Infocommunications Journal

A PUBLICATION OF THE SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS (HTE)

December 2021 **Volume XIII** **Number 4** **ISSN 2061-2079**

MESSAGE FROM THE EDITOR-IN-CHIEF

From AI-based Optimisation of Ion Thrusters to Machine Learning on the Roads and in Factories .... *Pal Varga* 1

Our reviewers in 2021 2

PAPERS FROM OPEN CALL

AI-Based Electrode Optimisation for Small Satellite Ion Thrusters .... *Árpád László Makara, András Reichardt and László Csurgai-Horváth* 3

Identification of the Possible Security Issues of the 464XLAT IPv6 Transition Technology .... *Ameen Al-Azzawi and Gábor Lencse* 10

Traffic Swarm Behaviour: Machine Learning and Game Theory in Behaviour Analysis ..... *Gergely Hollósi, Csaba Lukovszki, Máté Bancsics, and Gábor Magyar* 19

A Survey on Machine Learning based Smart Maintenance and Quality Control Solutions . ... *Attila Frankó, and Pál Varga* 28

CALL FOR PAPERS / PARTICIPATION

GLOBECOM 2022 / IEEE Global Communications Conference IEEE GLOBECOM, Rio de Janeiro, Brazil 37

ADDITIONAL

Guidelines for our Authors 36

Technically Co-Sponsored by

IEEE ComSoc™
IEEE Communications Society

hte

IEEE
HUNGARY SECTION

Editorial Board

Editor-in-Chief: PÁL VARGA, Budapest University of Technology and Economics (BME), Hungary
Associate Editor-in-Chief: ROLLAND VIDA, Budapest University of Technology and Economics (BME), Hungary
Associate Editor-in-Chief: LÁSZLÓ BACSÁRDI, Budapest University of Technology and Economics (BME), Hungary

JAVIER ARACIL
Universidad Autónoma de Madrid, Spain
LUIGI ATZORI
University of Cagliari, Italy
PÉTER BARANYI
Széchenyi István University of Győr, Hungary
JÓZSEF BÍRÓ
Budapest University of Technology and Economics, Hungary
STEFANO BREGNI
Politecnico di Milano, Italy
VESNA CRNOJEVIÇ-BENGIN
University of Novi Sad, Serbia
KÁROLY FARKAS
Budapest University of Technology and Economics, Hungary
VIKTORIA FODOR
Royal Technical University, Stockholm
EROL GELENBE
Imperial College London, UK
ISTVÁN GÓDOR
Ericsson Hungary Ltd., Budapest, Hungary
CHRISTIAN GÜTL
Graz University of Technology, Austria
ANDRÁS HAJDU
University of Debrecen, Hungary
LAJOS HANZO
University of Southampton, UK
THOMAS HEISTRACHER
Salzburg University of Applied Sciences, Austria
ATTILA HILT
Nokia Networks, Budapest, Hungary
JUKKA HUHTAMÄKI
Tampere University of Technology, Finland
SÁNDOR IMRE
Budapest University of Technology and Economics, Hungary
ANDRZEJ JAJSZCZYK
AGH University of Science and Technology, Krakow, Poland
FRANTISEK JAKAB
Technical University Kosice, Slovakia
GÁBOR JÁRÓ
Nokia Networks, Budapest, Hungary
MARTIN KLIMO
University of Zilina, Slovakia
DUSAN KOCUR
Technical University Kosice, Slovakia
ANDREY KOUCHERYAVY
St. Petersburg State University of Telecommunications, Russia
LEVENTE KOVÁCS
Óbuda University, Budapest, Hungary
MAJA MATIJASEVIC
University of Zagreb, Croatia
VACLAV MATYAS
Masaryk University, Brno, Czech Republic
OSCAR MAYORA
FBK, Trento, Italy
MIKLÓS MOLNÁR
University of Montpellier, France
SZILVIA NAGY
Széchenyi István University of Győr, Hungary
PÉTER ODRY
VTS Subotica, Serbia
JAUDELICE DE OLIVEIRA
Drexel University, USA
MICHAL PIORO
Warsaw University of Technology, Poland
ROBERTO SARACCO
Trento Rise, Italy
GHEORGHE SEBESTYÉN
Technical University Cluj-Napoca, Romania
BURKHARD STILLER
University of Zürich, Switzerland
CSABA A. SZABÓ
Budapest University of Technology and Economics, Hungary
GÉZA SZABÓ
Ericsson Hungary Ltd., Budapest, Hungary
LÁSZLÓ ZSOLT SZABÓ
Sapientia University, Tirgu Mures, Romania
TAMÁS SZIRÁNYI
Institute for Computer Science and Control, Budapest, Hungary
JÁNOS SZTRIK
University of Debrecen, Hungary
DAMLA TURGUT
University of Central Florida, USA
ESZTER UDVARY
Budapest University of Technology and Economics, Hungary
SCOTT VALCOURT
Roux Institute, Northeastern University, Boston, USA
JÓZSEF VARGA
Nokia Bell Labs, Budapest, Hungary
JINSONG WU
Bell Labs Shanghai, China
KE XIONG
Beijing Jiaotong University, China
GERGELY ZÁRUBA
University of Texas at Arlington, USA

Indexing information

Infocommunications Journal is covered by Inspec, Compendex and Scopus.
Infocommunications Journal is also included in the Thomson Reuters – Web of ScienceTM Core Collection, Emerging Sources Citation Index (ESCI)

Infocommunications Journal

Technically co-sponsored by IEEE Communications Society and IEEE Hungary Section

Supporters

FERENC VÁGUJHELYI – president, Scientic Association for Infocommunications (HTE)

Editorial Office (Subscription and Advertisements)**:**
Scientic Association for Infocommunications
H-1051 Budapest, Bajcsy-Zsilinszky str. 12, Room: 502
Phone: +36 1 353 1027
E-mail: info@hte.hu • Web: www.hte.hu

Articles can be sent also to the following address:
Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
Phone: +36 1 463 4189, Fax: +36 1 463 3108
E-mail: pvarga@tmit.bme.hu

Subscription rates for foreign subscribers: 4 issues 10.000 HUF + postage

Publisher: PÉTER NAGY

HU ISSN 2061-2079 • Layout: PLAZMA DS • Printed by: FOM Media

From AI-based Optimisation of Ion Thrusters to Machine Learning on the Roads and in Factories

Pal Varga

BY THE END OF 2021 the area of Infocommunications has really got dominated by the application of Artificial Intelligence-related methods. Even in this final issue of our journal, 3/4 or the articles are putting AI or machine learning into the focus as the approach for solving the otherwise very diverse problems.

Let us have a brief overview of the articles included in the current, last 2021-issue of the Infocommunications Journal.

Intheirpaper, Árpád László Makara, András Reichardt and László Csurgai-Horváth present their first results of a system for controlling a small satellite's ion thruster's ion beam using merely the electric field. To summarize the application area, their work is aimed to optimize the electrostatic ion engine accelerator electrodes for orbit correction operations with minimum fuel usage. In order to solve the problem they used supervised machine learning techniques, along with the Least Mean Square method update steps. Since the satellite operates in space, a vacuum chamber can be a good modeling environment, but it is a very difficult to perform any measurement there, not to mention optimisation. Therefore, the authors choose to use a simulation-based approach. As they summarize their results, they found that thrusting material does not affect the ability to control. Their solution based on the optimisation of boundary conditions provides a result within an acceptable number of iteration steps. The process is fast and easy to parameterise. In addition, this procedure can be independent of the model and can be applied to a more complex physical ion thruster.

464XLAT is a prominent, standardised method to quickly deploy limited IPv4 access service to IPv6-only networks without encapsulation. The aim of Ameen Al-Azzawi and Gábor Lencse in their paper was to identify possible security issues related to 464XLAT transition. They applied the traditional STRIDE method and analysed vulnerabilities of data flows, data stores, processes and interactions in the dimensions of STRIDE. Besides, they built a test-bed and through that they analyzed how the provider-side translator behaves when it comes to DDoS attacks. The main finding is that the double translation mechanism of 464XLAT proved its effectiveness in terms of IPv4 literals communications over IPv6 infrastructure, although there are clear security vulnerabilities that have to be addressed as part of a complete system deployment.

Vehicular traffic has various peculiarities, one of them being the behaviour of the participants as individuals or as group members. Behaviour analysis is a currently hot topic in this domain. Gergely Hollósi, Csaba Lukovszki, Máté Bancsics, and Gábor Magyar are investigating traffic swarm behavior in their paper, especially by using machine learning methods and game theory in behaviour analysis. Their article outlines and compares various methods for driver behaviour analysis, even in traffic situations with complex interactions. Although state of-the-art machine learning models, latent space exploration techniques and game theory methods are successfully applied to image processing and various other problems, it is hard to find an approach that addresses traffic behaviour analysis as a whole. The authors suggest us to deeply consider the idea of traffic swarms, because this approach does not limit the interactions among drivers to the surrounding vehicles and the interactions are not bound to kinematic properties either. This allows us to analyse the actions as a result of complex behaviour instead of only trajectories or visible interactions of the vehicles.

The paper by Attila Frankó and Pál Varga surveys the machine learning based smart maintenance and quality control solutions. The authors aim to categorize the maintenance and quality control-related tasks at the various parts of the smart manufacturing ecosystem, and map machine learning solutions to these. The paper provides an overview of machine learning usage at various fields within diagnostics (especially fault detection and root cause analysis), prognostics, predictions, health management, and intelligent quality control – including non-visual and computer visual-aided types. Within smart maintenance, the hottest topic seems to be related to machine learning with unlabeled data, since in this early-mid phase of smart factory development the data is mostly unlabeled. In relation to non-visual quality control, the challenge is to enable high-data-rate and enhance flexibility in ever-changing smart factories, whereas for CV-based applications the main challenge is to improve the inference time and decrease the required computational capacity to enable image recognition- based quality inspection in production lines.

Infocommunications Journal finishes the year 2021 with these papers and wishes happy 2022 to all its readers, authors, and reviewers.

Pal Varga received his Ph.D. degree from the Budapest University of Technology and Economics, Hungary. He is currently an Associate Professor at the Budapest University of Technology and Economics and also the Director at AITIA International Inc. His main research interests include communication systems, Cyber-Physical Systems and Industrial Internet of Things, network traffic analysis, end-to-end QoS and SLA issues – for which he is keen to apply hardware acceleration and artificial intelligence, machine learning techniques as well. Besides being a member of HTE, he is a senior member of IEEE, where he is active both in the IEEE ComSoc (Communication Society) and IEEE IES (Industrial Electronics Society) communities. He is Editorial Board member of the Sensors (MDPI) and Electronics (MDPI) journals, and the Editor-in-Chief of the Infocommunications Journal.

Our reviewers in 2021

The quality of a research journal depends largely on its reviewing process and, first of all, on the professional service of its reviewers. It is my pleasure to publish the list of our reviewers of 10 countries in 2021 and would like to express my gratitude to them for their devoted work.

Your Editor-in-Chief

Levente Alekszejenkó
BME, Hungary

Ruba Almahasneh
Széchenyi István University, Hungary

Mehran Amini
BME, Hungary

Péter Babarczi
BME, Hungary

László Bacsárdi
BME, Hungary

Gábor Bergmann
BME, Hungary

Gergely Biczók
BME, Hungary

Vilmos Bilicki
University of Szeged, Hungary

Ulf Bodin
Lulea University of Technology, Sweden

Daniela Cancila
CEA, France

Levente Csikor
National University of Singapore

László Csurgai-Horváth
BME, Hungary

Tadeusz Dobrowiecki
BME, Hungary

Sara El Gaily
BME, Hungary

Andrea Farkasvölgyi
BME, Hungary

Gábor Fazekas
University of Debrecen, Hungary

Luis Lino Ferreira
ISEP, Portugal

Dániel Ficzere
BME, Hungary

Attila Frankó
AITIA International Inc., BME, Hungary

István Frigyes
BME, Hungary

Attila Gludovacz
ELTE, Hungary

László Gyöngyösi
BME, Hungary

Dániel Hadházi
BME, Hungary

Csaba Hegedűs
TATA Consulting Services, Hungary

Attila Hilt
Nokia Networks, Hungary

Gábor Horváth
BME, Hungary

Péter Horváth
BME, Hungary

Árpád Huszák
BME, Hungary

Miren Illarramendi
Mondragon University, Spain

Sándor Imre
BME, Hungary

Csaba Johanyák
John von Neumann University, Hungary

Adi Karahasanovic
Combitech, Sweden

Andrey Koucheryavy
St. Petersburg State University of Telecommunications, Russia

Kálmán Kovács
BME, Hungary

László Kovács
AITIA International Inc., Hungary

Szilveszter Kovács
University of Miskolc, Hungary

János Ladvánszky
Ericsson Research, Hungary

Milán Magdics
BME, Hungary

Markosz Maliosz
BME, Hungary

Gábor Magyar
BME, Hungary

Attila Mihály
Ericsson Research, Hungary

Attila Mátyás Nagy
BME, Hungary

Lajos Nagy
BME, Hungary

Gábor Németh
BME, Hungary

Péter Orosz
BME, Hungary

László Osváth
BME, Hungary

Dávid Papp
BME, Hungary

Dorottya Papp
BME, Hungary

Zsolt Papp
BME, Hungary

Alija Pasic
BME, Hungary

Balázs Pejó
BME, Hungary

Sándor Plósz
Heriot-Watt University, UK

Zoltán Pödör
ELTE, Hungary

Rama Rao T.
SRM Institute of S.T., India

Csaba Rotter
Nokia Bell-Labs, Hungary

Al-Radhi Mohammed Salah
BME, Hungary

Ágoston Schranz
BME, Hungary

Gábor Soós
BME, Hungary

Burkhard Stiller
University of Zürich, Switzerland

György Strausz
BME, Hungary

Áron Szabó
Sigma Technology, Hungary

Luca Szegletes
BME, Hungary

Szabolcs Szilágyi
University of Debrecen, Hungary

Dávid Sztahó
BME, Hungary

Judit Tamás
Eszterházy University, Hungary

Balázs Tarján
SpeechTex Inc., BME, Hungary

Miklós Telek
BME, Hungary

Tamas Tettamanti
BME, Hungary

László Toka
BME, Hungary

Tamás Tóthfalusi
BME, Hungary

Ákos Tündik
Nokia Networks, Hungary

Boldizsár Tüű Szabó
Széchenyi István University, Hungary

Eszter Udvary
BME, Hungary

Pal Varga
BME, Hungary

Rolland Vida
BME, Hungary

AI-Based Electrode Optimisation for Small Satellite Ion Thrusters

Árpád László Makara[1], András Reichardt[2] and László Csurgai-Horváth[3]

