's spatial extent; S_{CNN} – the stride. Fully connected layer FC_{CNN} computes the class scores. Each neuron n in this layer will be connected to all the numbers in the previous volume as in ordinary NN. This layer requires two hyper-parameters: HID_{CNN} – number of hidden layer neurons; CL_{CNN} – number of output classes.

General structure of the CNN: INPUT [$W_{CNN}0 \times H_{CNN}0 \times D_{CNN}0$] $\rightarrow$ $CONV_{CNN}1$ [$K_{CNN}1$, $F_{CNN}1$, $P_{CNN}1$, $S_{CNN}1$] = OUT_{CNN} [$W_{CNN}11 = (W_{CNN}0 - F_{CNN} + 2P_{CNN}) / S_{CNN} + 1 \times H_{CNN}11 = (H_{CNN}0 - F_{CNN} + 2P_{CNN}) / S_{CNN} + 1 H_{CNN}11 \times K_{CNN}1 > D_{CNN}0$, $K_{CNN}1 / D_{CNN}0 = int$] $\rightarrow$ $POOL_{CNN}1$ [$F_{CNN}P_{CNN}1$, $S_{CNN}P_{CNN}1$] = OUT_{CNN} [$W_{CNN}12 = W_{CNN}11 / F_{CNN}P_{CNN}1 \times H_{CNN}12 = H_{CNN}11 / F_{CNN}P_{CNN}1 \times K_{CNN}1$]$\rightarrow$…$CONV_{CNN}n$ [$K_{CNN}n$, $F_{CNN}n_{CCN}n$, $P_{CNN}n$, $S_{CNN}nC_{CNN}n$] = OUT_{CNN} [$W_{CNN}n1 \times H_{CNN}n1 \times K_{CNN}n1$] $\rightarrow$ $POOL_{CNN}n$ [$F_{CNN}P_{CNN}n$, $S_{CNN}P_{CNN}n$] = OUT_{CNN} [$W_{CNN}n2 = W_{CNN}2 / F_{CNN}P_{CNN}n \times H_{CNN}n2 = H_{CNN}2 / F_{CNN}P_{CNN}n \times K_{CNN}n1$] $\rightarrow$ FC_{CNN}[HID_{CNN}, CL_{CNN}] = OUT_{CNN} [$1 \times CL_{CNN}$].

Mathematical model for possible crossing point detection and collision probability evaluation: As input data such parameters of each vehicle are used: *LFn* – coordinates of the left front angle of the vehicle; *RFn* – coordinates of the right front angle of the vehicle; *LRn* – coordinates of the left rear angle of the vehicle; *RRn* – coordinates of the right rear angle of the vehicle; *Vn* – speed of the vehicle; *Tn* – trajectory of the motion of the vehicle; *Ln* – length of the vehicle.

The model finds out whether the object is in the control area or not after all the input data is selected. We propose that Object 2 is in the control area of Object 1 only if Object 2 is in front of Object 1 or on the same level. If Object 2 is in the control area of Object 1, then calculations are done, otherwise no calculations are needed. As the coordinates of left and right angles of vehicles are known, formulas of the straight lines of the left and right sides are calculated and crossing points of these lines are detected.

The next step is to detect the minimal and maximal distances till the possible crossing point. The minimal distance till the crossing point is calculated as follows: For Object 1: Coordinates of the crossing point (R1; L2) minus coordinates of the right front angle of Vehicle 1 minus half of the length of Vehicle 1: $Dist1min = (R1;\ L2) - (RF1) - (L1/2)$. For Object 2: Coordinates of the crossing point (R1; L2) minus coordinates of the left front angle of Vehicle 2 minus half of the length of Vehicle 2: $Dist2min = (R1; L2) - (LF2) - (L2/2)$. The maximal distance till the crossing point is calculated as follows: For Object 1: Coordinates of the crossing point (L1; R2) minus coordinates of the left front angle of Vehicle 1 plus half of the length of Vehicle 1: $Dist1max = (L1; R2) - (LF1) + (L1/2)$. For Object 2: Coordinates of the crossing point (L1; R2) minus coordinates of the right front angle of Vehicle 2 plus half of the length of Vehicle 2: $Dist2max = (L1; R2) - (RF2) + (L2/2)$.

After distances are calculated, minimal and maximal time till the crossing point is calculated for both objects: $Timenmin = Distnmin\ /\ Vn$; $Timenmax = Distnmax\ /\ Vn$.

The algorithm of the electric vehicle collision risk solving system has been proposed based on basic collision risk and vulnerabilities of accidents [64]. The proposed algorithm is available in the Doctoral Thesis.

Mathematical model for the neural network. The mathematical model is represented by the following sets: $U \subset (U_1, \dots, U_n)$ – a set of transport units as subsets of different types that could be different for different transport safety tasks, e.g., $U^1 = (U_1^1, \dots, U_{n1}^1)$ – a subset of railway transport units; $U^2 = (U_1^2, \dots, U_{n2}^2)$ – a subset of road vehicles; $U^3 = (U_1^3, \dots, U_{n3}^3)$ – a subset of aerial vehicles, etc. $P = (p_1, p_2, \dots, p_c)$ is a set of infrastructure objects where the collision of vehicles takes place, e.g., for railway transport, it could be level-crossings, switches, etc. For this research, the crossing section is assumed as a short straight segment of the route or trajectory.

The geographical coordinates of all crossing of possible routes or trajectories of transport units are defined by these sets:

$$\chi_b^p = \{\chi_b^{p_1}, \chi_b^{p_2}, \dots, \chi_b^{p_c}\}, \psi_b^p = \{\psi_b^{p_1}, \psi_b^{p_2}, \dots, \psi_b^{p_c}\};$$

$$\chi_e^p = \{\chi_e^{p_1}, \chi_e^{p_2}, \dots, \chi_e^{p_c}\}, \psi_e^p = \{\psi_e^{p_1}, \psi_e^{p_2}, \dots, \psi_e^{p_c}\},$$

where

$\chi_b^{p_i}$ – latitude of the beginning point of the crossing sector;

$\psi_b^{p_i}$ – longitude of the beginning point of the crossing sector (crossing);

$\chi_e^{p_i}$ – latitude of the ending point of crossing sector;

$\psi_e^{p_i}$ – longitude of the ending point of the crossing sector;

c – number of crossing points on the trajectory;

t^p_{vest} – safe closing time of each trajectory crossing point $p \in P^2$.

There is no information on whether the output value is correct or not, that is why there is no possibility to use error backpropagation algorithm.

A random sequential delta law self-training algorithm and target function were developed for the neural network training.

The function of optimization is defined by two criteria: collision possibility P with the aim of minimizing; changes of the vehicles' speed ΣΔvi with the aim of minimizing.

The first criterion is related to safety. The situation considered to be dangerous if trajectories of two transport vehicles have a common crossing point and there exists a probability that transport vehicles will arrive at the crossing point of their trajectories at the same time.

The second criterion is related to the specific characteristics of transport traffic, such as the times of departure and arrival. That is why it is necessary to make minimal speed changes of such type of vehicles.

Based on the individual weighted criteria, the target function was developed:

$$F(\Delta v) = \begin{cases} P = \max(P_{IJ}) \rightarrow min \\ \sum \Delta v_i \rightarrow min \end{cases},$$

where Δv is change of the speed of vehicle; P is maximal collision probability; P_{IJ} is each i-th vehicle collision possibility with each j-th vehicle; and Δv_i is the change of speed of the i-th vehicle.

Mathematical model for the immune neural network. The immune neuro-fuzzy module (INFM) calculates the necessary movement parameters change of the UEV for the collision probability minimization by using the immune neural network after the collision probability has been recognized.

Each UEV analyses the situation for itself. UEVs in the control area are detected. These UEVs are ordered for a more precise definition of the situation. For ordering following parameters are used: the first UEV in the input data (X) always is UEV itself; three parameters are used for ordering other UEVs: the horizontal movement direction τ_{UEVn}; the vertical movement direction φ_{UEVn}; and the distance till the crossing point d_{UEVn}.

Ordering of other UEVs is done according to the slope τ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of τ_{UEVn}, then ordering of these UEVs is done according to the slope φ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of φ_{UEVn}, then ordering of these UEVs is done according to the distance to the crossing point with these UEV d_{UEVn}.

Training process of the proposed INN depends on the current situation s_j that is solved. Current situation is different for each situation participant, because each UEV sets different order of UEVs according to the own position. A set of situations stored in the immune memory (IM) is represented as follows: $\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \dots, \mathrm{s}_\mathrm{m}\}$.

There are no identical situations stored in the IM because of verification, which defines either this is the same situation or it is a new situation. If the situation is the same, data of this situation can be updated. If it is a new situation, new record of the situation occurs in the IM.

Each situation s_j contains input data X_j and the number of participants n_j. All the data in IM are stored in clusters for easier and faster match finding process. For example, if three vehicles are participating in the possible collision situation, there is no need to find a similar situation in the group of situations with two participants. Therefore, method of clustering is used for data storage in IM and faster AA work. Situation s_j: $s_j =< X_j >$, $S_k \subseteq S, s_j \in S_k, |s_j| = |X|$, where X_j is input data; S_k is a subset of all situations stored in IM and contains only those situations where the dimensions of the situation s_j data X_j are the same as the dimensions of the given situation X.

The proposed INN can be made of one or multiple layers, depending on the task that is solved. The INN, proposed in this research, consists of an input layer, one hidden layer, and an output layer.

The input layer consists of input data X that describes the situation. The situation for n-th UEV in general may be described by the set of the following subsets:

$X = (v_0, v_1, \tau_1, \varphi_1, d_1 \dots, v_n, \tau_n, \varphi_n, d_n) = (x_0, x_1, x_2, x_3, x_4, \dots, x_{4n-3}, x_{4n-2}, x_{4n-1}, x_{4n})$,

where n –is the number of other vehicles ($n = 0$ is own vehicle; $n > 0$ is all others vehicles); v_n, x_{4n-3} is speed of the n-th electric vehicle; and τ_n, x_{4n-2} is horizontal movement direction of the n-th electric vehicle. The direction of movement of another electric vehicle ($n > 0$) is relative to one's own ($n = 0$) direction, but $\tau_0 = 0$; φ_n, x_{4n-1}– vertical movement direction of the n-th electric vehicle. The direction of movement of another electric vehicle ($n > 0$) is relative to one's own ($n = 0$) direction, but $\varphi_0 = 0$; d_n, x_{4n} is distance till the possible crossing point of own vehicle with another vehicle's $n > 0$ trajectory. Thus $d_0 = 0$.