Abstract—**Computing capacities for numerical modelling are available to an unprecedented extent today. The spread of various artificial intelligence (AI) -based solutions (which in many cases are also resource-intensive operations) is also facilitated by this increase in capacity, which offers several new opportunities in this area. On the one hand, optimization tasks can be done quickly, on the other hand, it is also possible to solve (estimate) problems where we cannot (for some reason) create a model for the initial problem. In our article, we investigate how to apply artificial intelligence-based solutions to electromagnetic field computing tasks as efficiently as possible. The required theoretical summary presents an implemented application: optimization of electrostatic ion engine accelerator electrodes for orbit correction operations. To solve each problem, we used methods from the supervised machine learning toolkit, usually along with LMS (least mean square method) update steps. All inputs required for AI were solved by numerical space calculation (primarily using the finite element method). The data input required to optimize the electrodes of an ion thruster can come from two sources: measurement data or simulation results. Given that the operating environment of a satellite can be modelled in a vacuum chamber, it is a particularly difficult issue to perform the measurement, but even more difficult in the case of optimisation. Therefore, an effective solution to the problem can only be achieved by simulation. The primary goal of this research is to optimise the fuel (in this case, the number of ions) during operation, with the stated aim of maximising the time of operation of the spacecraft.**

Index Terms—**ion thruster, electrode, optimisation, simulation, artificial intelligence, small satellite**

I. INTRODUCTION

The launch system of a spacecraft is a complex structure and it consists of typically several rocket stages that often use different types of high-power propulsion systems. A typical and commonly applied structure can be observed in the Ariane rocket family [1]. Usually high-power booster(s) with solid propellant can be found in the first stage, extended with a cryogenic core stage with liquid propellant. The boosters are operating in the first launch phase for a couple of minutes and after the separation are returning to the ground for later reuse. The main stage operates up to its separation when the spacecraft's performance value an appropriate height and speed reached. At this point the upper stage is ignited to place the payload(s), e.g. satellite(s) to their final orbit. The above-mentioned rocket engines are providing very high thrust in the kN-MN range for several minutes of duration. When the satellite already reached its planned height and speed the propulsion system supports the spacecraft's further orbital maneuvers and changes its position by firing thrusters. Depending on the mission's type, the task of the propulsion subsystem may perform apogee injection e.g. to reach a final geostationary orbit. For that one a few hundred of N thrust level is required. In order to perform minor orbit control, like modifying the inclination, maintenance of the orbit, low power thrusters with few times 10N is required. The orientation of a satellite should be also controlled in order to maintenance the spin rate, perform axes stabilization or rotate the satellite to a specific direction. This kind of maneuver requires a few N of thrust.

Satellite propulsion subsystems have many different operating principles and may use different propellant types. Chemical propulsion systems with monopropellants or bipropellants may provide higher thrusts. However, the resulted chemical products may influence the external environment of the spacecraft and it could be intolerable by the mission's goal, especially when there are sensitive measurement devices among the payloads. Cold gas systems with neutral gases are operating in the lower power ranges. The primary choice is nitrogen as its relatively large molecular size prevents the fuel leakage. An alternative propellant is argon, when nitrogen cannot be applied for specific reasons.

The electric propulsion systems are using ionizable gases as the propellant. Electrical power supplied by an external energy source is used to accelerate the propellant to extreme velocities and thereby achieve very high specific impulses. However, the power as well as the thrust is limited by the available electrical energy delivered by batteries, solar generators or radioisotope thermoelectric generators (RTGs). The electric propulsion systems have very low thrust, comparing to the previous methods. Thrusters for electric propulsion require propellants which can be easily evaporated and ionized and which have a high molecular weight. Therefore the development of thrusters for electric propulsion concentrated on the use of inert gas xenon, which can be stored in high-pressure gas tanks. Xenon has a high molecular weight and can be quite easily ionised. The idea of electric propulsion is not new - NASA Glenn Research Center has been a leader in ion propulsion technology development since the late 1950s, with its first test in space - the Space Electric Rocket Test 1— flying

[1,2,3] Budapest University of Technology and Economics Department of Broadband Infocommunication and Electromagnetic Theory, Budapest, Hungary

[1] corresponding author (e-mail: makara.arpadlaszlo@edu.bme.hu)

[2] (e-mail: reichardt.andras@vik.bme.hu)

[3] (csurgai-horvath.laszlo@vik.bme.hu)

DOI: 10.36244/ICJ.2021.4.1

on July 20, 1964 [2]. In [3] and [4] the principles of operation and the several types of thrusters that are either operational or in advanced development are discussed. Ion thrusters (based on a NASA design) are now being used to keep over 100 geosynchronous Earth orbit communication satellites in their desired locations and there are other missions with electric propulsion system as well.

To the beginning of 2021, the use of ion thrusters has once again come to the fore [5]. An open question for the future, which is the minor version for an ion thruster that can work (especially for small satellites in space orbits close to Earth). To minimise it, on the one hand, it is necessary to choose the right fuel. On the other hand, adapting the structure of the spacecraft to this. Previous simulation results available in the literature are primarily particle-based plasma simulations, all of which are faithfully modeled by a given arrangement (such as the T5 model, flow simulation of an electrostatic ion thruster [6]).

One possible, well-functioning method for electrostatic ion engines is Monte Carlo Collision Method-based simulations [7]. Another proven method is the PIC (Particle In Cell) to solve the problem, for which we can see results for DP1 [8]. Also worth mentioning is the Hall Thrusters, where the magnetic field is also present in the direction of the plasma. Such simulation expiration dates are fundamentally more complex than the electrostatic cases that are the subject of recent research [9].

The most crucial goal of the current research is to investigate the effect that can be achieved by controlling the potential of each individual electrode when using electrostatic ion thrusters. Based on the available results, two assumptions should be take into account in order to get correct results: first of all, ions must not condense on the electrodes or on any component of the spacecraft, including other ions. On the other hand, the density and initial direction of the ions in the space of the accelerator electrodes cannot be arbitrary. With this approach, it is primarily possible to study how the movement of ions can be influenced by changing the potentials, and based on this, the amount of fuel consumption can be optimized. The results of the current research were designed for a general structure, while using typical values from previous spacecraft.

The design of the acceleration electrodes of ion thrusters is a complex problem. In ground conditions only vacuum-chamber measurements are appropriate to simulate the real conditions of space, that is a particularly complex issue. Therefore simulations are feasible solutions in order to develop and optimize the electrodes. The problem area is an electromagnetic field computing task where applying an AI-based method is a novel and encouraging approach to develop the optimal accelerator electrodes of ion thrusters. In this paper we provide a finite element based solution to calculate quasi-electrostatic field of the internal the thruster and we perform simulations on the outlets. The applicability of this thruster method for small satellites is also investigated.

The organization of this paper is as follows. Section II. provides a brief introduction to electric engines. Section III. describes the presentation of our ion engine model made in the earlier phase of the research [10]. In Section IV., we present the ion engine optimisation implemented based on our simulation model. Last but not least, in Section V. we summarise our results.

II. PRINCIPLES OF ION ENGINE OPERATION

A. Electric propulsion engine types

Electric propulsion (EP) can be categorized by different ways [11]. First type of ion thrusters are utilizing only electric field to accelerate ions (see Fig. 1.). The other type is using magnetic field and electric field to accelerate and control ion or plasma jets [4]. This type is more sophisticated but due to its size, it is not possible to implement within CubeSat dimensions (the standard dimension 1 unit CubeSat is 10 cm × 10 cm × 10 cm size) [12]. Both types are using electric field acceleration grid (AG). It is formed as two grids separated from each at a few centimeters having a potential difference between them. Ions are entering through the input grid and leaving through the output grid at a higher speed.

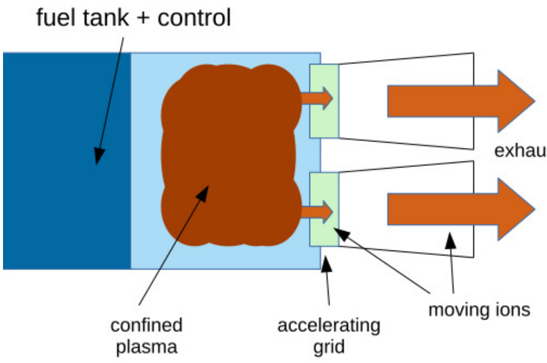


Fig. 1. Principle of an ion thrust engine operation.

B. Electric propulsion system proposed

We propose a simple electric propulsion system for small satellites based on the principle by accelerating an ionized gas moving outward of the spacecraft and thrusting the spacecraft to the opposite direction. The gas should be charged (ionised) allowing the acceleration by electric field. Our model is based on a simple discharge chamber where gas is ionized and an acceleration grid that moves ions outward. Control of ion jets are performed using electrodes attached to the inner surface of the nozzle and driven by a potential. In this paper we analyze the ion jet control inside the nozzle by applying only an electric field.

C. Possible system for CubeSats

CubeSats have limited space and mass. Currently there are spinoffs offering a 2 unit large propulsion system based on xenon or iodine [13]. With this system, an additional steering mechanism can be used to change the orientation of the satellite based on deflecting ions flying out from the thruster. The deflector system can bend the beams, and thereby it can help to maneuver the satellite. The system is shown on Fig. 2.

The rectangular formations are the electrodes of the deflector system. It can turn the ion beam and thereby it can turn the satellite too.

The arrangement shown in Fig. 1. is used in the present research. We did not examine how the plasma will be produced, handled, stored and how to control the flow of sufficient ions into the accelerator space. Furthermore, the modeling of the hollow electrode is also neglected.

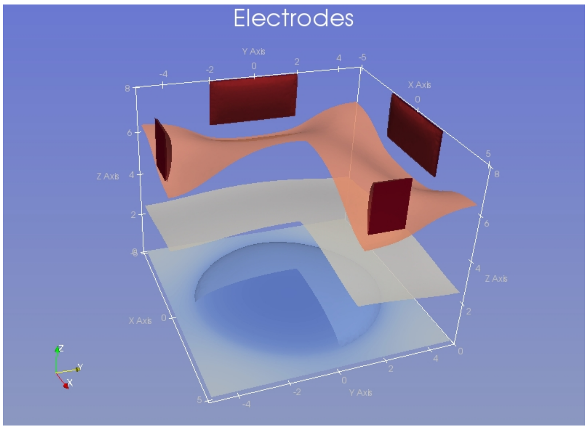


Fig. 2. Outline of the nozzle. Small rectangles on the sides are control electrodes. The tube on the bottom is the upper side (top side) of the acceleration grid. The top of the grid is connected to zero voltage. The potential of the electrodes is used to control ions movements. Sizes are measured in centimeters. Only for the first row of electrodes shown.

III. SIMULATION OF AN ELECTROSTATIC THRUSTER

The electromagnetic field created by control electrodes can be modeled as a constant field with very slow change in time [14]. We disregard the physical extent of the ions, only their mass counts for acceleration. The ions have a sufficiently low density in the space between the accelerator electrodes to correct the assumption that they do not collide with each other and that the field they create is negligible. Then the electrostatic Laplace-Poisson equation is as follows (the partial differential equation, PDE):

$$\nabla \varepsilon_0 \nabla \varphi = 0. \tag{1}$$

where ε_0 is the vacuum permittivity, φ is the potential. In the simulation region we solve equation (1) with boundary conditions (see Section III-A.) and get the field that moves the ions. The problem was solved by the finite element method (FEM). A possible arrangement of the nozzle is shown in Fig. 3. The force acting on the embankments can be expressed by Lorentz's law [15]:

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{2}$$

where $\mathbf{F}$ is the force, q is the charge of the moving particle, $\mathbf{E}$ is the electric field, $\mathbf{v}$ is the velocity of the moving particle and $\mathbf{B}$ is the magnetic induction. The field $\mathbf{B}$ generated by the hum of the ions is negligible compared to the effects of the electrodes, and the change in the potentials of the electrodes is slow enough hence their effects are not significant.

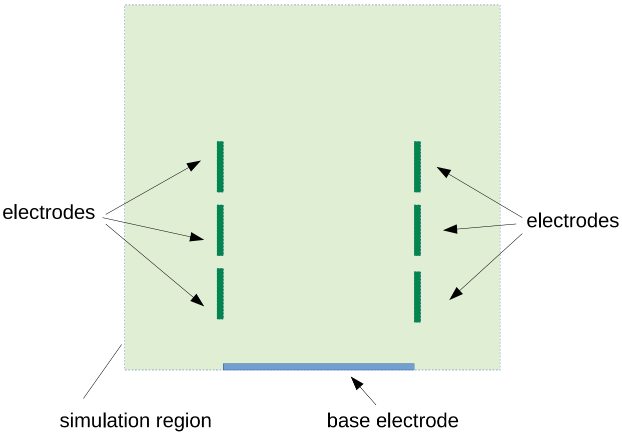


Fig. 3. Geometry of simulated rectangular shaped nozzle

After solving this model we can calculate the trajectory of ions starting from base electrode by solving equation of motion (3) (which is the Newton's second law) with force calculated from potential (4), [16]:

$$m_{\text{ion}} \cdot \frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2} = \mathbf{F} \tag{3}$$

$$\mathbf{F} = (-1) \cdot q_{\text{ion}} \cdot \nabla \varphi \tag{4}$$

where $\mathbf{r}$ is the local vector of the moving ion, t is the time.

The current research is limited to two-dimensional cases for faster runtime. A simulation result (the red line is the route of a single moving ion) is shown in Fig. 4., which was generated using MATLAB PDETool [17]. The three-dimensional case is a magnitude difference in computing capacities. While in the two-dimensional case the steps are of the order of $\mathcal{O}(N^2)$, in this case $\mathcal{O}(N^3)$, where N is the number of FEM simulations elements (degrees of freedom). The more accurate the simulation result we intend to achieve, the more points we need (denser mesh). For our method, this is the most resource-intensive part where we chose the smaller one.

The effect of the actual ion beam is negligible from the point of view of simulation so that the constituent ions can be simulated one by one. Then it is enough to calculate the resulting $\mathbf{E}$ field and then the motion of each ion. The force of the thruster radius can be calculated as a result of these individual forces. This approximation is valid as long as the resulting electric field strength is given by the potential of the electrodes.

A. The boundary conditions

There are two ways to consider different physical objects during FEM simulation. One possible solution is to represent it in material parameters (application of non-zero charge density or relative dielectric constant in sub-areas). The other is to apply boundary conditions (BC) [18].

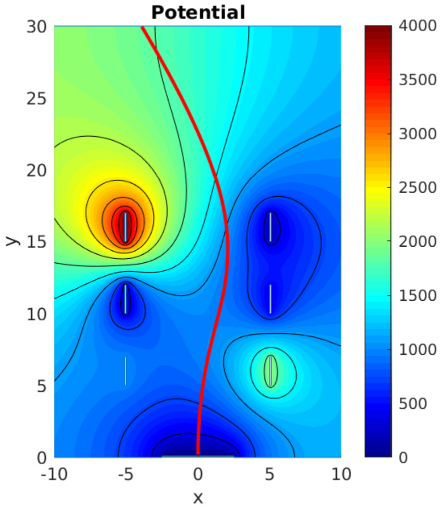


Fig. 4. Electrostatic potential simulated (colored contour) and particle trajectory (red line). Sizes are measured in centimeters.

Each electrode in the simulation means a Dirichlet-type boundary condition. There are two practical reasons for this: from an operational point of view, only the effect of potential is interesting, and on the other hand, its value can be easily optimised. Denote the potential of the i^{th} electrode by

$$\varphi_i = U_i \tag{5}$$

It is necessary to select a reference point with a value of zero, for which the base electrode is best suited (see Fig. 3.). This is not a real electrode but a connecting hole between the plasma space and the acceleration space. In the two-dimensional model, this is also a similar boundary condition to the real electrodes, with only a constant zero voltage.

It is necessary to handle the space environment at a sufficient distance from the substantial part of the ion thruster (see Fig. 3.). In a typical operating environment of the small satellites, charged particles are present in negligible amounts. Therefore, the atmosphere acts as a kind of ideal insulator that can be modelled with a homogeneous Neumann-type boundary condition:

$$\frac{\partial \varphi}{\partial n} = 0 \tag{6}$$

That is enough for the simulation. To describe more accurately a real-world system, it would still be necessary to model the environment of the satellite and the electrodes (such as frame, electronics, etc.). With both boundary conditions and material properties, these can be perfectly modelled; however, they are not required for current research results, so we omit them for the sake of simplicity.

IV. OPTIMISATION

Optimisation of parameters means, on the one hand, the physical parameters of the spacecraft and, on the other hand, the optimisation of operational resources. For each satellite, the following parameters can be modified to optimise the expected performance:

- the geometry of the electrodes and the thruster,
- the type of ion used as fuel,
- number of accelerator electrodes,
- size of the accelerator electrodes
- electrode potential.

In the case of small satellite design, the standards and mission goal essentially limit the possibilities of the spacecraft size and weight, so there is no possibility for large-scale optimisation here. Therefore, the primary goal is to optimise the use of energy (typically electricity) and fuel (in this case, the number of ions) during operation, with the stated aim of maximising the time of operation of the spacecraft. In the case of electrostatic ion engines, the desired thrust can be achieved along with two strategies: accelerating more ions with lower voltages or accelerating less fuel particles with much higher energies than before. A longer operating time can be ensured for the thruster unit if the ions are available for as long as possible. This cannot be easily replaced, so the task is to minimalise the fuel (ion) consumption.

Thus, the subject of the current research is the optimisation of the individual operational parameters.

A. *Parameter optimization*

As we know the expected movement of our spacecraft, so the application of supervised machine learning is necessary. Given that the partial differential equation of the electrostatic model presented in the simulation part describes the desired behaviour with sufficient potential, the regression problem itself is given.

The surface of each electrode can be considered equipotential. The values of this form is the weight vector $\mathbf{W}$ (where $\mathbf{W} = [U_1, U_2 .... U_N]^T$, i is the i^{th} electrode potential). These values are, by the way, the boundary conditions of the FEM model presented in Section III. The error function is interpreted as follows:

$$K = e^2(k) = [t(k) - a(k)]^2 \tag{7}$$

where e is the value of the error, t is the expected output, a is the instantaneous value of the estimate, and k is the number of iterations. The error is actually the difference between the expected trajectory of each ion and the simulation result. This difference can be defined according to several approaches, which will be discussed in more detail in Section IV-B. The task is to minimize the resulting K error function, thus optimizing the orientation corrections. This optimisation is implemented with supervised machine learning within that the Least Mean Square method [19]:

$$\mathbf{W_{next}} = \mathbf{W_{prev}} + 2 \cdot \mu \cdot e \cdot \mathbf{p} \tag{8}$$

where μ is the learning-rate, e is the error, $\mathbf{p}$ is scaling factor or input parameters. Using an LMS iterative procedure, the residuum part will always remain, which can be somewhat reduced by averaging the repeated calculations. The optimisation flowchart is shown in Fig. 5.

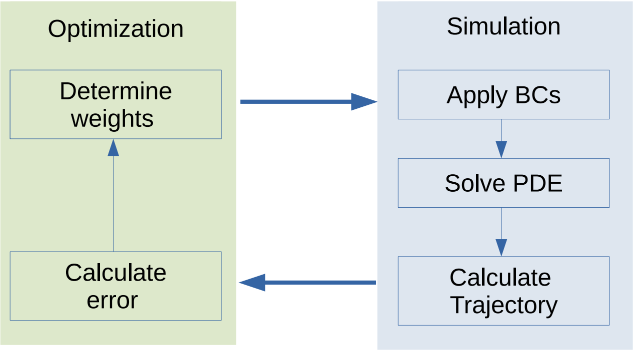


Fig. 5. Schematic flow chart of the optimization and simulation process as an infinite loop.

B. Choosing the target

One of the most important pillars of the supervised machine learning is the designation of target states (t). If this is flawed (or distorted), the training process will not move towards the ideal solution. In the present case, taking advantage of the momentum retention, we expect that the ion beam is pushing the spacecraft along this path. In doing so, the trajectory of each ion must be determined. This can be done either by prescribing control points or control regions. Fig. 6. shows a 4-zone control arrangement.

The error is actually the distance between the control point or zone and the ion trajectories at a given altitude:

$$e_i = t_i - a_i = \text{distance}(a_x, c_x)|_{y=\text{const}} \tag{9}$$

where e_i is the i^{th} control zone error, a_x is the ion place, in a given altitude, and c_x is the control zone at same altitude. In the case of parameters with acceleration and velocity dimensions, it is expedient to prescribe the control zone in the width of the entire calculation range or to interpret the control moment on a time basis. Distance means Euclidean distance for each quantity.

When prescribing a route, it is worth defining these points at several heights, and by applying a weighting form the resulting error is:

$$e = \frac{1}{N}\sum_{i=1}^{N} \omega_i (t_i - a_i) \tag{10}$$

where w is the given control point's scaling/importance factor, N is the number of the control points/zones.

The role of each control point is different. At the beginning of the arrangement (close to the base electrode), their primary function is to prevent ions from collision with the electrodes. In contrast, the points at the end of the engine are almost exclusively responsible for orienting the ion in the correct direction. It is advisable to use as many control elements that are still proportional to the degrees of freedom (this is actually a nonlinear curve fitting task). The width of the zones should be selected in proportion to the width of the starting point and the initial angle of the ions. If we select too many reference points, two things can happen. First, the number of iterations required can be greatly increased. On the other hand, it is conceivable that the error function will have several local minimal relatively close to each other, and the algorithm will jump between each other. The latter is especially typical for small-angle turns, where straight travel also results in a small error. In Fig. 7. there is a relatively small degree case with well-chosen control points.

C. More complex error criteria - status based error

Not only the path of the ion but also its other parameters may be useful. Typically, this can be the value of acceleration (per coordinate) and its position (also per coordinate). There is a set of adjustable parameters (i.e., voltage of the accelerator electrodes) on which all the parameters we are looking for depending. For each element of the state, we can write an error function as before. There is no certainty that the zero position of each function will be in the same place. For this reason, we need to prioritize which parameter is how important to us. Of course, we can also define criteria that are interpreted not at a specific place but at a given time (after a step). Then the name of the reference value is correct, but the procedure is the same.

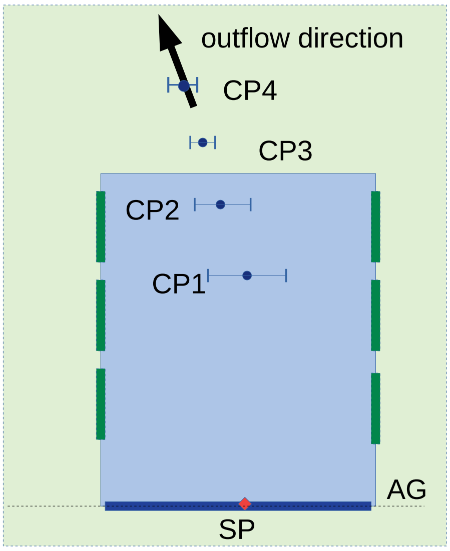


Fig. 6. Control points of trajectory definition. CPs are control points, where ions should fly through. AG is the top of acceleration-grid, SP designates starting point.

As it was shown in the previous subsection the cost function with highest priority is optimised. The potentials are then adjusted according to the next parameter in the sequence, taking care that the higher priority parameter cannot deteriorate by more than a predefined value. As soon as the error has been reduced to a sufficient level, the next parameter is consumed, taking care that two values cannot deteriorate more than the individual limit.

In fact, we do the same thing as in the case of tension. We narrow down the range of possible solutions, thus most likely ignoring the best solution. Calculating the parameters at a lower location in the priority queue is actually a ”roaming” around the optimum of another parameter, looking for a good location suitable for both values. The steps in the procedure are the following:

1) Optimisation of the parameter with highest priority (typically by averaging multiple runs),
2) Specify fault tolerance,
3) Optimisation of the second parameter in the priority queue (similar to the first point) so that the error for the higher priority parameters cannot leave the tolerance band,
4) Repeating steps 1.-3. so that the tolerance values for the higher priorities cannot be violated by each optimisation.