Discrepancies between input data and data of the situation that is stored in the IM is represented as follows: $\mathcal{E} = \{\varepsilon_1, \dots, \varepsilon_k\}$, where $\varepsilon_j = |X - X_j| = \sum |x_i - x_{ij}|, X_j \in S_k$.

A situation with the smallest discrepancy – closest match – is represented as follows: $\varepsilon_\alpha = \min(\varepsilon)$.

The hidden layer is represented as follows: $\mu_{HID} = \{\mu_1, \dots, \mu_c\}$, where μ is specialized μ_h neuron. Specialized μ_h neuron of the hidden layer is represented by following subsets: $\mu_h = \{I_{\mu h}, W_{\mu h}, AF_{\mu h}, O_{\mu h}\}$, where $I_{\mu h}$ is input of the μ_h neuron; $W_{\mu h}$ are weights of the μ_h neuron; $AF_{\mu h}$ is activation function of the μ_h neuron; and $O_{\mu h}$ is output of the μ_h neuron. Input data of μ_h neuron of the hidden layer is represented by the following subsets: $I_{\mu h} = \{X, \alpha, \beta\}$, where α is the number of the situation with a smallest discrepancy ε_α; and β is the signal received from training algorithm TA. Each μ_h neuron of the hidden layer stores weights for all situations stored in IM. Number of the weights of the hidden layer depends on the amount of participants n in the situation plus additional weight b_i, which is also related to the situation. Set of weights of the μ_h neuron of the hidden layer is represented as follows: $W_{\mu h} = \{< \alpha_1, W_1 >, \dots, < \alpha_m, W_m >\}$, where $W_i = (w_{0i}, w_{1i}, w_{2i}, \dots, w_{4ni}, b_i)$, where i is the index of μ_h neuron. A random number z is generated and the weight coefficient is shifted during the training process to receive new values of weights W_{ih}: $w_{ji}^{t-1} - z \le w_{ji}^{t} \le w_{ji}^{t-1} + z$. Activation function of the μ_h neuron of the hidden layer is represented as a pure linear function: $O_{\mu h} = AF_{\mu h}(X, W_i) = \sum_{j=0}^{n_i*4} x_j w_{ji} + b_i$.

Output layer is represented as follows: $\mu_{OUT} = \{\mu_1, \dots, \mu_d\}$, where μ is specialized μ_p neuron. Number of specialized μ_p neurons of the output layer depends on the number of unknowns in the solved task. There are three unknowns in the task proposed in this research

($\Delta v_{UEV}, \Delta\tau_{UEV}, \Delta\varphi_{UEV}$), so the formula of the output layer for the collision prevention task of the Thesis is represented as $\mu_{OUT} = \{\mu_1, \mu_2, \mu_3\}$. Specialized μ_p neuron of the output layer is represented the by following subsets: $\mu_p = \{I_{\mu p}, W_{\mu p}, AF_{\mu p}, O_{\mu p}\}$, where $I_{\mu p}$ – input of the μ_p neuron; $W_{\mu p}$ – weights of the μ_p neuron; $AF_{\mu p}$ – activation function of the μ_p neuron; $O_{\mu p}$ – output of the μ_p neuron. Number of the inputs of the output layer depends on the number of μ_h neurons in the hidden layer. Input data of the μ_p neuron is represented by the following subsets: $I_{\mu p} = \{O_{\mu h}, \alpha, \beta\}$, where $O_{\mu h}$ – output data of the μ_h neurons of the hidden layer; α – number of the situation with the smallest discrepancy ε_α; β – the signal received from training algorithm TA. Number of the weights of the output layer depends on the number of μ_h neurons of the hidden layer plus additional weight b_i, which is also related to the situation. The set of weights of the μ_p neuron of the output layer is represented as follows: $W_{\mu p} = \{< \alpha_1, W_1 >, \ldots, < \alpha_m, W_m >\}$, where $W_i = (w_{0i}, w_{1i}, w_{2i}, \ldots, w_{4ni}, b_i)$. Type of the activation function of μ_p neuron of the output layer depends on the solved task. Different types of functions can be used according to the desired result: $O_{\mu p} = AF_{\mu p}(O_{\mu h}, W_i) = f(r)$, where $r = \sum_{j=0}^{h} y_j w_{ji} + b_i$, where n – the change limit of speed Δv_i in horizontal movement direction $\Delta\tau_i$ or vertical movement direction $\Delta\varphi_i$. Logarithmic sigmoid function is used as activation function of μ_p neuron of the output layer in this research: $O_{\mu p} = AF_{\mu p}(O_{\mu h}, W_i) = \log(\frac{1}{1+e^{-r}})$. This type of function is used because it is possible to set limits for speed and direction changes. It is important because different types of vehicles have different parameters and limits of speed and trajectory change. However, limits of the speed and distance changes are checked by the proposed target function also. Outputs of the μ_p neurons also depend on the solved task. For example, rail transport cannot change direction in any moment of time. Only the speed change can be done. As a result, there will be only one μ_p neuron and only one input $O_{\mu p}$. Three outputs of the output layer are proposed in this research: $O_{\mu p1} = \Delta v_{UEV}$; $O_{\mu p2} = \Delta\tau_{UEV}$; $O_{\mu p3} = \Delta\varphi_{UEV}$, where Δv_{UEV} – necessary speed change of the own UEV; $\Delta\tau_{UEV}$ – necessary horizontal movement direction change of the own UEV; and $\Delta\varphi_{UEV}$ – necessary vertical movement direction change of the own UEV.

Target function was proposed to define the interest of the UEV: $TF = w(x, a_1, a_2, \ldots, a_n) \rightarrow opt$, where *TF* – target function – the objective of UEV; *x* – state of the environment; and a_i – action of the i-th UEV. The ultimate goal of the target function is to minimize the collision probability of UEV by minimal changes of the speed and direction in the given state of environment. The target function evaluates the state of the environment and then assesses how the actions of the UEV will impact the situation.

The information of each UEV depends on the state of environment: $y_i = y_i(x)$, where y_i is information of the i-th UEV.

The decision rule of the i-th UEV results in an action of i-th electric vehicle and depends on information $a_i = \rho_i(y_i)$, where ρ_i is the decision rule of the *i*-th UEV.

Interaction between the *i*-th and *j*-th UEV: $q_{ij} = \partial w / \partial a_i \partial a_j$.

A set of decision rules is optimal if $E(w(x, (\rho_1(y_1), \ldots, \rho_n(y_n)))) \rightarrow \max$ for a given probability distribution on x.

For anti-collision test, the set of possible points of potential collisions is defined: $P = (p_1, p_2, \ldots, p_c)$.

The location L^{UEV} of UEV is represented by three subsets $<\chi_c^{UEVS}, \psi_c^{UEVS}, \eta_c^{UEVS}>$, whicht are latitutde χ, longitude ψ, and altitude η: $\chi_c^{UEV} = \{\chi_c^{UEV_1}, \chi_c^{UEV_2}, \ldots, \chi_c^{UEV_n}\}$; $\psi_c^{UEV} = \{\psi_c^{UEV_1}, \psi_c^{UEV_2}, \ldots, \psi_c^{UEV_n}\}$; $\eta_c^{UEV} = \{\eta_c^{UEV_1}, \eta_c^{UEV_2}, \ldots, \eta_c^{UEV_n}\}$, where χ_c^{UEV} – latitude of the current electric vehicle point; ψ_c^{UEV} – longitude of the current electric vehicle point; and η_c^{UEV} – altitude of the current electric vehicle point.

Horizontal movement direction of the other electric vehicle is used as an input data. The horizontal movement direction θ^{UEV} of UEVs is represented as follows: $\theta^{UEV} = \{\theta^{UEV1}, \theta^{UEV2}, \ldots, \theta^{UEVn}\}$.

Vertical movement direction of the other electric vehicle is used as an input data. The vertical movement direction ω^{UEV} of UEVs is represented as follows: $\omega^{UEV} = \{\omega^{UEV1}, \omega^{UEV2}, \ldots, \omega^{UEVn}\}$.

The safety criterion is the following: $D = |UEV_i UEV_j| = \sqrt{(\chi_c^j - \chi_c^i)^2 + (\psi_c^j - \psi_c^i)^2 + (\eta_c^j - \eta_c^i)^2} > D_{safe}$, where D_{safe} is the safety distance limit for each pair of $< UEV_i, UEV_j >$, $i = 1..n$, $j = 1..n$, $i \neq j$.

Permissible changes of direction depend on the UEV specifications and other circumstances. Restrictions for the own horizontal movement direction change were also defined: $\tau_1^{UEV_i} < \tau^{UEV_i} < \tau_2^{UEV_i}$. Restrictions for the own movement direction (in vertical plane) change were also defined: $\varphi_1^{UEV_i} < \varphi^{UEV_i} < \varphi_2^{UEV_i}$. Restrictions for the speed change were also defined: $v_1^{UEV_i} < v^{UEV_i} < v_2^{UEV_i}$.

$$\begin{cases} P_{max}(\chi_c^{UEV}, \psi_c^{UEV}, \eta_c^{UEV}, \Delta\tau, \Delta\varphi, \Delta v) = max(P_{IJ}) \to min \\ \Delta\tau_\Sigma(\Delta\tau) = \sum_{i=1}^{n} \Delta\tau_i \to min \\ \Delta\varphi_\Sigma(\Delta\varphi) = \sum_{i=1}^{n} \Delta\varphi_i \to min \\ \Delta v_\Sigma(\Delta v) = \sum_{i=1}^{n} \Delta v_i \to min \\ D = |UEV_i UEV_j| > S \\ \Delta\tau_1 < \Delta\tau_i < \Delta\tau_2 \\ \Delta\varphi_1 < \Delta\varphi_i < \Delta\varphi_2 \\ \Delta v_1 < \Delta v_i < \Delta v_2 \\ i = 1..n, j = 1..n, i \neq j \end{cases} \tag{2.1}$$

The general target function with anti-collision criteria is provided in Formula (2.1), where: P_{max} – maximal collision probability from the set of probabilities of collision for all pairs of UEVs; $\Delta\tau = (\Delta\tau_1, \ldots, \Delta\tau_n)$ – set direction changes in horizontal plane of all UEVs; $\Delta\varphi = (\Delta\varphi_1, \ldots, \Delta\varphi_n)$ – set direction changes in vertical plane of all UEVs; $\Delta v = (\Delta v_1, \ldots, \Delta v_n)$ – set of speed changes of all UEVs; $P_{IJ} = (P(<UEV_1, UEV_2>),\ldots,P(<UEV_i, UEV_j>),\ldots,P(<UEV_{n-1}, UEV_n>))$ – set of probabilities of collision for all pairs of UEVs $<UEV_i, UEV_j>$, $i \neq j$, $i,j = 1..n$.