D. Choosing the input parameters or the scaling factor

The parameter $\mathbf{p}$ can be denoted with two names: an input parameter or a scaling factor. The former is a common name for typical neural networks, which can be used in the present case. The latter presupposes a higher level of abstraction, where each parameter loses its physical meaning and encodes a prior knowledge [20]. Regardless of interpretation, the values of $\mathbf{p}$ affect the learning rate (μ):

$$\mathbf{P} = \mathbf{p} \cdot \mathbf{p}^T. \tag{11}$$

The maximum eigenvalue (λ_{max}) or trace of the $\mathbf{P}$ matrix is critical. The reciprocal of this value is an upper limit on the maximum value of the learning rate to keep the process stable [20]:

$$\mu_{max} < \frac{1}{\lambda_{max}} \leq \frac{1}{trace\mathbf{P}}. \tag{12}$$

In the case of assigning a physical input parameter to the same voltages, the natural choice is the surface (or, in two-dimensional cases, the perimeter) of each electrode. This is actually the same as the boundary condition area within our FEM model calculation range. This actually means that each potential is weighted by the size of the electrodes. The process can be speeded up by consistently multiplying the values one of the sides by minus one. Thus, the opposing electrodes receive a more drastic change during each update step. Overall, the system convering faster towards the solution; however, the voltage values of each electrode pair may oscillate at the beginning of the process.

In the present arrangement, the process can be accelerated (slowed down if incorrectly selected) if the vector $\mathbf{p}$ is a scaling factor (or more complex, a scaling function). In this case, the prior knowledge should be used that towards the end of the arrangement, the current electrodes contribute more to the translation than the previous ones. Thus, in the present study, the values of these scale factors were typically chosen to be larger than the others (usually by an order of magnitude). During testing, in most (but not always) cases, we found that the residual error value (K) was closer to zero than if we had chosen them equally. No drastic acceleration was generally observed in the number of iteration steps (except for a few more specific cases). However, due to the more precise fit, the expected result can be found sooner, so overall it is faster to find the potentials for each form of movement. Presumably, one possible improvement of the procedure would be to create a deterministic, more efficient scaling rule.

Variation of the input parameter/scaling factor can be understood as a time-varying iteration rate. In the event that we do so and start from a large value in time (so that the process is still stable), we take advantage of the rapid initial changes in the LMS. After a few iterations (typically 20-30), it is advisable to reduce its value cyclically. This reduces the rate of change but also the value of the residuum part. In the tests performed in the current research, we found that with well-chosen scaling factors, the procedure usually takes the order of 100 steps. Thus, in essence, the calculation of a given form of movement (the most resource-intensive element of which is the FEM simulation) can be solved even on a personal computer.

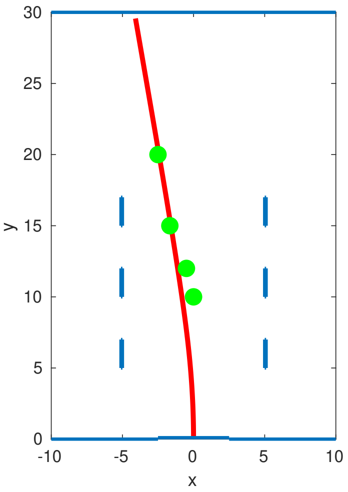


Fig. 7. Optimization result with control points. Sizes are measured in centimeters.

E. Physical limitations

There are several physical constraints that need to be respected to optimise the space of the model. For a non-relativistic initial model, the first and most important is to figure out the velocity of ions. In the case where the velocity of each ion becomes comparable to the speed of light (e.g., it reaches 10% of the speed of light), the result becomes increasingly inaccurate. The procedure would give a false result also if the speed of light was not the maximum of the available speed. However, it is necessary for the algorithm to indicate when the ions are achieved at this speed.

Another critical physical limitation is the finite voltage of the electrodes. In the case of a spacecraft, of course, the voltage level can only be produced in a specific range. This can typically be on the order of a few kV [8]. This can be different for every electrodes. The output of the optimisation step must always satisfy the following inequality:

$$\mathbf{U}_{min} \leq \mathbf{W} \leq \mathbf{U}_{max} \tag{13}$$

where U_{min} is the lower and U_{max} is the upper limit for the possible voltage value in the spacecraft. It can be interpreted as both continuous and discrete quantities.

By not allowing the electrodes to take on any value, some of the minimum possible locations of the function K are omitted. A trivial example of this is when we prescribe as much acceleration as possible (without an upper bound). An obvious solution is to shape the most significant accelerating potentials, but the physical limitations of the spacecraft do not allow it. Thus, the task is to find the best solution in a local environment of the error function.

V. CONCLUSION

We presented the first results of a system that can control ion thruster's ion beam using only electric field. The effect of guiding electrodes can be controlled by the magnitude of electrode potential and connection status of the other electrodes. It is found that thrusting material does not affect the ability to control. A solution based on the optimisation of boundary conditions provides a result within an acceptable number of iteration steps. The process is fast and easy to parameterise. In addition, the procedure can be independent of the model and can be applied to a more complex physical thruster. The topic of future research is how to interpret the error in the case of three-dimensional procedures. In this paper, we examined only one starting point and start velocity; it is straightforward to use more points and velocities to optimise. Of course, such a simple optimisation can not be used to completely resolve that problem. In our future work, we intend to examine the usability of the more complex error function and three-dimensional arrangements.

REFERENCES

[1] Arianespace, *Ariane 5 User's Manual Issue 5 Revision 2*. Ariane, Oct. 2016.

[2] Ronald J. Cybulski, Daniel M, Shellhammer, Robert R. LoveII, Edward J. Domino, and Joseph T. Kotnik, *RESULTS FROM SERT I ION ROCKET FLIGHT TEST*, 1965th ed.

[3] J. E. Pollard, D. E. Jackson, D. C. Marvin, A. B. Jenkin, and S. W. Janson, "Electric propulsion flight experience and technology readiness," Jun. 1993. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1993-2221

[4] K. Holste, P. Dietz, S. Scharmann, K. Keil, T. Henning, D. Zschätzsch, M. Reitemeyer, B. Nauschütt, F. Kiefer, F. Kunze, J. Zorn, C. Heiliger, N. Joshi, U. Probst, R. Thüringer, C. Volkmar, D. Packan, S. Peterschmitt, K. T. Brinkmann, H.-G. Zaunick, M. H. Thoma, M. Kretschmer, H. J. Leiter, S. Schippers, K. Hannemann, and P. J. Klar, "Ion thrusters for electric propulsion: Scientific issues developing a niche technology into a game changer," *Review of Scientific Instruments*, vol. 91, no. 6, p. 061101, Jun. 2020. [Online]. Available: **DOI**: 10.1063/5.0010134

[5] I. Levchenko, S. Xu, G. Teel, D. Mariotti, M. L. R. Walker, and M. Keidar, "Recent progress and perspectives of space electric propulsion systems based on smart nanomaterials," *Nature Communications*, vol. 9, no. 1, p. 879, Dec. 2018. [Online]. Available: http://www.nature.com/articles/s41467-017-02269-7

[6] M.JugrootandJ.K.Harvey,"Simulation of flows within an electrostatic ion thruster for space missions," *The Aeronautical Journal* (1968), vol. 105, no. 1053, p. 613–618, 2001.

[7] B.Korkut,Z.Li,andD.A.Levin,"3-d simulation of ion thruster plumes using octree adaptive mesh refinement," *IEEE Transactions on Plasma Science*, vol. 43, no. 5, pp. 1706–1721, 2015.

[8] J. Wang, D. Brinza, and M. Young, "Three-Dimensional Particle Simulations of Ion Propulsion Plasma Environment for Deep Space 1," *Journal of Spacecraft and Rockets*, vol. 38, no. 3, pp. 433–440, May 2001. [Online]. Available: https://arc.aiaa.org/doi/10.2514/2.3702

[9] K. Shan, Y. Chu, Q. Li, L. Zheng, and Y. Cao, "Numerical Simulation of Interaction between Hall Thruster CEX Ions and SMART-1 Spacecraft," Jun. 2015. [Online]. Available: https://www.researchgate.net/publication/282494427_Numerical_Simulation_of_Interaction_between_Hall_Thruster_CEX_Ions_and_SMART-1_Spacecraft **DOI**: 10.1155/2015/418493

[10] A. Makara, A. Reichardt, and L. Csurgai-Horváth, "Visualization and simulation of ion thrusters possibly usable by small satellites, In Selected papers of 6th International Conference on Research, Technology and Education of Space (H-SPACE 2020)." HSPACE2020-FP-59, 2020: Hungarian Astronautical Society, Feb. 2020. [Online]. Available: http://space.bme.hu/wp-content/uploads/2021/01/Proceedings_Papers_HSPACE-2020.pdf#page=62&zoom=74,0,9

[11] D. Krejci and P. Lozano, "Space propulsion technology for small spacecraft," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 362–378, 2018.

[12] E. Kulu, 'What is a CubeSat?', Nanosats Database. [Online]. Available: https://www.nanosats.eu/cubesat [Accessed: Dec. 18, 2021]

[13] ThrustMe, "ThrustMe," 2021, https://www.thrustme.fr/. [Online]. Available: https://www.thrustme.fr/

[14] R. G. Jahn and E. Y. Choueiri, "Electric Propulsion," in *Encyclopedia of Physical Science and Technology*. Elsevier, 2003, pp. 125–141. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B0122274105002015 **DOI**: 10.1016/B0-12-227410-5/00201-5

[15] M. CHEN, A. SUN, C. CHEN, and G. XIA, "Particle simulation of grid system for krypton ion thrusters," *Chinese Journal of Aeronautics*, vol. 31, no. 4, pp. 719–726, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936118300384 **DOI**: 10.1016/j.cja.2018.01.020

[16] R. W. Hockney and J. W. Eastwood, Computer Simulation Using Particles, 1st ed. Bristol England; Philadelphia: CRC Press, Jan. 1989.

[17] MATLAB, "Partial Differential Equation Toolbox," 2021. [Online]. Available: https://www.mathworks.com/products/pde.html

[18] K. Simonyi, *Theoretische Elektrotechnik*. Leipzig: Barth, 1993.

[19] K. P. Murphy, Machine learning: a probabilistic perspective. Cambridge, Mass. [u.a.]: MIT Press, 2013.

[20] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús, *Neural Network Design*, 2nd ed. Wrocław: Martin Hagan, Sep. 2014.

Árpád László Makara is currently a PhD student at the Department of Broadband Infocommunications and Electromagnetic Theory of the Budapest University of Technology and Economics (BME). The topic of his master thesis work was the simulation and optimization of electrostatic ion thrusters for CubeSats. For two semesters, he held exercises at BME for BSc electrical engineering students on the subject of Introduction to Electromagnetic Fields. His current research topic is relating to millimeter wavelength indoor propagation for 5G, and his first results in this field were presented on the EFOP-3.6.2 workshop in November, 2020.

András Reichardt is currently a assistant lecturer at Budapest University of Technology and Economics (BME), Department of Broadband Infocommunications and Electromagnetic Theory. His current research interests focus on electromagnetic field calculation, mathematical modeling of complex systems, and equivalent circuit representations of the problems. He currently teaches in many areas such as nanoscience, electromagnetic theory, circuit theory, and discrete signal analysis.

László Csurgai-Horváth is employed as an associate professor at Budapest University of Technology and Economics (BME), Department of Broadband Infocommunications and Electromagnetic Theory. His current research interests are focused on indoor and outdoor propagation, measurements and modelling of the rain attenuation on terrestrial and satellite radio links, time series synthesis and cognitive spectrum management. He has been involved in several international projects such as SatNEx, COST Action IC-0802 and QoSMOS. Recently, he has led some ESA-founded research projects relating to the Alphasat propagation and communications experiment and an ESA technology transfer demonstrator project for 5G indoor propagation measurements. Currently, he teaches space technology and channel modelling at BME in Budapest, Hungary.

Identification of the Possible Security Issues of the 464XLAT IPv6 Transition Technology

Ameen Al-Azzawi, and Gábor Lencse

Abstract—**This paper focuses on one of the most prominent IPv6 transition technologies named 464XLAT. The aim is to analyze the security threats that this technology might face. After carrying out the threat analysis using STRIDE method that stands for Spoofing, Tampering, Repudiation, Information Disclosure and Elevation of Privilege, and using DFD (Data-Flow Diagram) as a core for the analysis, we summarized the security vulnerabilities and attack points possibilities within this infrastructure. We have also built a testbed for 464XLAT topology using several virtual machines, which were created using Debian image. We used our testbed to perform DoS (Denial of Service) attack against the PLAT (provider-side translator) and monitor PLAT's performance and the number of packets being translated under attack by different number of clients using the hping3 command.**

Index Terms—**464XLAT, DNS, IPv4aaS, IPv6, STRIDE, Translation**

I. INTRODUCTION

After the depletion of the IPv4 address pool, several technologies were invented to provide a practical solution for this matter. The high number of *IPv6 transition technologies* are surveyed and they are classified into different categories regarding the importance of their security analysis in [1]. The methodology for the identification of potential security issues of different IPv6 transition technologies has been defined in [2]. That paper also includes a detailed security analysis of DNS64 (DNS extensions for network address translation from IPv6 clients to IPv4 servers) [3] and a shorter security overview of stateful NAT64 (Network address and protocol translation from IPv6 clients to IPv4 servers) [4]. The combination of DNS64 plus NAT64, however, has its own drawbacks, especially that the connection can only be established from the IPv6 only client and not supporting IPv4 literals [5]. Then came 464XLAT [6] with its double translation mechanism, where it did sort out the issues presented by the DNS64 +NAT64 solution.