Each i-th UEV is looking for its own direction and/or speed change solution $< \Delta\tau_i, \Delta\varphi_i, \Delta v_i >$ according to the task. The target function for a single UEV can be expressed as shown in Formula 2.2, where P_{max} represents the highest probability of collision between the own UEV_0 and all other UEVs within the control area; $\Delta\tau_0$ is direction change in horizontal plane of the own UEV_0; $\Delta\varphi_0$ is direction change in vertical plane of the own UEV_0; Δv_0 is speed change of the own UEV_0; $P_{0j} = (P(<UEV_0, UEV_1>),\ldots,P(<UEV_0, UEV_j>),\ldots, P(<UEV_0, UEV_n>))$ is a set of probabilities of collision between the own UEV_0 and all other UEVs within the control area, $j = 1..n$.

$$\begin{cases} P_{max}(\chi_c^{UEV}, \psi_c^{UEV}, \eta_c^{UEV}, \Delta\tau_0, \Delta\varphi_0, \Delta v) = \max(P_{0j}) \to min \\ \Delta\tau_0 \to min, \Delta\varphi_0 \to min, \Delta v_0 \to min \\ D = \left|UEV_0 UEV_j\right| > S \\ \Delta\tau_1 < \Delta\tau_0 < \Delta\tau_2, \Delta\varphi_1 < \Delta\varphi_0 < \Delta\varphi_2, \Delta v_1 < \Delta v_0 < \Delta v_2 \\ j = 1..n \end{cases} \quad (2.2)$$

Function of the decision-making module F_{DM} is represented as follows: $F_{DM} = TF(P_{max}, \Delta v_{UEV}, \Delta\tau_{UEV}, \Delta\varphi_{UEV}) \to min.$

Thus, we can evaluate the result of training the INN without a teacher with the help of the proposed target function and make a decision about accepting the solution or continue the training.

3. Developed algorithms for dangerous situation recognition and prevention of electric vehicles

Developed algorithms for different tasks of the research are provided in the Chapter 3 of the Doctoral Thesis: algorithm for recognition of red signal light; algorithm for CNN for object recognition for dangerous situation recognition and prevention of electric transport; algorithm for collision probability evaluation of electric transport; algorithm for neural network for collision probability evaluation and minimization; algorithm for novel INN for unsupervised collision probability evaluation and minimization for dangerous situation recognition and prevention of electric vehicles.

Algorithm for recognition of the red traffic light signal for dangerous situation recognition and prevention of electric transport. This algorithm was developed to distinguish the red traffic light signal from the signals of other colors.The proposed algorithm can be found in Chapter 3 of the Doctoral Thesis.

Algorithm for convolutional neural network for object recognition for dangerous situation recognition and prevention of electric transport. CNN is a traditional neural network that is used in this research for object recognition tasks. It is an essential component in ensuring the safety of electric vehicles. A detailed description of the algorithm can be found in Chapter 3 of the Doctoral Thesis.

Algorithm for collision probability evaluation of electric transport. STEP 1: Data obtaining. Vehicle receives the information about own parameters: speed and coordinates of the center point by using GNSS and other vehicles speed and coordinates of the four angles of the vehicle by using RF. STEP 2: The necessity of future calculations is detected after the information has been received. STEP 2.1: Calculations of the front and rear angle coordinates of the vehicle are done: $RF = (\chi_{UEV} + \frac{w_{UEV}}{2};\ \psi_{UEV} + \frac{h_{UEV}}{2})$; $\mathrm{LF} = (\chi_{UEV} - \frac{w_{UEV}}{2};\ \psi_{UEV} + \frac{h_{UEV}}{2})$; $RR = (\chi_{UEV} + \frac{w_{UEV}}{2};\ \psi_{UEV} - \frac{h_{UEV}}{2})$; $LR = (\chi_{UEV} - \frac{w_{UEV}}{2};\ \psi_{UEV} - \frac{h_{UEV}}{2})$, where RF – coordinates of the right front angle of the vehicle; LF – coordinates of the left front angle of the vehicle; RR – coordinates of the right rear angle of the vehicle; LR – coordinates of the left rear angle of the vehicle; χ_{UEV} – latitude of the center point of the vehicle; ψ_{UEV} – longitude of the central point of the vehicle; w_{UEV} – width of the vehicle; h_{UEV} – length of the vehicle. STEP 2.2: Movement direction calculation. Front and rear angles are compared for this purpose. STEP 2.3: Detecting objects in the control area – in front of the vehicle or on the same level. For this purpose, coordinates of the vehicle's angles are compared, similar as it was done in STEP 2.2. If the object is not in the control area, then no further calculations are needed, otherwise STEP 3 is performed. STEP 3: Calculation of coordinates of the crossing point: $yR_1 = k_1 * xR_1 + bR_1$; $yL_1 = k_1 * xL_1 + bL_1$; $yR_2 = k_2 * xR_2 + bR_2$; $yL_2 = k_2 * xL_2 + bL_2$; $yyR1R2 = cross(yR_1,\ yR_2)$; $yyR1L2 = cross(yR_1,\ yL_2)$; $yyL1R2 = cross(yL_1,\ yR_2)$; $yyL1L2 = cross(yL_1,\ yL_2)$, where yR_1 – line of the right side of UEV1; yL_1 – line of the left side of UEV1; yR_2 – line of the right side of UEV2; yL_2 – line of the left side of UEV2; k_1, k_2 – angle coeficients; $yyR1R2$ – coordinates of the right rear angle of the crossing area; $yyR1L2$ – coordinates of the left rear angle of the crossing area. Only χ_{UEV} coordinates are considered for calculating the crossing point, because according to the algorithm, own ψ_{UEV} coordinate of each vehicle is equal to zero and only χ_{UEV} coordinate can be changed. STEP 4: Calculation of distance till the crossing point is done: $ddR1R2 = \sqrt{(yyR_1R_2 - yRR_1)^2}$; $ddL1R2 = \sqrt{(yyL_1R_2 - yLR_1)^2}$; $ddR1L2 = \sqrt{(yyR_1L_2 - yRF_1)^2}$; $ddL1L2 = \sqrt{(yyL_1L_2 - yLF_1)^2}$. STEP 4.1: Calculation of the minimal distance till the crossing point: $d_1^{min} = (ddR1L2, ddL1L2)$. To detect the minimal distance, it is necessary to calculate the distance for the vehicle's left and right sides. The smallest value will be the minimal distance. STEP 4.2: Calculation of the maximal distance till the crossing point: $d_1^{max} = (ddR1R2, ddL1R2)$. To detect the maximal distance, it is necessary to calculate the distance for the vehicle's left and right sides. The biggest value will be the maximal distance. STEP 5. Calculation of the time till the crossing point. STEP 5.1: Calculation of the minimal time till the crossing point: $t_n^{min} = d_1^{min} / V_n$, where V_n is speed of the vehicle. STEP 5.2: Calculation of the maximal time till the crossing point: $t_n^{max} = d_1^{max} / V_n$. STEP 6: Evaluation of collision probability is done: **If** $t_2^{min} > t_1^{min}$ **and** $t_1^{max} > t_2^{max}$ **and** $t_2^{max} > t_1^{min}$, **then** $P1 = \frac{t_1^{max} - t_2^{min}}{t_1^{max} - t_1^{min}}$; $P2 = \frac{t_1^{max} - t_2^{min}}{t_2^{max} - t_2^{min}}$; $P = P1 * P2$. **If** $t_2^{min} < t_1^{min}$ **and** $t_2^{max} < t_1^{max}$ **and** $t_1^{min} < t_2^{max}$, **then** $P1 = \frac{t_2^{max} - t_1^{min}}{t_2^{max} - t_2^{min}}$; $P2 = \frac{t_2^{max} - t_1^{min}}{t_1^{max} - t_1^{min}}$; $P = P1 * P2$. **If** $t_1^{min} < t_2^{min}$ **and** $t_1^{max} < t_2^{min}$ **or** $t_2^{min} < t_1^{min}$ **and** $t_2^{max} < t_1^{min}$, **then** P = 0. **If** $t_1^{min} <= t_2^{min}$ **and** $t_1^{max} >= t_2^{max}$, **then** $P = \frac{t_2^{max} - t_2^{min}}{t_1^{max} - t_1^{min}}$.

If $t_2^{min} <= t_1^{min}$ **and** $t_2^{max} >= t_1^{max}$, **then** $P = \frac{t_1^{max}-t_1^{min}}{t_2^{max}-t_2^{min}}$. Depending on the task, it is possible to transform the collision probability value into fuzzy values. However, in the provided novel INN algorithm, fuzzy logic was not taken into consideration.