However, the application of 464XLAT may involve various security vulnerabilities. Therefore, it is essential to analyze the security threats that might affect this promising technology. According to [6], 464XLAT in general is very quick to deploy and has minimal IPv4 resource requirements and maximum IPv4 address sharing efficiency. Moreover, 464XLAT employs traffic engineering and capacity planning without the indirection or obfuscation of a tunnel [6].

Previously, we have published a conference paper [7], in which we have analyzed the potential vulnerabilities of 464XLAT. In this paper we are taking it one step further by building 464XLAT topology with Linux based virtual machines and actually monitoring the operation of 464XLAT and the performance of its translators under DoS attack. (This paper is an extension of our former conference paper[7].)

The main focus of this paper is to highlight security threats facing the network infrastructure as a result of deploying 464XLAT within the network topology and building the testbed where an actual topology is tested with several attacking clients.

In Section II, we discuss the operation of 464XLAT and its structure, section III is about operation of the STRIDE method, its elements and how it works, section IV is about 464XLAT security revealed by applying STRIDE on it, while in section V, we mention some previous publications regarding 464XLAT / NAT64 security threats and in section VI, we build the testbed and explain its infrastructure and its topology elements. In section VII, we demonstrate an attack scenario using hping3 command and adding several clients gradually. In section VIII, we analyze the results of our attack. In section IX, we present our plans for future research and the significance of our results. In section X, we summarize and conclude the value of the results the paper came up with where we prove that 464XLAT is effective technology and also susceptible to attack possibilities such as DoS.

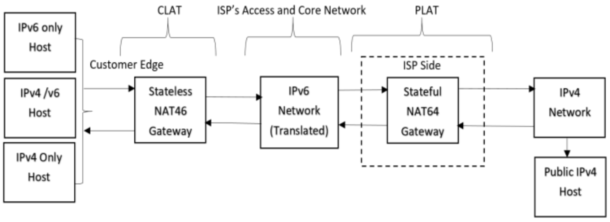


Fig. 1. Overview of 464XLAT Architecture

A. Azzawi is with the Department of Networked systems & Services, Budapest University of Technology & Economics, Budapest, Hungary.
(e-mail: alazzawi@hit.bme.hu)
G. Lencse is with the Department of Networked systems & Services, Budapest University of Technology & Economics, Budapest, Hungary.
(e-mail: lencse@hit.bme.hu)

Submitted: May 11, 2021.

DOI: 10.36244/ICJ.2021.4.2

II. THE OPERATION OF 464XLAT

The main structure of 464XLAT, as shown in Fig. 1, is divided in two sides; CLAT & PLAT.

A. CLAT (customer-side translator)

CLAT algorithmically translates 1:1 private IPv4 addresses to global IPv6 addresses and vice versa [6]. It acts as IPv6 router, DNS proxy and DHCP server for local client as well. Normally, CLAT must know its own prefix and PLAT side prefix in order to use it as destination for its outgoing packets [6].

As for IPv6 client, its own packets will pass through the CLAT without the need to any translation process and will be forwarded to the PLAT directly.

B. PLAT (provider-side translator)

It translates N:1 global IPv6 addresses with the previously set CLAT prefix to public IPv4 addresses and vice versa [6], it actually implements a stateful NAT64 gateway as described in RFC 6146 [4]. We give an easy introduction to understand to the operation of 464XLAT by Fig. 2. The client in the bottom left hand side corner of the figure (using private IPv4 address 192.168.1.2) wants to connect to the server in the top left hand side corner (using public IPv4 address 198.51.100.1). The prefix at CLAT side is 2001:8db:aaaa::/96, whereas the prefix at PLAT side is 2001:8db:1234::/96. CLAT translates the IPv4 packet into an IPv6 packet, in which the source address will be 2001:db8:aaaa::192.168.1.2, and the destination address will be 2001:db8:1234::198.51.100.1.

At the PLAT side, the 2001:db8:1234::/96 prefix is discovered in the destination address, and an IPv4 packet is built using the embedded 198.51.100.1 IPv4 address as destination address, and the source IPv4 address is chosen from the pool of 192.0.2.1-192.0.2.100 (this time it happened to be 192.0.2.1). Source port is also replaced, when needed, and the connection is registered into the state table of the NAT64 translator to be able to perform the stateful translation in the reverse direction, too. (Please refer to RFC 6146 [4] for further details of the stateful NAT64 translation.)

Besides double translation, there are two other possible scenarios. If both the client and the server have IPv6 addresses, then there is no translation at all, but native IPv6 is used. If the client has an IPv6 address, but the server has only and IPv4 address, then there are two possible modes of operation:

- If DNS64 is configured, then the DNS64 server returns an IPv4-embedded IPv6 address, and only a single translation happens at the PLAT. (This is the DNS64 + NAT64 solution.).
- If no DNS64 is configured, then the client uses IPv4 and double translation happens as described above.

III. THE OPERATION OF STRIDE

STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege [8]. These are general threats that any network device/node might be susceptible to.

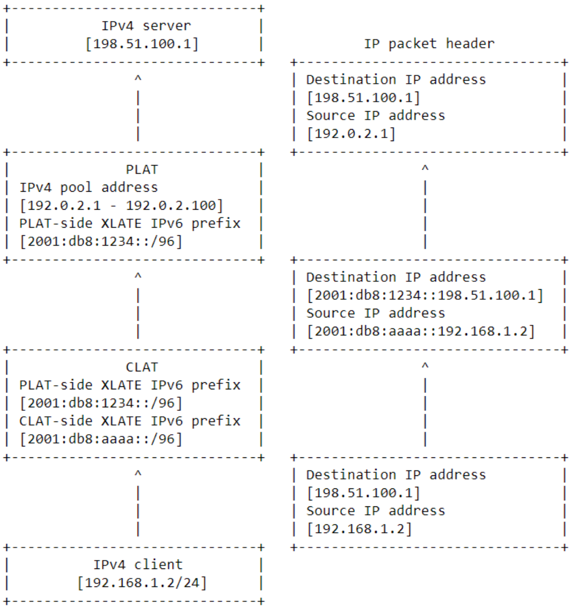


Fig. 2. 464XLAT Packet Processing

A. *Spoofing:* when an attacker claims to be someone else by changing his real source IP address in order to bypass a filter or an IDS (intrusion detection system) and also to perform DDOS (Dedicated DoS attack) against the potential target [7][8].

B. *Tampering:* the process of changing the content of data flow on its way to the destination, for example, the attacker might alter the packet destination to a malicious server [8]. In wireless communication however, a possibility of MIM (man in the middle) can be used to achieve this attack, such as intercepting HTTP and HTTPS connection between HTTP(S) such as mobile or desktop browser [9].

C. *Repudiation:* it is the claim of a user of not doing an act, while he actually did, like DNS resolution request or in case of ATM money withdrawal where customer might withdraw a specific amount of money then claims that he has not performed any transaction [2]. This threat often appears on the business layer (above network layer in TCP/IP or above application layer such as HTTP/HTML).

D. *Information Disclosure:* an attacker gets sensitive information, which could be used in various ways, e.g. it might help him in hacking, like knowing who's talking to whom by monitoring DNS traffic or TTL value of the packet, which gives the attacker an idea of many hops has the packet gone through then the packet original or source address location will be compromised [2].

E. *Denial of Service:* The attacker can flood a system with useless requests to consume as much processing power as possible in order to prevent the targeted machine from serving legitimate (useful) ones. For example, it can flood a DNS server with huge number of useless queries to prevent legitimate queries from getting a response [2].

TABLE I. VULNERABILITIES OF DIFFERENT DFD ELEMENTS [2]

DFD Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data Flows		✓		✓	✓	
Data Stores		✓		✓	✓	
Processes	✓	✓	✓	✓	✓	✓
Interactors	✓		✓			

F. *Elevation of Privilege:* bypassing the authority matrix of specific organization, like getting root permission on a specific server [9].

The STRIDE method uses the DFD (Data Flow Diagram) of the investigated system in order to examine the critical areas within the system, so it comes up with total security analysis using the four types of elements of the DFD (Data Flows, Data Stores, Processes and Interactors).

Data flow models usually applied on network & architecture systems rather than software products, but they can be applied on both [8]. STRIDE has different approaches regarding threat models:

- Assets-centered threat model: anything the attacker wants to access, control or damage. According to [10], assets-centered threat model is being conducted using 4 approaches: DREAD, Trike, OCTAVE and PASTA. For instance, OCTAVE, which stands for Operationally Threat Asset and Vulnerability Evaluation, is a robust approach but its rather complicated, it takes considerable time to learn and get familiar with its process. Furthermore, its documentation is voluminous [10].
- Attacker-centered threat model: it is based on knowing the attacker, his motivations and skills. It is useful but hard to implement [8].
- Software-centric threat model: focuses on software being built and the deployed systems, it's the best approach for threat modeling [8], because it supposes that software developers are the best people to understand the software they are developing, which makes the software an ideal starting point to trigger the threat modeling process.

In general, the best models are diagrams that help participants understand the software and find threats against it. Each element of the DFD has its own security threats as explained in Table I. It means each element is susceptible to some threats while not susceptible to others [8].

IV. SECURITY ISSUES OF 464XLAT

We presented DFD of 464XLAT in a previous paper [11]. Nevertheless, we made some slight changes on the DFD and after applying the STRIDE method on the DFD diagram of 464XLAT in Fig.3, some security threats are visible at the points (1-11), which represent the threat possibilities within the DFD diagram. In this section, we carefully examine all the elements of the DFD for all possible threats & attacks in details. (Please see the summary of vulnerabilities in Table II.)

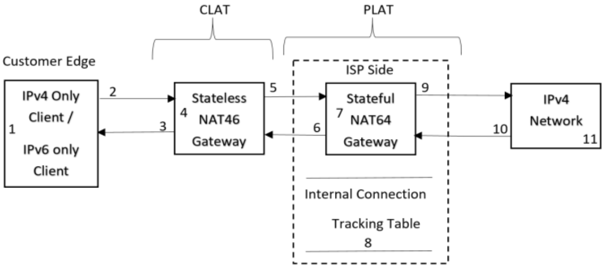


Fig.3 DFD for the threat Analysis of 464XLAT

A. IPV4 / IPV6 Client

1) Spoofing:

- The spoofed IP could be used to send useless packets to the CLAT and this scenario is considered as DOS attack against the CLAT.

2) Repudiation: the client might deny the request he made in the first place.

B. Data flow from IPv4 only client to NAT46

1) Tampering: it can be used as an attack against the domain name or changing the IP address of packet destination, which might be used to direct the packet towards fraudulent server, this kind of attack is also called FoS (Failure of Service) because it prevents the real client from receiving an answer to its real query [2].

2) Information Disclosure: an attacker might be interested in knowing the browsing habits of the requester, and the packet itself might contain some sensitive information sent by the client himself [2].

3) Denial of Service: flooding the gateway with unwanted requests to prevent the real query from getting an answer.

C. Data flow from NAT46 gateway to the client

1) Tampering: an attack against the client, for example sending misleading information at application level or breaking the connecting sending a RST at TCP level.

2) Information Disclosure: an attacker getting access to sensitive data.

3) Denial of Service: sending high number of forged replies to the client to prevent if form processing the genuine ones.

D. NAT46 Gateway (CLAT)

1) Spoofing: in this case, spoofing means unauthorized user controls the gateway and translate the private IPv4 to the wrong IPv6 and send the packet to different destination.

2) Tampering: an attacker tampered with the data within the gateway itself by which might result in e.g. returning the wrong IPv6 address [2].

3) Repudiation: after spoofing the CLAT, an attacker might deny sending a packet that was actually sent by the CLAT himself while hiding his own identity. Logging is the key here, if the database administrator is not fully trusted, then a system in another privilege domain has to be installed.

4) Information Disclosure: an attacker might make use of the browsing data and queries made by the requester in order to hack the main requester later on.

5) Denial of Service: it could be by an attacker spoofing an IP of legitimate user flooding the CLAT with huge number of requests, see section IV.B.3.

6) Elevation of Privilege: it happens when an attacker gain access to a service he shouldn't be getting in the first place. However, it mainly happens due to inside job [2] and the attacker might gain the right of admin or root to whatever he likes later on.

E. Data flow from NAT46 to NAT64

1) Tampering: the packet destination IP might be altered while it's on its way to NAT64 gateway.

2) Information Disclosure: see section IV.B.2.

3) Denial of Service: after spoofing the NAT46, attacker might send numerous useless packets to the NAT64 gateway.

F. Data flow from NAT64 to NAT46 gateway

1) Tampering: see section IV.C.1.

2) Information Disclosure: it is possible that an attacker might access the packet details on its way back to NAT46 gateway and extract sensitive information out of it.

3) Denial of Service: flooding the NAT46 gateway with unwanted packets to prevent it from translating the genuine traffic.

G. NAT64 Gateway (PLAT)

1) Spoofing: an attacker might take control (spoof) the gateway and do many malicious activities with it, see section IV.D.1.

2) Tampering: an attacker might change the content of packet details withing the gateway, see section IV.D.2.

3) Repudiation: see section IV.D.3.