Algorithm for neural network for collision probability evaluation and minimization for dangerous situation recognition and prevention of electric transport: Initialization: Index of training group element e = 1; chosen for the correction n sn = 1; chosen for the correction weight sw = 1; retraining = false. STEP 1: Take element e = {d_{tr}^e, v_{tr}^e, d_{tr}^e, v_{tr}^e} from the training set. STEP 2: x = (e_1, e_2, …, e_n). STEP 3: Read the x_{min} and x_{max} parametres, which limit the n network output. STEP 4: Calculate output n adder values: $\sum_j = (\sum_{i=1}^{2n} x_i * w_{ij}) + b_j$, $j = \overline{1..n}$. Generate output layer n output value by positively and negatively saturated linear activation function: $\Delta v_{tr} = \begin{cases} x_{min}, & \sum_j \le x_{min} \\ \sum_j, & x_{min} < \sum_j < x_{max} \\ x_{max}, & \sum_j \ge x_{max} \end{cases}$. STEP 5: Save the previous valuation, if it exists P_{tr0}, $\Sigma\Delta v_{tr0}$. STEP 6: Evaluate the solution that was found using the target function [P_{tr}, $\Sigma\Delta v_{tr}$] = TF(Δv_{tr}). STEP 7: If $P_{tr} > P_{safe}$ or $\Sigma\Delta v_{tr} > \Sigma\Delta v_{trlim}$, then proceed to STEP 8. STEP 8: If the last element of the training set is not reached, $e \neq e_{max}$, then e = e + 1 and go to STEP 1, else if there is no need to retrain the network, then FINISH, else e = 1 and go to STEP 1. STEP 9: Weight correction occurs sequentially: If (sn ≠1 and sw ≠ 1) or ($P_{tr0} < P_{tr}$ and $\Sigma\Delta v_{tr0} < \Sigma\Delta v_{tr}$), that means if the element is not first and the result is worse than it was before, then weight correction is done, $w_{sw,\,sn} = w_{sw,\,sn} - k$, where *k* is a random number, k = random (xmin, xmax)/10 000. If sw > 2n, then sn = sn + 1, else sw = sw + 1. If sn > n, then sn = 1, sw = 1. STEP 10: If weight correction was done, then neural network must be retrained. Retrain = true. STEP 11: Go to STEP 3.

Novel algorithm for immune neural network for unsupervised collision probability evaluation and minimization for dangerous situation recognition and prevention of electric vehicle. STEP 1: Receive input data DAT from n UEVs located in the area of visibility. These data are locations <$\chi_c^{UEV}, \psi_c^{UEV}, \eta_c^{UEV}$>, speed v^{UEV}, horizontal movement direction θ^{UEV}, and vertical movement direction ω^{UEV} of UEV: DAT = ($\chi_c^{UEV}, \psi_c^{UEV}, \eta_c^{UEV}, \theta^{UEV}, \omega^{UEV}, v^{UEV}$), where χ_c^{UEV} – latitude of the UEV actual position; ψ_c^{UEV} – longitude of the UEV actual position; η_c^{UEV} – altitude of the UEV actual position; θ^{UEV} – actual horizontal movement direction of the UEV; ω^{UEV} – actual vertical movement direction of the UEV; v^{UEV} – actual speed of the UEV. STEP 2: The proposed INN requires data about other vehicles' location in relation to the own UEV location. DAT data needs to be proceeded before it will enter the input layer of the proposed INN. STEP 2.1: Input data DAT contains coordinates of other vehicles' position <$\chi_c^{UEV}, \psi_c^{UEV}, \eta_c^{UEV}$ >, own position is known as well. Distances to the possible crossing points with other vehicles d_{UEV} are calculated. Algorithm for evaluation of collision probability of electric transport is proposed above. Only minimal distance d_n^{min} is calculated in this step. STEP 2.2: The step is to organize the UEVs for a more precise definition of the situation. The first UEV is always UEV itself. The other UEVs are ordered according to their horizontal movement direction τ_n^{UEV} in relation to the own UEV, starting from 0 degrees and proceeding clockwise. If multiple UEVs have the same value of τ_n^{UEV}, they are then ordered according to their vertical movement direction φ_n^{UEV} in relation to the own UEV, starting from 0 degrees and proceeding clockwise. If multiple UEVs have the same value of φ_n^{UEV}, they are then ordered according to the distance to the crossing point with the own UEV, d_n^{UEV}. Horizontal movement directions of other UEVs τ_n in relation to the own UEV direction are calculated as follows: $\tau_n = \tan^{-1}((\tan\theta_n^{UEV} - \tan\tau_1)/(1+\tan\tau_1 \tan\theta_n^{UEV}))$, where τ_1 – direction of the own UEV in horizontal plane; θ_n^{UEV} – direction of other UEV in horizontal plane. Vertical movement directions of other UEVs φ_n in relation to the own UEV direction are calculated as follows: $\varphi_n = \tan^{-1}((\eta_n^{\mathrm{UEV}} - \eta_1)/d_n)$, where η_1 – altitude of the own UEV; η_n^{UEV} – altitude of other UEV; d_n – horizontal distance between own UEV and the n-th UEV. Horizontal distance between own UEV and the n-th UEV: $d_n = \sqrt{(\Delta\eta^2 + \Delta d^2)}$, where $\Delta\eta$ – is the difference in altitude between the two UEVs; Δd – is the horizontal distance between the two UEVs. Actions provided in STEP 2 transform input data DAT to input data X: $X = (v_0, v_1, \tau_1, \varphi_1, d_1 \dots, v_n, \tau_n, \varphi_n, d_n) = (x_0, x_1, x_2, x_3, x_4, \dots, x_{4n-3}, x_{4n-2}, x_{4n-1}, x_{4n})$, where *n* – number of other vehicles, *n = 0* – own vehicle, *n > 0* – all others vehicles; v_n, x_{4n-3} – speed of the n-th electric vehicle; τ_n, x_{4n-2} – horizontal movement direction of the n-th electric vehicle. Direction of movement of another electric vehicle (*n > 0*) is relative to one's own (*n = 0*) direction, but $\tau_0 = 0$; φ_n, x_{4n-1}– vertical movement direction of the n-th electrical vehicle. Direction of movement of another electric vehicle (*n > 0*) is relative to one's own (*n = 0*) direction, but $\varphi_0 = 0$; d_n, x_{4n} – distance till the possible crossing point of own vehicle with another vehicle's *n > 0* trajectory. Thus $d_0 = 0$. STEP 3: The calculation of collision probability P is intended to determine whether it is necessary to minimize the risk of collision. If no, end of the algorithm. If yes, then go to the next step. Algorithm for collision probability evaluation task of electric transport is proposed above. STEP 4: After input data X enters the input layer of INN, data X is sent to the specialized μ neurons and affinity algorithm (AA). The AA (X, S) checks all situations S stored in the IM $\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \dots, \mathrm{s}_\mathrm{m})$, calculates the set of discrepancies $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_\mathrm{k})$, where $\varepsilon_\mathrm{j} = \sum_{\mathrm{i}=0}^{\mathrm{n}} \sum_{\mathrm{k}=1}^{2} \left(\frac{X_{\mathrm{ik}} - X_{\mathrm{ik}}^{\mathrm{j}}}{X_{\mathrm{ik}}}\right)^2$, and finds the closest match ε_α, where $\varepsilon_\alpha = \min(\varepsilon)$. STEP 5: When the μ neuron receives input data X, it activates and increases iteration counter $t = t + 1$. When situation number α is received, set of weights W_μ are selected from the memory of the μ neuron. If there is no similar situation in the IM and $\alpha = 0$, then $W_\mu = 0$. STEP 6: Input data X, situation number α, received from the affinity algorithm AA, and signal β, which indicates the need to recalculate the weights of μ neurons, are the input data of each μ neuron of the hidden layer μ_{HID}. Feed forward input through the NN is done. Formulas are provided in Chapter 2. Outputs for own vertical movement direction change $O_{\mu p3} = \Delta\varphi_{UEV}$, own horizontal movement direction change $O_{\mu p2} = \Delta\tau_{UEV}$ and own speed change $O_{\mu p1} = \Delta v_{UEV}$ are generated as a result. STEP 7: TF calculates the collision probability P_{max} that is maximal collision probability from the set of probabilities of collision for all pairs of UEVs P_{IJ}. TF uses updated data received directly from the UEV embedded device D_{TR}. STEP 7.1: The TF function defines the directions τ^{UEV} and φ^{UEV} of each UEV in relation to the own UEV. STEP 7.2: Next step is to detect the crossing point $(\chi_\mathrm{p}, \psi_\mathrm{p}, \eta_p)$ in 3D space. STEP 7.3: If the crossing point $(\chi_\mathrm{p}, \psi_\mathrm{p}, \eta_p)$ is found and is located on the way of motion, then go to STEP 7.4.

Else go to STEP 7.6. STEP 7.4: The distance between altitudes of the i-th and own UEV is calculated for the (χ_p, ψ_p, η_p) point $\Delta\eta = \eta_p^i - \eta_p^{own}$. STEP 7.5: If $\Delta\eta \leq D_{safe}$, then it is assumed that a potentially dangerous point exists, and the probability of collision P is calculated. Algorithm for electric transport collision probability evaluation task is proposed above. STEP 7.6: If the crossing point (χ_p, ψ_p, η_p) is not found, then the trajectories are parallel and D_{safe} should be checked for safe passing. STEP 8: If $P_{max} > P_{safe}$, where P_{safe} is maximal acceptable (safe) collision probability, then it is checked if the solution is better than the previous or worse. If $t = 1$, then signal β is sent to all μ neurons and repeated from STEP 6. If $1 < t < T_{max}$ and $P_{max2} > P_{max1}$, then signal β is sent to all μ neurons. The μ neurons return the previous values of W_μ and repeat from STEP 6. If $1 < t < T_{max}$ and $P_{max2} \leq P_{max1}$, then signal β is sent to all μ neurons and repeated from STEP 6. If $t \geq T_{max}$, then the situation cannot be solved in the defined time, so the safe solution is necessary. In this research such solution is speed reduction $\Delta v_i = v$ and END algorithm, else go to STEP 9. STEP 9: If $P_{max} \leq P_{safe}$, then the calculated speed Δv_{UEV}, horizontal and vertical movement directions $\Delta\tau_{UEV}$ and $\Delta\varphi_{UEV}$, changes are accepted as the solution and sent to the embedded electronic device for UEV control. Match error ε_a is compared with a maximal possible match error ε_{lim} responsible for creation of a new record in the immune memory (IM) or replacing the existing. If $\varepsilon_\alpha > \varepsilon_{lim}$, then each μ neuron saves a new set of weights W_{m+1} that was used for solving this situation and IM saves the situation X as $S_{m+1} = X$ and $m = m + 1$. Else, if $\varepsilon_\alpha \leq \varepsilon_{lim}$, then each μ neuron updates set of weights W_α and the record α in the IM is updated $s_\alpha = X$. STEP 10: END of the algorithm.