4) Information Disclosure: see section IV.D.4.

5) Denial of Service: DoS attack might come in a way that affect the NAT64 Gateway (PLAT), such as Exhaustion of source port and public IPv4 address pool, which is an issue since the gateway uses 63K[1] number of source ports per public IPv4 address. An enhanced algorithm presented by [12] helps in tackling this issue in details.

6) Elevation of Privilege: one of the elevation problems is called buffer overflow attack [13], which could happen if a device like NAT64 getting inputs from both sides and that might affect its memory storage units.

H. Internal connection tracking table

1) Potential attackers have no direct access to it, they can influence its content in indirect ways only.

1) Denial of Service: The attacker may initiate fake connections (either using his real IPv6 address or fake ones) and thus achieve the insertion of fake entries into the connection tracking table. The high number of fake entries may slow down the operation of the NAT64 gateway or even prevent legitimate users from establishing further connections, when the table is full. If PLAT applies a connection limit per source IPv6 address, then the attacker may exhaust the available number of connections for legitimate users by spoofing their IPv6 addresses and initiating fake connections.

I. Data flow from PLAT to IPv4 Server

1) Tampering: attacker might change the source IP address of the packet so the IPv4 server will not know, who sent the packet in the first place.

2) Information Disclosure: see Section IV.B.2

3) Denial of Service: an attacker might spoof the IP and flood the IPv4 server with plenty of undesired requests.

J. IPV4 server / IPv4 network

1) Spoofing: Source IP address might be spoofed, see section IV.A.1.

2) Repudiation: denying of sending a request is viable in this case, *see section IV.A.2.*

K. Data flow from IPV4 server / IPv4 network to PLAT

1) Tampering: the attacker might send TCP-RST packets to erase the mapped entries within the NAT64 gateway.

2) Information Disclosure: it is possible that an attacker might access the packet details on its way back to NAT64 gateway and extract sensitive information out of it, like TTL value or browsing habits [2].

3) Denial of Service: flooding the *NAT64* gateway with unwanted requests to prevent it from translating the real traffic.

L. Summary of the results

To summarize the attacks or the vulnerabilities within 464XLAT structure, we concluded the following threats:

A. Spoofing of NAT46 or NAT64 gateways results in altering packets destination or returning the wrong IP address to the requester.
B. DoS: denying access of a legitimate user to his authorized traffic and obstructing the function of NAT46 & NAT64 gateways.
C. FoS: preventing the real client from receiving an answer to its real query.
D. Leaking of confidential information like IP address, TTL value and browsing habits.
E. Tampering with NAT64 tracking table: loosing of mapped entries.
F. Privileges level altering: getting root privilege will increase the inside job attack very often.
G. Buffer overflow attack in case of NAT64, which affects the storage (connection tracking table) entries and might erase them accidently.

---

[1] Similarly to NAT devices, NAT64 gateways usually use source port numbers from the range of 1024 – 65536.

TABLE II. SUMMARY OF 464XLAT THREATS

DFD Element	***Threat***	***Possible attacks***
1	Spoofing & Repudiation	DoS attack against the CLAT
2, 3	Tampering, Information Disclosure and Denial of Service	FoS, collecting unauthorized information, DoS
4	All STRIDE Elements	FoS, DoS and unauthorized access,
5, 6	Tampering, Information Disclosure and Denial of Service	FoS, collecting unauthorized information, DoS
7	All STRIDE Elements	FoS, DoS and unauthorized access,
8	Only indirect attacks	Tampering with Connection Tracking Table; DOS attack (exhaustion of connection tracking table, slowing down look up speed)
9, 10	Tampering, Information Disclosure and Denial of Service	FoS, collecting unauthorized information, DoS
11	Spoofing & Repudiation	DoS attack against the PLAT

V. RELATED WORK

Very few papers have been published regarding our topic. However, [14] has focused on the IPv6 security issues as far as cellular networks concern and it came up with different categories of possible attacks. They demonstrated three different DoS attacks on NAT64 block targeting features that only exist in IPv6 cellular networks:

A. *NAT overflow attack:* According to [14], most of service providers tend to drop the source address of a spoofed packet and replace it with a public IPv4 address. Therefore, a host can send & receive packets using single private IPv4 address assigned by NAT.

As a result, the maximum of external mapping for single targeted service is 65,535. Meanwhile, in IPv6 cellular networks, a device can utilize 2^{64} IPv6 addresses. So, if a device creates mapping on NAT64 using all the 2^{64} IPv6 addresses, the result will be 65,535 * 2^{64} mappings, which can lead to overload for NAT64 [14] . It also showed that NAT64 gateway will stop the mapping process for any incoming request after 1500 entries (depending on the preset value) within its tracking table (if the requester is sending from the same IP address targeting the same service) and sends back TCP-RST packet back to the requester as response for the TCP-SYN packet. However, this policy of NAT64 can be exploited as DoS attack [14].

B. *NAT wiping attack*: The targeted victim in this case is the mapping entry itself. NAT64 uses the N:1 mapping criterion. If an adversary targets the external IPv4 of NAT64 gateway, N hosts are sharing the same external IPv4 address will be liable to DoS attack. The adversary will send malicious TCP-RST packets to wipe out the target mappings within the NAT64. As a result, the mapped users to the very same external IPv4 address will be denied access to their service.

To do so, the attacker needs to know the TCP 5-tuple of the targeted service (Protocol, Destination IP address, port number, External IP address of NAT64 and External port number of NAT64).

C. NAT Bricking attack: it's type of DoS attack which also exploits the N:1 mapping algorithm adopted by NAT64. Basically, the adversary can send huge number of requests using the external IPv4 address(es) of the NAT64 gateway [14]. However, big vendors (google, YouTube, etc.) have IP blocking approach if it exceeds specific number of requests per minute. Nevertheless, [14] has done an experiment to target Google scholar [2] website, which is an IPv4 based site. So, the IPv6 cellular host sends 150 requests per minute to trigger CAPTCHA request. Every time CAPTCHA request emerges, adversary source IP address is being changed by turning the airplane mode on and off, this process was repeated 1000 times. Finally, the NAT bricking attack was able to trigger CAPTCHA request for a total of 631 external IPv4 from Google Scholar, including one of the victim's external IP address [14].

Moreover, [15][15] has explained that the majority of the transition technologies use some form of NAT, NAT44, NAT64, NAT46, etc. and how it is a myth that NAT is putting the user inside this secured box of protective shield from the outside attackers, the sequence of communications below explains how vulnerable the NAT client could be:

1- Attacker attracts the victim towards specific website.
2- Victim clicks on the malicious URL and enters the page.
3- The page has a hidden form connecting to http://attacker.com:6667 (IRC port).
4- The victim submits the form without his consent.
5- An HTTP connection is created to the (fake) IRC server.
6- The form as well has hidden value which sends:" OPEN DCC CHAT PORT".
7- Router sees an "IRC connection" then open a port back through the NAT.
8- The attacker now has an open path to the network.

The very same process could have been applied using FTP NAT helper if not IRC.

According to [15], todays preferred transition technologies are 6rd, DS-Lite and 464XLAT, while risk Mitigation Strategies can be summarized as follow:

1- Minimizing the need for SP-NAT (Service-Provider NAT).

---

[2] Google, Google Scholar. https://scholar.google.com.

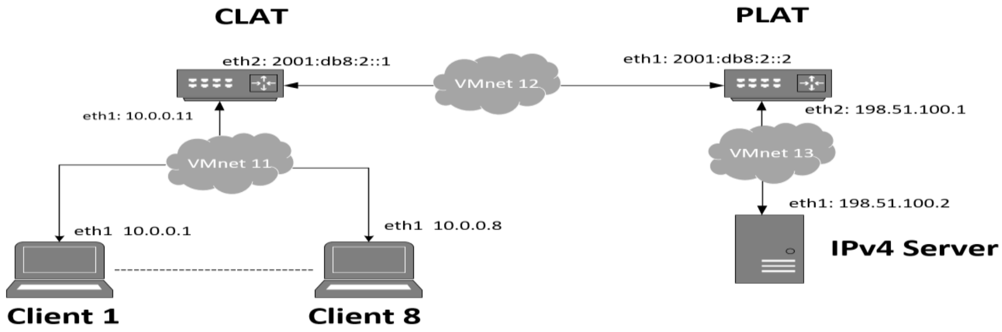


Fig. 4 464XLAT Testbed

2- The more IPv6 established sessions, the less you rely on SP-NAT and all the security issues associated with that.
3- Search for a transition plan that uses native IPv6 such as 464XLAT & DS-Lite.

As for the testbed, several attempts were conducted by researchers to build an efficient testbed in order to test the transition technologies, their weak spots and vulnerabilities. A successful testbed was built by Marius Georgescu [16], in which he measured the latency, throughput and packet loss by adopting 464XLAT transition technology and some other methods as well.

VI. 464XLAT TESTBED

The testbed was built through remote access to a Windows-based virtual machine with the following specifications:

- Intel(R) Xeon(R) Silver 4215 CPU @ 2.50GHz, (16 VCPUs)
- 16.0 GB RAM
- 64-bit Windows 10 operating system.

Further virtualization was created by installing VMware workstation 12 Player. Several virtual machines were created using Debian image, which was created by using the debian-vm tool of Daniel Bakai [3]. Every machine had Debian 8.9 operating system.

A. Test Topology

The topology of the 464XLAT testbed is shown in Fig. 4, it can be divided into two sides:

- On the left side, there are four clients (10.0.0.1 -- 10.0.0.8) and the CLAT.
- On the right side, there are the PLAT and the IPv4 server.

B. Testbed Implementation

In our testbed, each virtual machine has the following specifications:

- Clients 1-8: 1GB RAM, 1 CPU core, and 20 GB hard disk.
  CLAT: 1GB RAM, 3 CPU cores, and 20 GB hard disk.
- PLAT: 1GB RAM, 3 CPU cores, and 20 GB hard disk.
- IPv4 Server: 1GB of RAM, 1 CPU core, and 20 GB hard disk.

Furthermore, the topology was build using three separated virtual networks: VMnet11, VMnet12, and VMnet13.

- VMnet11: the network between the eight clients and CLAT eth1. The network is IPv4 only.
- VMnet12: the network between CLAT eth2 and PLAT eth1. The network is IPv6 only.
- VMnet13: the network between PLAT eth2 and IPv4 server eth1. The network is IPv4 only.

Table III shows the Linux and VMware settings used for each virtual machine.

C. Test configuration

The main configuration blocks are within CLAT (stateless translator) and PLAT (stateful translator).

In both cases, the configurations included three steps:

- configuring TAYGA to run
- configuring the operating parameters of TAYGA
- further settings

The details of their settings are presented below.

1) Configuring CLAT

It was set in the **`/etc/default/tayga.conf`** file that TAYGA should be started at operating system boot time:

```
RUN="yes"
CONFIGURE_NAT44="no"
```

Here, we did not intend to use TAYGA as stateful NAT64 because CLAT is a stateless NAT46 translator.

The operating parameters of TAYGA were set in the **`/etc/tayga.conf`** file as follows:

```
tun-device nat64
ipv4-addr 10.0.0.9
ipv6-addr 2001:db8:2::9
prefix 2001:db8:a::/96
map 10.0.0.1 2001:db8:c::10.0.0.1
map 10.0.0.2 2001:db8:c::10.0.0.2
map 10.0.0.3 2001:db8:c::10.0.0.3
map 10.0.0.4 2001:db8:c::10.0.0.4
map 10.0.0.5 2001:db8:c::10.0.0.5
map 10.0.0.6 2001:db8:c::10.0.0.6
map 10.0.0.7 2001:db8:c::10.0.0.7
map 10.0.0.8 2001:db8:c::10.0.0.8
```

---

[3] D. Bakai, "Debian VM", [Online]. Available: https://git.sch.bme.hu/bakaid/debian-vm

TABLE III. LINUX AND VMWARE NETWORK SETTING FOR VIRTUAL MACHINES

VM name	Clients 1-8	CLAT	PLAT	IPv4 Server
eth0 Linux setting	DHCP	DHCP	DHCP	DHCP
eth1 Linux setting	Static IPv4: 10.0.0.1-8	Static IPv4: 10.0.0.11	Static IPv6: 2001:db8:2::2/64	Static IPv4: 198.51.100.2
eth2 Linux setting	N/A	Static IPv6: 2001:db8:2::1/64	Static IPv4: 198.51.100.1	N/A
eth0 VMware setting	NAT	NAT	NAT	NAT
eth1 VMware setting	VMnet11	VMnet11	VMnet12	VMnet13
eth2 VMware setting	N/A	VMnet12	VMnet13	N/A

As for further settings, the following bash shell script was responsible for setting up routes and enabling Linux kernel routing:

```
#!/bin/bash
ip route add 198.51.100.0/24 dev nat64
ip route add 2001:db8:c::/96 dev nat64
ip route add 2001:db8:a::/96 via 2001:db8:2::2
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
ip route del 2001:db8:a::/96 dev nat64
```

We note that the last command was to delete a routing rule that was automatically set by TAYGA.