4. Prototype and computer model developed for testing of proposed algorithms

Several computer models and prototypes were developed and described in Chapter 4 of the Doctoral Thesis to prove the workability of the developed algorithms: computer model for testing the algorithm of traffic light red signal recognition for dangerous situation recognition and prevention task; computer model for testing the algorithm of CNN for object recognition for dangerous situation recognition and prevention of UEV; computer model for testing the algorithm of UEV collision probability evaluation (Fig. 5.1); computer model for testing the algorithm of ANN for collision probability evaluation and minimization for dangerous situation recognition and prevention of UEV (Fig. 5.2); computer model for testing the novel algorithm of INN for unsupervised collision probability evaluation and minimization for dangerous situation recognition and prevention of UEV (Fig. 5.3). Object-oriented programming was used for developing the computer models. A database was developed additionally for saving the results of the computer simulations.

Electric scheme with an unsupervised immune memory for UEV based on a single board computer was also developed and described in Chapter 4 of the Doctoral Thesis. The proposed electric circuit was developed for the UEV – quadcopter, but it can be applied for other types of electric vehicles as well because the developed collision prevention device is multifunctional and can be used with different types of electric vehicles.

5. Experimental testing of the proposed algorithms

Experimental testing of the developed algorithms, based on the computer models, is described in Chapter 5 of the Doctoral Thesis.

Experimental testing of the proposed algorithm of traffic light red signal recognition method for dangerous situation recognition and prevention of electric transport. Real time recognition experiment was made by using the traffic light prototype. The proposed system is trained to distinguish the red signal from the signals of other colors without any mistakes.

Experimental testing of the proposed algorithm of convolutional neural network for object recognition for dangerous situation recognition and prevention of electric transport. Several experiments were conducted to demonstrate the efficiency of the proposed algorithm. In the first experiment, CNN was trained to recognize objects such as humans, cars, and trees using a set of 5 different silhouettes for each object. CNN was able to recognize all three pictures that differed from the training set without mistakes. In the second experiment, CNN was trained to recognize traffic lights, and in the third experiment, it was trained to recognize road turns. Finally, in the fourth experiment, CNN was trained to recognize wagons. The results of experiments show that CNN is a suitable method for object recognition tasks.

Experimental testing of the proposed algorithm of collision probability evaluation of electric transport. Each object was calculating all the parameters and collision probability according to own location and parameters during these experiments.

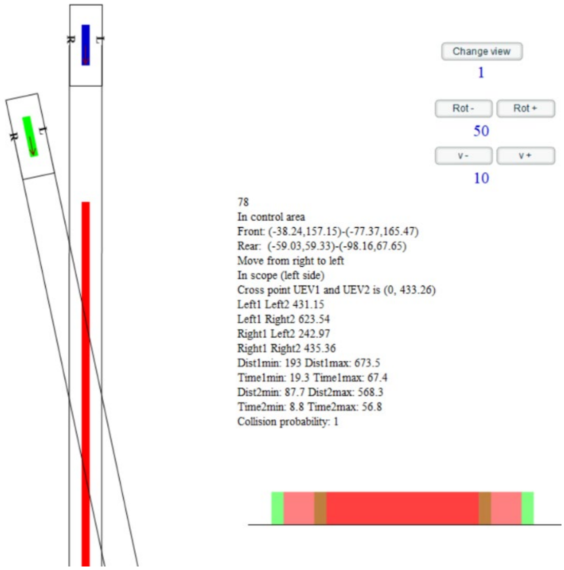


Fig. 5.1. Testing of the algorithm of collision probability evaluation of UEVs.

His own coordinates are (0; 0), and other UEV's coordinates and crossing point coordinates are calculated comparatively to own coordinates. The same speed of two UEVs was used, for the computer simulation. The developed computer model also provides the usage of different speeds of the objects. As the results of the computer experiment show, the collision probability depends on the distance till the crossing point, available time for the reaction, and the UEV speed.

Experimental testing of the proposed algorithm of neural network for collision probability evaluation and minimization for dangerous situation recognition and prevention of electric transport. Traditional neural network (ANN) experiment with training was done. ANN was trained to make the decision about the speed change to prevent the collision of UEVs. ANN input and output n amount is dynamic because the number of UEVs can be changed. The following situation was chosen for the experiment: one train; one bus; trajectories of the train and bus have a point. In this situation, ANN consists of 4 input and 2 output layers. Each element from the set is sent to the ANN input layer during the training process. When changes of speed Δv1 and Δv2 for the train and bus are found, these values are evaluated by TF.

ANN self-training experiment was done too. The experiment was also based on the danger level estimation. Coordinates and speed of two objects were entered. The proposed system calculates a possible crossing point and collision probability. If the collision probability is higher than specified, the system tries to minimize the collision probability by minimal changes of the speed.

Experimental testing of the algorithm for the collision prevention of multiple vehicles was done (Fig. 5.2). The main idea of the experiment was to set the same coordinates of the target point for three different unmanned aerial vehicles (UAV) and to make sure that the proposed algorithm is working correctly and UAVs will not collide.

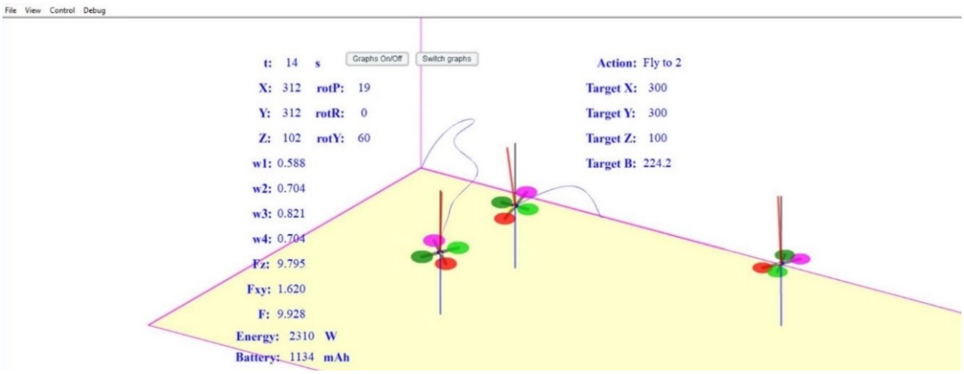


Fig. 5.2. Testing of the algorithm of NN for collision probability evaluation and minimization for dangerous situation recognition and prevention of UEVs.

As a result, three UAVs reached the target point coordinates without colliding and each of them has changed only one target coordinate – height, according to the developed algorithm. Experiments show that ANN can be useful in collision probability minimization tasks.

Experimental testing of the proposed novel algorithm of the immune neural network for unsupervised collision probability evaluation and minimization for dangerous situation recognition and prevention of electric vehicles. Three different experiments were made to prove the efficiency of the proposed algorithm. The real part of city transport system was taken as a model for the first computer experiment. The inputs for INN are motion speed of all UEVs and their distance to crossing point, including own UEV. According to this data, each UEV trains its own INN to get a speed change satisfying the target function. The decision to accelerate or brake is adjustable by specific collision sensitivity index. The first set of weights appropriate to target function is taken to memory pool at the beginning of self-training algorithm. The number of iterations to find the optimal speed change decision is limited to 200. If there is no result, speed is decreased in double. The use of 3 types of transport control were compared: 110 collisions were detected during simulation without motion control; 19 collisions were detected during simulation with ANN; no collisions were detected during simulation with a novel INN.

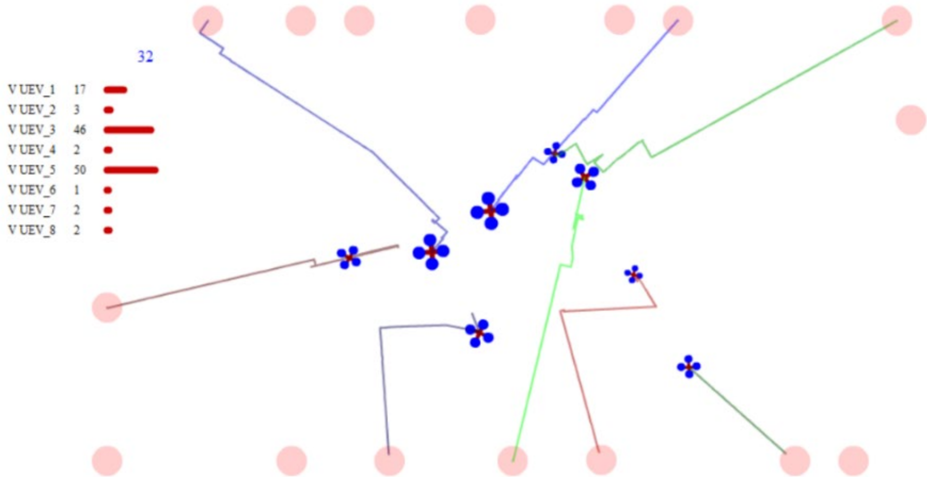


Fig. 5.3. Testing of the novel algorithm of INN for unsupervised collision probability evaluation and minimization for dangerous situation recognition and prevention of UEVs.

Second experiment was made for the group of 8 UAVs and was 10 minutes long (Fig. 5.3). Each UAV had its own size and speed, but these parameters were not changed during the simulations. UAVs were able to change the trajectory of motion (XY coordinates) or to change their speed for collision prevention. The proposed model simulates the behavior of real vehicles, that is why their decisions and output data differ. Data delays and errors were used to reproduce the conditions of the real time experiment during these simulations. In this experiment, the use of 3 types of transport control were compared. On average, 8 UAVs completed 419 trips, and 60 collisions were detected during a 10-minute simulation without motion control. The number of collisions was reduced to zero during simulations with ANN and the proposed novel INN.

While the number of trips was almost twice as high during simulations with the proposed INN compared to simulations with ANN (Fig. 5.4), the results of computer simulations demonstrate that the proposed INN is effective in minimizing the probability of collisions while also reducing the necessary computation time and increasing the number of trips.