2) *Configuring PLAT*

It was set in the `/etc/default/tayga.conf` file that TAYGA should be started at operating system boot time:

```
RUN="yes"
CONFIGURE_NAT44="no"
```

Here, we used TAYGA as a stateful NAT64 translator, but we set the `iptables` rule by hand (see below).

The operating parameters of TAYGA were set in the `/etc/tayga.conf` file as follows:

```
tun-device nat64
ipv4-addr 198.51.100.9
ipv6-addr 2001:db8:2::9
prefix 2001:db8:a::/96
map 10.0.0.1 2001:db8:c::10.0.0.1
map 10.0.0.2 2001:db8:c::10.0.0.2
map 10.0.0.3 2001:db8:c::10.0.0.3
map 10.0.0.4 2001:db8:c::10.0.0.4
map 10.0.0.5 2001:db8:c::10.0.0.5
map 10.0.0.6 2001:db8:c::10.0.0.6
map 10.0.0.7 2001:db8:c::10.0.0.7
map 10.0.0.8 2001:db8:c::10.0.0.8
```

As for further settings, the following `bash` shell script was responsible for setting up routes and enabling Linux kernel routing:

```
#!/bin/bash
ip route add 10.0.0.0/24 dev nat64
ip route add 2001:db8:a::/96 via 2001:db8:2::1
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

Furthermore, the below `iptables` command was applied:

```
iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
```

The aim of this command is to perform a stateful NAT44 translation using the well-known Netfilter framework. It was necessary, because TAYGA is only a stateless NAT64 translator by itself, and thus it requires an additional stateless NAT44 translator to implement stateful NAT64.

VII. SAMPLE DOS ATTACK

The aim of the DoS attack is to exhaust the resources of the PLAT device. It is carried out by sending high frequency ping requests from the eight clients to the IPv4 server using the `hping3` command.

To carry out the attack, SSH authentication were established between client one (10.0.0.1) and the rest of the network elements in order to be able to carry out the experiments by a script. The script was responsible for starting traffic capture by using `tshark` and for starting the attacking program on all clients at (mostly) the same time by a script.

The attack was through a `hping3` command with specific arguments:

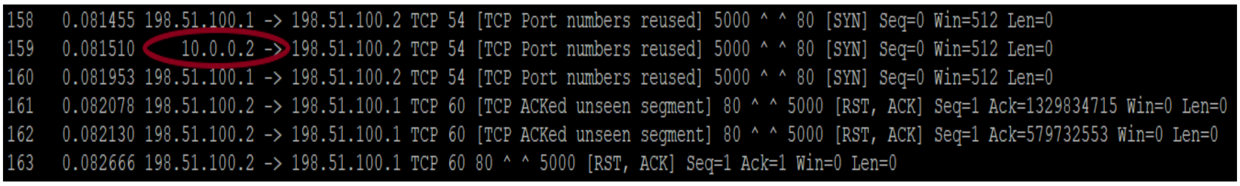
```
hping3 -S -p80 -s5000 -k 198.51.100.2 -iu1500
```

-S: TCP Syn attack.
-p: destination port number.
-s: source port number.
-k: to maintain the same port number and avoid its increment.
-iu: to control the number of sent packets per second.

We note that the last parameter does not directly set the packet rate, but it specifies a kind of targeted delay in microseconds. Different packet rates can be applied by manipulating the “-iu” argument of the `hping3` command (e.g. 100 packets/s, 1000 packets/s, etc.). The experiments were carried out using various packet rates. For the demonstration of the DoS attack, we selected a rate (about 460 packets/s) at which almost all packets were correctly translated by the PLAT, when only a single client was used, but a significant amount of the frames were not correctly translated, when 8 attacking clients were used. Please see a fragment of the `tshark` output in Fig 5. The incorrectly translated frame is highlighted by a red oval.

The attack process was as below:

a. Start the measurements, then wait for 10 seconds to monitor the performance before the attack.
b. Start attack with client 1, then wait for 100 seconds to monitor the effect of one client.



```
158   0.081455 198.51.100.1 -> 198.51.100.2 TCP 54 [TCP Port numbers reused] 5000 ^ ^ 80 [SYN] Seq=0 Win=512 Len=0
159   0.081510     10.0.0.2 -> 198.51.100.2 TCP 54 [TCP Port numbers reused] 5000 ^ ^ 80 [SYN] Seq=0 Win=512 Len=0
160   0.081953 198.51.100.1 -> 198.51.100.2 TCP 54 [TCP Port numbers reused] 5000 ^ ^ 80 [SYN] Seq=0 Win=512 Len=0
161   0.082078 198.51.100.2 -> 198.51.100.1 TCP 60 [TCP ACKed unseen segment] 80 ^ ^ 5000 [RST, ACK] Seq=1 Ack=1329834715 Win=0 Len=0
162   0.082130 198.51.100.2 -> 198.51.100.1 TCP 60 [TCP ACKed unseen segment] 80 ^ ^ 5000 [RST, ACK] Seq=1 Ack=579732553 Win=0 Len=0
163   0.082666 198.51.100.2 -> 198.51.100.1 TCP 60 80 ^ ^ 5000 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

Fig. 5. PLAT eth2 **`tshark`** capture

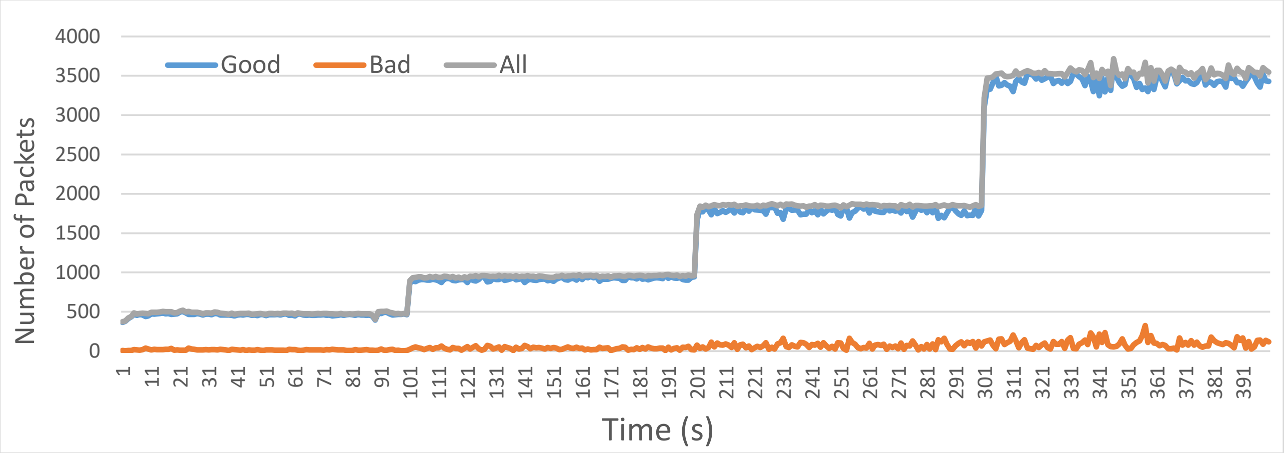


Fig. 6 Measurements with 460 packets/s per client load: the number of good/bad/all packets as a function of time (number of attacking clients: 1, 2, 4, and 8)

c. Start attack with client 2, then waiting for 100 seconds.
d. Start attack with client 3 & client 4, then wait for 100 seconds.
e. Start attack with clients 5, 6, 7, 8, then wait for 100 seconds.
f. End the measurements, then stop the eight attacks.

VIII. RESULTS AND ANALYSIS

Fig. 6 shows the number of correctly translated, not translated and all frames per second as a function of time in case of 460 packets/s rate. It is fairly obvious that the number of sent packets are increasing by doubling the number of clients (1, 2, 4, 8). Every spike in the graph represents new added client(s).

- From 0-100 second (only client1).
- From 100-200 second (client 1& 2).
- From 200-300 second (client 1,2,3 and 4).
- From 300-400 second (client 1,2,3,4,5,6,7 and 8).

The number of packets arriving at the PLAT is slightly fewer than the ones sent from the CLAT. The average value of sent packets when 8 clients were applied was around 3500 packets /s. Furthermore, the orange line represents the number of untranslated packets by **`iptables`**.

We divided the packets into three types (good, bad, all). "Good" label represents the properly translated packets, "Bad" label represents packets that failed in the masquerading process and kept their original source IP address. As for "All" label, it is fairly obvious that it represents the total number of good and bad packets together.

The MASQUERADING feature is supposed to change the source IP address of packets leaving the PLAT and heading towards IPv4 server. That means every packet heading towards IPv4 server should have the source IP address of PLAT eth2 interface (198.51.100.1). However, some packets are passing through the filter without getting translated by keeping their original IP address (10.0.0.1-8) instead of PLAT eth2 interface IP address (198.51.100.1) as shown in Fig. 5. We don't know the exact root cause for this behavior. What we know is the frequency of these untranslated packets increases by increasing the number of applied clients as illustrated by Fig. 6.

IX. PLANS FOR FUTURE RESEARCH AND THE SIGNIFICANCE OF OUR RESULTS

A. Plans for Future Research

As for our future research focus, we plan to implement other types of DoS attacks against the PLAT in order to exhaust:

- its TCP source port number pool,
- its internal connection tracking table.

We are also planning to find mitigation for this and for other types of attacks, too.

B. Significance of our Results

464XLAT is very important for network operators, especially ISPs (Internet Service Providers). It is among the five most important IPv4aaS technologies that has to be supported by all customer edge routers [17]. It has several advantages like its port number efficiency and its wide spread support in cellular networks [18]. Therefore, its security properties may be extremely important decision factors for network operators.

Other areas of applications include any kinds of IoT [19] and/or LoRaWAN systems [20], where IPv6 may be needed in the access network due to the high number of devices, but some legacy devices still need IPv4 support. Further important application segment is the intelligent transportation systems in smart cities [21] for the same reason.

X. Conclusion

Threat analysis of the 464XLAT IPv6 transition technology was performed by applying the STRIDE method in order to point out the potential vulnerabilities of the technology. This method of using the data flow diagram of the 464XLAT system to analyze its potential security vulnerabilities has proven its efficiency.

The double translation mechanism of 464XLAT proved its effectiveness in terms of IPv4 literals communications over IPv6 infrastructure. However, it has some security issues and vulnerabilities such as DoS attack possibility. Some faulty translated packets were monitored and their percentages increased by adding more load to the topology and therefore affects the PLAT performance. Some readings have visible fluctuation and this was mainly caused by the instability of hping3 command and its packet rate controlling ratio. In general, the experiment proved that 464XLAT is an effective transition technology to establish a stable connection over different IP versions infrastructure.

References

[1] G. Lencse and Y. Kadobayashi, “Comprehensive Survey of IPv6 Transition Technologies: A Subjective Classification for Security Analysis”, *IEICE Transactions on Communications*, vol. E102-B, no.10, pp. 2021-2035. **doi**: 10.1587/transcom.2018EBR0002

[2] G. Lencse and Y. Kadobayashi, “Methodology for the identification of potential security issues of different IPv6 transition technologies: Threat analysis of DNS64 and stateful NAT64”, Computers & Security, vol. 77, no.1, pp. 397-411, 2018, **doi**: 10.1016/j.cose.2018.04.012.

[3] M. Bagnulo, A Sullivan, P. Matthews and I. Beijnum, “DNS64: DNS extensions for network address translation from IPv6 clients to IPv4 servers”, IETF RFC 6147, 2011, **doi**: 10.17487/RFC6147.

[4] M. Bagnulo, P. Matthews and I. Beijnum, “Stateful NAT64: Network address and protocol translation from IPv6 clients to IPv4 servers”, IETF RFC 6146, 2011, **doi**: 10.17487/RFC6146.

[5] S. Répás, T. Hajas, G. Lencse, “Application compatibility of the NAT64 IPv6 transition technology,” in Proc. 37th International Conference on Telecommunications and Signal Processing, Berlin, 2014, pp. 49–55, **doi**: 10.1109/TSP.2015.7296383.

[6] M. Mawatari, M. Kawashima, C. Byrne, “464XLAT: Combination of stateful and stateless translation, IETF RFC 6877 (2013).

[7] A. Al-Azzawi and G. Lencse, "Towards the Identification of the Possible Security Issues of the 464XLAT IPv6 Transition Technology," 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), 2020, pp. 439-444, **doi**: 10.1109/TSP49548.2020.9163487.

[8] A. Shostack, “Threat modeling: Designing for security”, 1st Edition, Wiley, Indiana, 2014.

[9] G. Fisher; L. Valenta, Monsters in the Middleboxes, accessed 12 September 2021, https://blog.cloudflare.com/monsters-in-the-middleboxes/.

[10] L. O. Nweke and S. D. Wolthusen, “A Review of asset-centric threat modelling approaches” International Journal of Advanced Computer Science and Applications (IJACSA), 11 (2), 2020. **doi**: 10.14569/IJACSA.2020.0110201

[11] A. Al-Azzawi, “Plans for the security analysis of IPv4aaS technologies”, in Proc.14th International Symposium on Applied Informatics and Related Areas, University of Obuda, Székesfehérvár, Hungary, 2019, pp. 101–105.