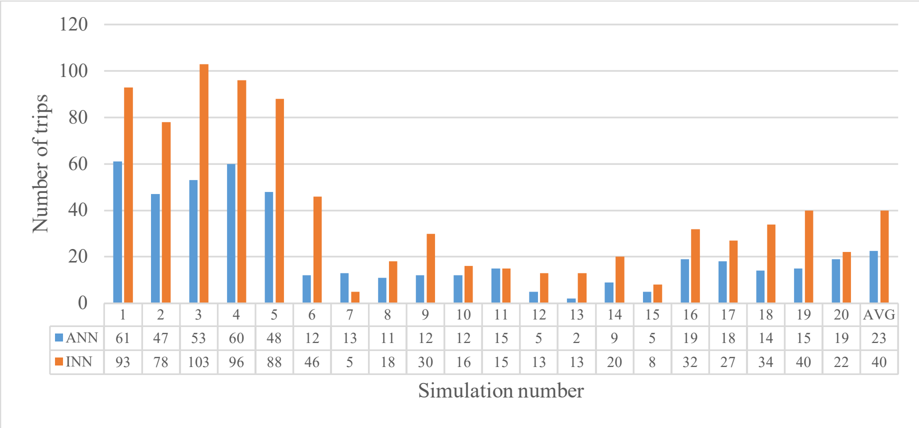


Fig. 5.4. Comparison of the results of experiments based on the average number of UEV trips during simulation.

It is necessary to choose the appropriate INN system parameters because they also influence the result. The purpose of the third experiment was to compare simulation results with different set of parameters and to understand their influence on the output data. The comparison is provided in the Doctoral Thesis.

Conclusions

Analyzing the obtained results, it can be concluded that the goal of the Doctoral Thesis has been achieved. During the development of the Thesis, the following has been done:

1. Industrial and scientific research analysis was done. It shows that despite the big number of inventions, the developed systems for unsupervised electric vehicle control do not fulfil safety tasks completely. That is why the topic of increasing the transport safety level by using the artificial intelligence systems is actual and needs to be well studied.
2. Centralized, decentralized, and distributed control system models were compared. The results of comparison show that distributed system is more preferable than centralized or decentralized. Distributed models are easier to implement, they have less components, they are cheaper for infrastructure owner, and they are not connected to the specific area, they have also decreased time for reaction and decreased risk at system failure. That is why the distributed system structure was used in this research.
3. The system structure of the proposed system was developed and described. All the functions are performed by the microcontroller or embedded computer, integrated in each electric vehicle, where the object recognition process and risk assessment are done, as well as opportunity assessment and decision making about necessary movement parameters change are done. Such solution helps to minimize data processing time because there is no need to transmit the data to the common center and backward.
4. The system structure was divided into subsystems, for better understanding of particular processes:

   a) two subsystems were based on the known methods: artificial neural network (ANN) for supervised collision probability estimation and minimization, and convolutional neural network (CNN) for the object recognition task;
   b) the third one was developed by the author within the scope of this study: immune neural network (INN) based technology of machine learning for unsupervised safe vehicle control.

   In this research, traditional neural network is included to compare its results with those of the proposed novel INN immune neural network. The objective is to draw conclusions on whether the novel network is better or worse than the traditional one.
5. The developed mathematical models were also divided according to the provided tasks:
   a) models for objects and signals recognition task;
   b) models for collision probability evaluation and location of the possible crossing point calculation task;
   c) models for collision probability minimization task.
6. Several algorithms were developed.
   a) the proposed algorithm for immune neural network for unsupervised collision probability evaluation and minimization for dangerous situation recognition and prevention of electric transport is a novel algorithm and is foreseen to be used for electric vehicle unmanned control;
   b) all the other algorithms are used as helping methods in providing the autonomous safety drive of the electric vehicles.
7. Several computer models and prototypes were developed and described in the Doctoral Thesis to prove the workability of the developed algorithms. The proposed computer models were used to test the algorithms and to solve the following tasks:
   a) traffic light red signal recognition;
   b) object recognition;
   c) collision probability evaluation and minimization;
   d) unsupervised collision probability evaluation and minimization.
8. An electrical circuit diagram of the collision prevention device with an unsupervised immune memory for unmanned electric vehicle based on a single board computer was developed and described. The proposed electrical circuit was developed for the electric vehicle – quadcopter, but it can be applied also for other types of electric vehicles because the developed collision prevention device is multifunctional and can be used with different types of electric vehicles.
9. A comparison of ANN and INN-based algorithms were done considering the impact on traffic safety and necessary time for decision calculation, where INN presents the best results, as described further. The results approve the proposed hypothesis – an immune neural network can make control decisions to prevent vehicle collisions with better performance than a traditional neural network.

The experiments done during the Doctoral research and the obtained results allow to make the following conclusions:

1. The proposed algorithm of traffic light red signal recognition method can distinguish the red signal from other signals without mistakes after the system has been trained.
2. CNN is a suitable method for object recognition process for dangerous situation recognition and prevention of electric transport. CNN must be trained in advance to minimize the necessary calculation time.
3. The collision probability is based on the distance till the crossing point, available time for the reaction, and vehicles' speed.
4. ANN method and algorithm are suitable for the collision probability evaluation and minimization for dangerous situation recognition and prevention of electric transport. It is possible to use a previously trained ANN or to use self-training.
5. The novel INN based technology of machine learning for unsupervised safe vehicle control is also suitable for the collision probability evaluation and minimization for dangerous situation recognition and prevention of electric transport. The proposed INN does not need to be trained in advance. The collision probability minimization process can be started even with an empty immune memory.
6. The proposed INN can be used for minimizing the collision probability, improving unsupervised transport safety, and faster data processing in real time conditions with minimal deviation from the task performance.
7. The proposed INN based algorithm is multifunctional and can be implemented into control systems of different types of electric vehicles. Depending on the electric vehicles specification, the system can obtain different input parameters, such as speed, location (altitude, latitude, longitude) and trajectory of motion, and produce different output data, such as speed change or movement direction change.
8. The proposed INN is better than the traditional ANN for dangerous situation recognition and prevention of electric transport because of the minimized number of detected collisions, which leads to the safer transportation process. The result of computer simulation shows that during the experiment where UEVs were able to change only the speed, but not the trajectory of motion, 19 collisions were detected during 30-minutes long simulation with ANN and no collisions were detected during simulation with the proposed INN.
9. The proposed INN is better than the traditional ANN in dangerous situation recognition and prevention of electric transport because of reduced calculation time, which leads to bigger number of safe trips. The results of computer simulations where UEVs were able to change their speed and trajectory of motion:
    a) without data transmission delays and errors, the use of INN helps to increase the number of trips by 70 % compared to the use of traditional ANN;
    b) with data transmission delays and inapproporiate maximal distance till other UEV, to start crash prevention, the use of INN helps to increase the number of trips by 92 % compared to the use of traditional ANN and to decrease the number of collisions by 25 % compared to the use of traditional ANN;

c) with data transmission delays and approporiate maximal distance till other UEV, to start crash prevention, the use of INN helps to increase the number of trips by 100 % compared to the use of traditional ANN;
d) with data transmission delays and errors, the use of INN helps to increase the number of trips by 82 % compared to the use of traditional ANN.

10. INN system parameters also influence the result. It is impossible to determine definitively which parameter values are best because the output data depends on unpredictable input parameters such as errors and delays. The INN system parameters must be adjustable depending on the situation.

Future research perspectives:

1. The theme of cybersecurity and loss of signal or communication was not considered in this research. It is considered as a prospect for future scientific research.
2. It is necessary to develop prediction algorithms for the location and velocity to continue the calculation if the data receiving is delayed.
3. The results of simulations show that the INN reduces the number of iterations and calculation time. It is necessary to analyze whether it will be sufficient for using low-powered systems.
4. It is necessary to make simulations by using multiple microcontrollers that will imitate UEVs and to compare already received results with those ones.

References

[1] S. De Nadai et al. "Enhancing safety of transport by road by on-line monitoring of driver emotions", 2016 11th System of Systems Engineering Conference (SoSE), Kongsberg, Norway, 2016, pp. 1–4, doi: 10.1109/SYSOSE.2016.7542941.

[2] Nikolajevs, A. and Mezitis, M. "Level crossing time prediction", 2016 57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), Riga, Latvia, 2016, pp. 1–3, doi: 10.1109/RTUCON.2016.7763105.

[3] Sell, R., Nikitenko, A., Ziravecka, A., Berkolds, K., Vitols, K., Czekalski, P. "Unmanned Electrical Vehicles and Autonomous System Simulation". Riga: RTU Press, 2021. 212 p. ISBN 978-9934-22-667-0. e-ISBN 978-9934-22-668-7.

[4] Repole, Donato. "Research of Parallel Computing Neuro-fuzzy Networks for Unmanned Vehicles". PhD Thesis. Riga: [RTU], 2021. 249 p.

[5] "Tesla Model S That Crashed into Fire Truck Had Autopilot Engaged", 2019, [Online]. Available: https://www.consumerreports.org/car-safety/tesla-model-s-that-crashed-into-fire-truck-had-autopilot-engaged/.

[6] "Field testing a new delivery system with Amazon Scout", 2019, [Online]. Available: https://blog.aboutamazon.com/transportation/meet-scout.

[7] "'Project Titan' Apple Car may have wide sliding doors and an adaptive stability system", 2019, [Online]. Available: https://appleinsider.com/articles/19/08/20/project-titan-apple-car-may-have-wide-sliding-doors-and-an-adaptive-stability-system.

[8] “Current topics from the world of Audi”, 2023, [Online]. Available: https://www.audi.com/en/experience-audi/mobility-and-trends/iaa-2019.html.

[9] “NVIDIA DRIVE End-to-End Platform for Software-Defined Vehicles”, 2023, [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/drive-platform/.

[10] “40+ Corporations Working on Autonomous Vehicles”, 2020, [Online]. Available: https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/.

[11] “Dual Motor Model S and Autopilot”, 2014, [Online]. Available: https://www.tesla.com/blog/dual-motor-model-s-and-autopilot.

[12] “Dual Motor Model S and Autopilot”, 2019, [Online]. Available: https://www.forbes.com/sites/greggardner/2019/09/03/ntsb-finds-teslas-autopilot-failed-in-2018-crash-with-fire-truck/#53e34a2d6b96.

[13] “Driver Errors, Advanced Driver Assistance System Design, Led to Highway Crash”, 2019, [Online]. Available: https://www.ntsb.gov/news/press-releases/Pages/NR20190904.aspx.

[14] “SharkSpotter combines AI and drone technology to spot sharks and aid swimmers on Australian beaches”, 2018, [Online]. Available: http://theconversation.com/sharkspotter-combines-ai-and-drone-technology-to-spot-sharks-and-aid-swimmers-on-australian-beaches-92667.

[15] “Amazon delivered its first customer package by drone”, 2016, [Online]. Available: https://eu.usatoday.com/story/tech/news/2016/12/14/amazon-delivered-its-first-customer-package-drone/95401366/.

[16] “Safran group”, 2023, [Online]. Available: https://www.safran-group.com.

[17] “Tesla Deaths”, 2023, [Online]. Available: https://www.tesladeaths.com/.

[18] Kornejevs, A., Gorobecs, M. “Neural Network Based UAV Optimal Control Algorithm for Energy Efficiency Maximization”. 2020 IEEE 61st International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2020): Conference Proceedings, Latvia, Riga, 5–7 November 2020. Piscataway: IEEE, 2020, Article number 9316556. ISBN 978-1-7281-9511-7. e-ISBN 978-1-7281-9510-0. Available from: doi:10.1109/RTUCON51174.2020.9316556.

[19] Dong Hwa Kim and Kyu Young Lee, “Neural networks control by immune network algorithm based auto-weight function tuning”, 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), Honolulu, HI, USA, 2002, pp. 1469–1474 vol.2, doi: 10.1109/IJCNN.2002.1007734.

[20] Li, Y., Feng, W., Zhu, X.K., Tan, K.C., Guan, X. and Ang, K.H. “PIDeasy and automated generation of optimal PID controllers”. 3rd Asia-Pacific Conference on Control and Measurement, Dunhuang, China, 31 August – 4 September 1998, pp. 29–33.

[21] K. J. Åström and T. Hägglund. “PID Controllers: Theory, Design, and Tuning, 2nd Edition”. ISA, 1995.

[22] Haykin S. “Neural networks”. A comprehensive foundation, Second edition. K. J. Prenctice Hall, 2006, 842 p.

[23] R. Pasti and L. Nunes de Castro, “A Neuro-Immune Network for Solving the Traveling Salesman Problem”, 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 2006, pp. 3760–3766, doi: 10.1109/IJCNN.2006.247394.

[24] D. Wang, C. Huo, Z. Tong, Y. Yang and Y. Wang, “Research on Vehicle Anti-collision Algorithm Based on Fuzzy Control”, 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 2361–2366, doi: 10.1109/CCDC.2019.8833461.

[25] Z. Liu, J. Chen, F. Lan, H. Xia, “Methodology of hierarchical collision avoidance for high-speed self-driving vehicle based on motion-decoupled extraction of scenarios”. Published in IET Intelligent Transport Systems (Volume 14, Issue 3, 2020) pp. 172–181.

[26] C. Rosales, J. Gimenez, F. Rossomando, C. Soria, M. Sarcinelli-Filho, and R. Carelli, “UAVs Formation Control with Dynamic Compensation Using Neuro Adaptive SMC”, 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 2019, pp. 93–99, doi: 10.1109/ICUAS.2019.8798282.

[27] L. Ling, Y. Niu, and H. Zhu, “Lyapunov method-based collision avoidance for UAVs”, The 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 2015, pp. 4716–4720, doi: 10.1109/CCDC.2015.7162758.

[28] M. Gorobetz, “Research of the genetic algorithms for optimal control of electric transport”, doctoral thesis, Riga, 2008.

[29] S. Roelofsen, A. Martinoli and D. Gillet, “3D collision avoidance algorithm for Unmanned Aerial Vehicles with limited field of view constraints”, 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 2016, pp. 2555–2560, doi: 10.1109/CDC.2016.7798647.

[30] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang, “Real-Time Traffic Flow Parameter Estimation from UAV Video Based on Ensemble Classifier and Optical Flow”, IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, pp. 54–64, Jan. 2019, doi: 10.1109/TITS.2018.2797697.

[31] Samarjit Kar, Sujit Das, Pijush Kanti Ghosh, “Applications of neuro fuzzy systems: A brief review and future outline”, Applied Soft Computing, Volume 15, 2014, pp. 243–259, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2013.10.014.

[32] S. V. Ioannou, A. T. Raouzaiou, V. Tzouvaras, R. P. Mailis, K. C. Karpouzis, S. D. Kollias, “Emotion recognition through facial expression analysis based on aneuro fuzzy network” 2005, pp. 423–435.

[33] N. M. Thanh, M. S. Chen, “Image denoising using adaptive neuro-fuzzy system”, IAENG International Journal of Applied Mathematics 36, 2007.

[34] Y. Chakrapani, K. Soundararajan, “Adaptive neuro-fuzzy inference systembased fractal image compression”, International Journal of Recent Trends inEngineering 2, 2009,pp. 47–51.

[35] V. John, K. Yoneda, Z. Liu, and S. Mita, “Saliency Map Generation by the Convolutional Neural Network for Real-Time Traffic Light Detection Using Template

Matching", in IEEE Transactions on Computational Imaging, vol. 1, no. 3, pp. 159–173, Sept. 2015, doi: 10.1109/TCI.2015.2480006.

[36] M. Z. Abedin, P. Dhar, and K. Deb, "Traffic Sign Recognition using SURF: Speeded up robust feature descriptor and artificial neural network classifier", 2016 9th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 2016, pp. 198–201, doi: 10.1109/ICECE.2016.7853890.

[37] R. Qian, Q. Liu, Y. Yue, F. Coenen, and B. Zhang, "Road surface traffic sign detection with hybrid region proposal and fast R-CNN", 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016, pp. 555–559, doi: 10.1109/FSKD.2016.7603233.

[38] Z. Yi, Y. Wang, D. Tian, G. Lu, and H. Xia, "A Road Safety Evaluation Method Based on Clustering Neural Network", 2010 International Conference on Optoelectronics and Image Processing, Haikou, China, 2010, pp. 108–111, doi: 10.1109/ICOIP.2010.60.

[39] L. Nassar and F. Karray, "Fuzzy Logic in VANET context aware Congested Road and Automatic Crash Notification", 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, Canada, 2016, pp. 1031–1037, doi: 10.1109/FUZZ-IEEE.2016.7737801.

[40] R. Qian, Y. Yue, F. Coenen, and B. Zhang, "Traffic sign recognition with convolutional neural network based on max pooling positions", 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016, pp. 578–582, doi: 10.1109/FSKD.2016.7603237.

[41] W. Zhang, L. Chen, W. Gong, Z. Li, Q. Lu, and S. Yang, "An Integrated Approach for Vehicle Detection and Type Recognition", 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl. Conf. on Autonomic and Trusted Computing and 2015 IEEE 15th Intl. Conf. on Scalable Computing and Communications and its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 2015, pp. 798–801, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.157.

[42] R. Ghosh, A. Mishra, G. Orchard, and N. V. Thakor, "Real-time object recognition and orientation estimation using an event-based camera and CNN", 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings, Lausanne, Switzerland, 2014, pp. 544–547, doi: 10.1109/BioCAS.2014.6981783.

[43] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional Neural Network Committees for Handwritten Character Classification", 2011 International Conference on Document Analysis and Recognition, Beijing, China, 2011, pp. 1135–1139, doi: 10.1109/ICDAR.2011.229.

[44] L. Chen, X. Guo, and C. Geng, "Human face recognition based on adaptive deep Convolution Neural Network", 2016 35th Chinese Control Conference (CCC), Chengdu, China, 2016, pp. 6967–6970, doi: 10.1109/ChiCC.2016.7554454.

[45] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks", 2016 IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, United Arab Emirates, 2016, pp. 1–6, doi: 10.1109/WIFS.2016.7823908.

[46] C. -D. Huang, C. -Y. Wang, and J. -C. Wang, “Human action recognition system for elderly and children care using three stream ConvNet”, 2015 International Conference on Orange Technologies (ICOT), Hong Kong, China, 2015, pp. 5–9, doi: 10.1109/ICOT.2015.7498476.

[47] H. Guan, W. Xingang, W. Wenqi, Z. Han, and W. Yuanyuan, “Real-time lane-vehicle detection and tracking system”, 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 2016, pp. 4438–4443, doi: 10.1109/CCDC.2016.7531784.

[48] S. Rawat, Z. A. Faridi, and P. Kumar, “Analysis and proposal of a novel approach to collision detection and avoidance between moving objects using artificial intelligence”, 2016 International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2016, pp. 135–138, doi: 10.1109/SYSMART.2016.7894505.

[49] J. A. Douthwaite, A. De Freitas, and L. S. Mihaylova, “An interval approach to multiple unmanned aerial vehicle collision avoidance”, 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 2017, pp. 1–8, doi: 10.1109/SDF.2017.8126384.

[50] R. Darío Fonnegra Tarazona, F. R. Lopera, and G. -D. G. Sánchez, “nti-collision system for navigation inside an UAV using fuzzy controllers and range sensors”, 2014 XIX Symposium on Image, Signal Processing and Artificial Vision, Armenia, Colombia, 2014, pp. 1–5, doi: 10.1109/STSIVA.2014.7010153.

[51] G. Stana, P. Apse-Apsitis, and V. Brazis, “Virtual energy simulation of induction traction drive test bench”, 2014 IEEE 2nd Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 2014, pp. 1–6, doi: 10.1109/AIEEE.2014.7020330.

[52] Ribickis, L., Gorobecs, M., Ļevčenkovs, A. “Neuro-Immune Algorithm for Embedded Real-Time Control System in Transport Safety Tasks”. Frontiers in Artificial Intelligence and Applications, 2018, Vol. 310, pp. 251–265. ISSN 0922-6389. e-ISSN 1879-8314. Available from: doi:10.3233/978-1-61499-929-4-251.

[53] Robert Fuller. “Neuro-Fuzzy Methods for Modeling & Fault Diagnosis”, Lisbon, Portugal, 2001, 77 p.

[54] K. Li, “The Challenges and Potential of Risk Assessment for Active Safety of Unmanned Tram”, 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 2018, pp. 22–27, doi: 10.1109/ICCAIS.2018.8570696.

[55] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang, “Real-Time Traffic Flow Parameter Estimation from UAV Video Based on Ensemble Classifier and Optical Flow”, IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, 2019, pp. 54–64, doi: 10.1109/TITS.2018.2797697.

[56] Korņejevs, A., Gorobecs, M., Alps, I., Ribickis, L. “Adaptive Traction Drive Control Algorithm for Electrical Energy Consumption Minimisation of Autonomous Unmanned Aerial Vehicle”. Electrical, Control and Communication Engineering, 2019, Vol. 15,

No. 2, pp. 62–70. ISSN 2255-9140. E-ISSN 2255-9159. Available from: doi:10.2478/ecce-2019-0009.

[57] Strupka, G. “Algorithm for Unmanned Aerial Vehicle to Supervise Applications for Civil and Power Engineering Tasks”. 17th International Symposium “Topical Problems in the Field of Electrical and Power Engineering” and “Doctoral School of Energy and Geotechnology III”: Proceedings, Estonia, Kuressaare, 15-20 January, 2018. Tallinn: Tallinn University of Technology, 2018, pp.149–152. ISBN 978-9949-832-13-2.

[58] M. Mofaddel, D. Tavangarian, “A Distributed System with a Centralized Organization”, Rostock, Germany, 1997.

[59] R. Dewan, N. Pahuja, S. Kukreja, Distributed operating system – an overview, International Journal of Research (IJR) Vol. 1, Issue 10, 2014.

[60] P. Graybeal, M. Franklin, D. Cooper, “Principles of accounting, Volume 2: Managerial Accounting”, 2019.

[61] Alom Md. Z., Taha Tarek M., Yakopcic C., Westberg S., Sidike P., Nasrin Mst. S., Hasan M., Van Essen B. C., Awwal Abdul A.S., Asari Vijayan K. 2019. “A State-of-the-Art Survey on Deep Learning Theory and Architectures” Electronics 8, no. 3: 292. https://doi.org/10.3390/electronics8030292.

[62] Hinton, G. E., Osindero, S., Teh, Y.-W. A fast learning algorithm for deep belief nets. Neural Comput. 2006, 18, 1527–1554.

[63] Nair, V., Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010, pp. 807–814.

[64] Hu, Yancai & Park, Gyei-Kark. (2020). Collision risk assessment based on the vulnerability of marine accidents using fuzzy logic. International Journal of Naval Architecture and Ocean Engineering. 12. 541–551. 10.1016/j.ijnaoe.2020.06.005.

[65] Sahin, Bekir & Senol, Yunus. (2014). A Novel Process Model for Marine Accident Analysis by using Generic Fuzzy-AHP Algorithm. Journal of Navigation. 68. 162–183. 10.1017/S0373463314000514.

[66] Sahin, Bekir & Yip, Tsz Leung. (2017). Shipping technology selection for dynamic capability based on improved Gaussian fuzzy AHP model. Ocean Engineering. 136. 233-242. 10.1016/j.oceaneng.2017.03.032.

[67] Chen, T., Grabs, E., Pētersons, E., Efrosinin, D., Ipatovs, A., Bogdanovs, N., Rjazanovs, D. Multiclass Live Streaming Video Quality Classification Based on Convolutional Neural Networks. Automatic Control and Computer Sciences, 2022, Vol. 54, No. 5, pp. 455–466. ISSN 0146-4116. e-ISSN 1558-108X. Available from: doi:10.3103/S0146411622050029.

[68] Grabs, E., Pētersons, E., Efrosinin, D., Ipatovs, A., Klūga, J., Sturm, V. Accuracy evaluation of supervised machine learning classification models for wireless network traffic. International Journal of Communication Networks and Distributed Systems, 2022, Vol. 28, No. 6, pp. 655–678.

[69] Bondarenko, A. Development of Knowledge Extraction Methodology from Trained Artificial Neural Networks. PhD Thesis. Riga: [RTU], 2020. 158 p.

[70] Namatevs, I., Sudars, K., Polaka, I. Automatic Data Labeling by Neural Networks for the Counting of Objects in Videos. In: Procedia Computer Science. Vol. 149: ICTE in Transportation and Logistics 2018 (ICTE 2018), Lithuania, Klaipeda, 1–1 January 2019. Amsterdam: Elsevier B.V., 2019, pp. 151–158. ISSN 1877-0509. Available from: doi:10.1016/j.procs.2019.01.118.
[71] Kadikis, R. Efficient Methods for Detection and Characterization of Moving Objects in Video. PhD Thesis. Riga: [RTU], 2018. 132 p.
[72] Kreicbergs, J., Smirnovs, J., Lama, A., Smirnovs, J., Zariņš, A. Road Traffic Safety Development Trends in Latvia. The Baltic Journal of Road and Bridge Engineering, 2021, Vol. 16, No. 4, pp. 58–78. ISSN 1822-427X. e-ISSN 1822-4288. Available from: doi:10.7250/bjrbe.2021-16.539.
[73] Amare, V., Smirnovs, J. Road Traffic Safety Analysis of Different Junction Types on the State Roads. IOP Conference Series: Materials Science and Engineering, 2021, Vol. 1202, Article number 012034. ISSN 1757-8981. e-ISSN 1757-899X. Available from: doi:10.1088/1757-899X/1202/1/012034/pdf.
[74] Bitins, A., Maklakovs, J., Bogdane, R., Chatys, R., Sestakovs, V. Using Adverse Event Pyramids to Assess Probabilities in Airline Safety Management. Transactions on Aerospace Research, 2021, No. 2, pp. 71–83. e-ISSN 2545-2835. Available from: doi:10.2478/tar-2021-0012.
[75] Jasiuniene, V., Cygas, D. Analysis of Older Pedestrian Accidents: A Case Study of Lithuania. The Baltic Journal of Road and Bridge Engineering, 2020, Vol. 15, No. 1, pp. 147–160. ISSN 1822-427X. e-ISSN 1822-4288. Available from: doi:10.7250/bjrbe.2020-15.465.
[76] Jasiuniene, V. Vaiskunaite, R. Road Safety Assessment Considering the Expected Fatal Accident Density. The Baltic Journal of Road and Bridge Engineering, 2020, Vol. 15, No. 2, pp. 31–48. ISSN 1822-427X. e-ISSN 1822-4288. Available from: doi:10.7250/bjrbe.2020-15.471.
[77] Dung, N. Developing Models for Managing Drones in the Transportation System in Smart Cities. Electrical, Control and Communication Engineering, 2019, Vol. 15, No. 2, pp. 71–-78. ISSN 2255-9140. e-ISSN 2255-9159. Available from: doi:10.2478/ecce-2019-0010.
[78] Buss, D., Abishev, K., Baltabekova, A. Driver's Reliability and Its Effect on Road Traffic Safety. In: Procedia Computer Science. Vol. 149: ICTE in Transportation and Logistics 2018 (ICTE 2018), Lithuania, Klaipeda, 1 January, 2019. Amsterdam: Elsevier B.V., 2019, pp. 463–466. ISSN 1877-0509. Available from: doi:10.1016/j.procs.2019.01.163.
[79] Freimane, J., Mezitis, M., Mihailovs, F. Maneuver Movements' Safety Increase Using Maneuver Locomotive Identification and Distance Control. Procedia Computer Science, 2017, Vol. 104, pp. 375–379. ISSN 1877-0509. Available from: doi:10.1016/j.procs.2017.01.148.
[80] Gorobecs, M., Ļevčenkovs, A. Intelligent Electric Vehicle Motion and Crossroad Control Using Genetic Algorithms. In: Proceedings of 10th International Conference on

Intelligent Technologies in Logistics and Mechatronics Systems (ITELMS), Lithuania, Panevezys, 21–22 May 2015, Panevezys, 2015, pp.1–6.
[81] "Elon Musk says next year's Tesla cars will be able to self-drive 90 percent of the time", 2014, [Online]. Available: https://www.theverge.com/2014/10/2/6894875/elon-musk-says-next-years-tesla-cars-will-be-able-to-self-drive-90-percent-of-the-time.
[82] "Elon Musk says Tesla's fully autonomous cars will hit the road in 3 years", 2015, [Online]. Available: https://www.businessinsider.com/elon-musk-on-teslas-autonomous-cars-2015-9.
[83] "Elon Musk's road to Twitter is paved with broken promises", 2022, [Online]. Available: https://www.washingtonpost.com/technology/2022/04/15/elon-musk-promises/.
[84] "Elon Musk is 'extremely confident' Tesla will release full autonomy in 'some jurisdictions' next year", 2020, [Online]. Available: https://electrek.co/2020/12/02/elon-musk-extremely-confident-tesla-release-full-autonomy-some-jurisdictions-2021/.
[85] "Elon Musk is once again promising Teslas will drive themselves in the near future — a claim he's been making since at least 2015", 2022, [Online]. Available: https://www.businessinsider.com/elon-musk-history-of-full-self-driving-promise-2022-1.
[86] "Elon Musk Promises Full Self-Driving Teslas in 2022", 2022, [Online]. Available: https://www.thestreet.com/lifestyle/cars/elon-musk-promises-full-self-driving-teslas-in-2022.
[87] Mike Erskine, David Milburn. Digital Train Control Functional Safety for AI Based Systems. In: Proceedings of International Railway Safety Council Conference 2019, 13–18 October 2019, Perth, Australia.
[88] Railway Gazette, SNCF targets autonomous trains in five years, September 2018.
[89] "Germany to introduce driverless trains by 2023", 2016, [Online]. Available: https://www.themanufacturer.com/articles/germany-to-introduce-driverless-trains-by-2023/.
[90] Smart Rail World, DB tells staff and unions to prepare for driverless operations by 2021, June 2016.
[91] Simmons, A., Furness, N. The main line ATO journey, IRSE News, Issue 251, January 2019.
[92] "Latvijas nacionālais terminoloģijas portāls", 2023, [Online]. Available: https://termini.gov.lv/.

Anna Beinaroviča was born in 1989 in Daugavpils. She obtained a Professional Bachelor's degree and an engineer's qualification in Railway Transport (2014) and a Professional Master's degree in Railway Transport (2016) from Riga Technical University.
From 2016 to 2020, she was the chief specialist of the Business Process Research and Development Department at LDZ CARGO Ltd. From 2017 to 2020, she was a research assistant at Riga Technical University. Currently, Anna Beinaroviča is a systems analyst at Meditec Ltd and a research assistant at Riga Technical University.
In 2017, she enrolled in the professional development programme "Programming languages C/C++" at "ITLAT Mācību centrs" Ltd. In 2022, she received a certificate of Certified Business Analysis Professional (CBAP) from the International Institute of Business Analysis (IIBA). Anna Beinaroviča is a co-author of 15 publications indexed in Web of Science and Scopus.
Her scientific interests are related to the application of artificial intelligence in transport systems, ensuring safe unmanned movement.