[12] M. S. Ferdous, F. Chowdhury, J. C. Acharjee, “An extended algorithm to enhance the performance of the current NAPT”, in Proc. International Conference on Information and Communication Technology, Dhaka, Bangladesh, 2007, pp. 315–318, **doi**: 10.1109/ICICT.2007.375401.

[13] A. D. Keromytis, “Buffer overflow attacks”, in H. C. A. van Tilborg, S. Jajodia (Eds.), Encyclopedia of Cryptography and Security, Springer, Boston, 2011, pp. 174–177.

[14] Hong, H., Choi, H., Kim, D., Kim, H., Hong, B., Noh, J., & Kim, Y. (2017), “When cellular networks met IPv6: Security problems of middleboxes in IPv6 cellular networks”, IEEE European Symposium on Security and Privacy (EuroS&P), 2017, **doi**: 10.1109/eurosp.2017.34.

[15] Itu.int. (2016). “IPv6 Security Mechanisms”, [online] Available: https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/Documents/s11-ipv6-securingtransitionmechanisms.pdf [Accessed 5 Feb. 2020].

[16] Georgescu, M., Hazeyama, H., Kadobayashi, Y. and Yamaguchi, S., 2014, May. Empirical analysis of IPv6 transition technologies using the IPv6 Network Evaluation Testbed. In International Conference on Testbeds and Research Infrastructures (pp. 216-228). Springer, Cham.

[17] J. Palet Martinez, H. M.-H. Liu, M. Kawashima, “Requirements for IPv6 Customer Edge Routers”, IETF RFC 8585, 2019, **doi**: 10.17487/RFC8585

[18] G. Lencse, J. Palet Martinez,, L. Howard, R. Patterson, I. Farrer, “Pros and Cons of IPv6 Transition Technologies for IPv4aaS”, IETF v6ops Internet Draft, work in progress, 2021, [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-transition-comparison

[19] A. Pekár, J. Mocnej, W. K. G. Seah, I. Zolotova, “Application Domain-Based Overview of IoT Network Traffic Characteristics”, ACM Computing Surveys, vol 53. no. 4. art. no. 87, **doi**: 10.1145/3399669

[20] I. Drotár, B. Lukács, M. Kuczmann, “LoRaWAN Network Performance Test”, Acta Technica Jaurinensis, vol. 13, no 4, pp. 268–280. **doi**: 10.14513/actatechjaur.v13.n4.547

[21] A. M. Nagy, V . Simon, “Traffic congestion propagation identification method in smart cities” Infocommunications Journal, vol. 13, no 1, March 2021, pp. 45-57. **doi**: 10.36244/ICJ.2021.1.6

Ameen Al-Azzawi received his MSc in Communication Engineering from Northumbria University, Newcastle, England in 2014.

He is now a PhD student at the Budapest university of Technology and Economics, Budapest, Hungary. He has been working full time on his research at MediaNets Laboratory in the Department of Networked Systems and Services since September 2019. His research focus is on IPv6 transition technologies and their security analysis.

Gábor Lencse received his MSc and PhD in computer science from the Budapest University of Technology and Economics, Budapest, Hungary in 1994 and 2001, respectively.

He has been working full time for the Department of Telecommunications, Széchenyi István University, Győr, Hungary since 1997. Now, he is a Networked Systems and Services, Budapest University of Technology and Economics, Budapest, Hungary as a Senior Research Fellow since 2005. The main area of his research is the performance and security analysis of IPv6 transition technologies.

Traffic Swarm Behaviour: Machine Learning and Game Theory in Behaviour Analysis

Gergely Hollósi, Csaba Lukovszki, Máté Bancsics, and Gábor Magyar

***Abstract*—High density traffic on highways and city streets consists of endless interactions among participants. These interactions and the corresponding behaviours have great impact not only on the throughput of traffic but also on safety, comfort and economy. Because of this, there is a great interest in deeper understanding of these interactions and concluding the impacts on traffic participants. This paper explores and maps the world of traffic behaviour analysis, especially researches focusing on groups of vehicles called traffic swarm, while presents the state-of-the-art methods and algorithms. The conclusion of this paper states that there are special areas of traffic behaviour analysis which have great research potential in the near future to describe traffic behaviour in more detail than present methods.**

***Index Terms*—traffic swarm, traffic behaviour, behaviour analysis, game theory, machine learning, deep learning**

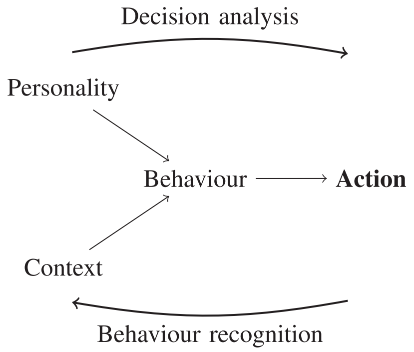


Fig. 1.: The relation of personality, behaviour and action. A specific personality produces an actual behaviour in a peculiar context. The *observable* actions are derivable from the behaviour.

I. INTRODUCTION

THE use of transport vehicles has become a part of everyday life for most people. Consequently, in recent decades, the volume and nature of traffic has changed dramatically. It is clear that traffic has become increasingly congested these days, especially on highways and also on the streets of crowded cities. Various vehicles, such as cars, trucks, cyclists, pedestrians, and novel vehicles are competing for the resources of public roads.

In this changing environment, traditional solutions can provide limited results. It is not necessarily economical to increase the amount of resources by building new routes or expanding roads by new lanes. Against this approach, it is necessary to study the more economic exploitation of the existing road resources. As the complexity of transport increases, so does the demand for safer transportation. To ensure safety on roads, a much deeper understanding of traffic situations must be achieved. Soon, with the spreading of higher level automation (or full self-driving capabilities) of vehicles, the predictability of mixed-traffic interactions is necessary to provide safe and efficient algorithms. Besides, an innovative and state-of-the-art solution must also meet the needs of comfortably and predictability requirements.

From the perspective of this paper, interactions between vehicles result in actions taken by vehicle drivers. Actions are controlled by behaviour which in turn is influenced by the driver's personality (see Fig. 1). Behaviour is also highly dependent on the context. The same person with a specific personality can behave differently in the same situation but in a different context. Basically, behaviour and personality are not observable, so we need to find ways to infer them from actions in specific contexts. For this, two main principles can be identified while analysing traffic behaviour (see Fig. 1). From the point of personality, the behaviour of the driver can be simulated or modelled in common situations; then the behaviour itself or the actions and its consequences can be analysed. This aspect is called *decision analysis*. In terms of actions, the actual behaviour can be deduced by various techniques, which are called *behaviour recognition* or identification.

The main goal of this paper is to review and survey research concerning vehicle driver behaviour, especially the ones that describe the behaviour of a vehicle groups called as traffic swarm in this paper. First, in the next section, traffic swarm concept is presented to define the term *traffic swarm behaviour*. The sections after present the state-of-the-art methods of behaviour identification and decision analysis in the literature. The taxonomy of the discussed methods is presented on Figure 2. Finally, the topic is concluded by some important observation and recommendation about future work.

II. TRAFFIC SWARM

Social swarms, such as ant colonies, birds, bee hives, flocks, herds, shape group's behaviour through simple decisions of individuals and interactions among them. In recent decades, the research community put focus on researching social swarms in order to understand their collaborative behaviour [1], [2], [3], [4], [5]. As a result of continuous evolution, these swarms have developed intelligent behaviours for survival. The basis of emerging behaviour is formed from the behaviour of individuals and from the communication that takes place among them. The method of communication is different for each species, e.g ants use pheromones to mark their path

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, email: {hollosi.gergely, lukovszki.csaba, bancsics.mate, magyar.gabor}@vik.bme.hu

DOI: 10.36244/ICJ.2021.4.3

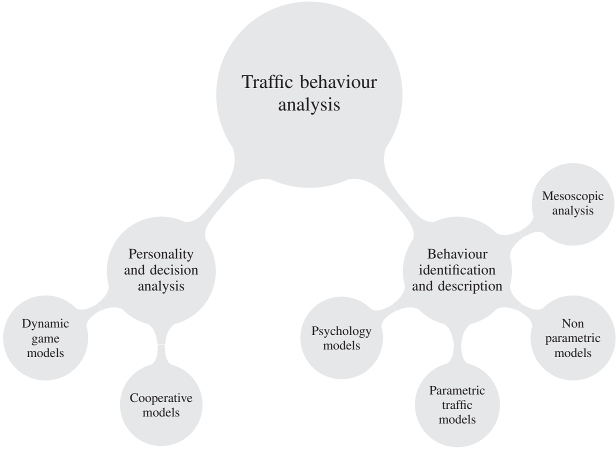


Fig. 2: Overview of the applicable methods of traffic behaviour analysis. The description of the groups and subgroups can be found in the text.

and alert others [2]. This type of communication was first introduced as the concept of stigmergy by Grassé [6]. It was an important step towards understanding the mechanisms of emergence, regulation, and control of collective activities in social insects [6], [7].

On public roads, public transport is also based on collaborative group behaviour. In traditional behaviour modelling, one models an approaching car's breaking as a single event, but in reality this action results in other actions done by the nearby drivers who react to the approaching car. In fact, even the drivers' simplest actions can be described as actions and reactions to each other. This interaction, as stigmergy, relies not only on the distance and the speed of the neighbouring cars, but also affected by the car's movements within the lane, or the observable personal reactions of the driver. Also the resulting action induces reactions of others. This collaborative behaviour is consistent with the behavioural patterns of social swarms.

III. BEHAVIOUR IDENTIFICATION AND DESCRIPTION

A. *Psychology based models*

The most common and classic way of analysing traffic behaviour is based on psychological behaviour models originally developed in psychological research. These models usually *a priori* determine a couple of behaviour categories by which the drivers can be characterised and classified. The categories are well known to the human perception, e.g. aggressiveness or cautiousness. Such categories result in a multi-axis (multi-dimension) space of behaviour patterns, where the dimensions can be described by continuous values, or – typically – by discrete values. For example, [8] uses a 7-point scale to each of the six behaviour categories to express the weight of that behaviour characteristic.

One of the early but exciting development of the cooperation between traffic theory and psychology is the Hierarchical Mental Model (HMM) of the driver [9], which states that the driver behaviour is composed of decision-makers communicating with mental models (see Fig. 3). It divides decision-making into the strategy selection level (e.g. reaching the destination, required travel time, etc.), the tactical level (which selects the next manoeuvre or task to perform), and the operational level (common tasks like car following or lane keeping).

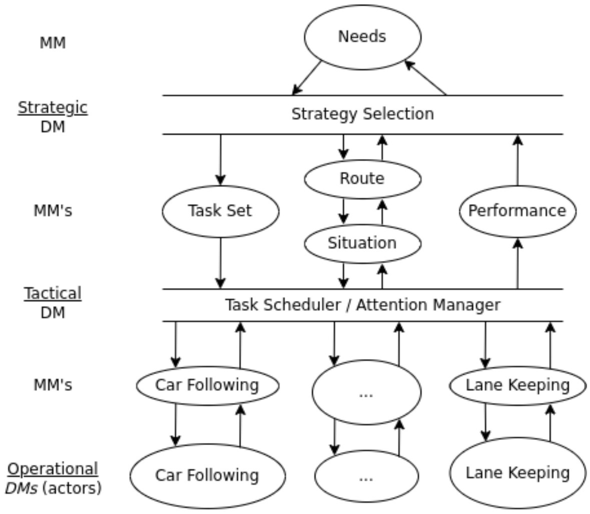


Fig. 3: The Hierarchical Driver Model (HDM). A vehicle driver has different levels of mental models and decision makers. The driver can be described as a couple of mental model running parallel, where each mental model runs at different level. HDM indicates that behaviour is determined not only by tactical decisions but by higher-level strategical decisions. For details, refer to [9].

The main point is that the observed behaviour can originate from either the tactical (e.g. emergency situation) or strategy selection (e.g. being in a hurry) level. However, in practical scenarios, identifying the mental source is not necessarily, one can readily observe and analyse the behaviour itself.

The usual way of analysing traffic behaviour in psychological models are as follows: first, one defines the behaviour dimensions or latent space axes, then identifies the features that can be observed, maps the features into behavioural latent space and finally, analyses the latent space with various methods or provides future predictions to unseen data.

The most important and also the most studied behavioural characteristic is the degree of aggression which is believed to principally determine the safety of driving [8], [10], [11]. E.g. in [12], aggressiveness is the only dimension studied, and a fuzzy logic method is applied to the observed acceleration to predict the level of aggressiveness on a four-level scale. However, not only aggressiveness can be considered in safety, but the carefulness of the driver is also an important category [13]. Attentive and distracted categories were used by [14], where a couple of observed features (driver pose and environmental factors) were used to predict the near-future trajectory of the car by using Convex Markov Chains. The features were processed in a non-supervised manner using k-means algorithm to identify the modes labelled with predefined categories. The DriveSafe application [15] also considers attentiveness and distraction behaviours.

A multidimensional behavioural latent space is defined in [16], [8]. They selected six behaviour dimensions (e.g. aggressiveness, carefulness, etc.) and nine different features; then they conducted surveys to map real-world data onto the six behaviour axes on a 7-point scale. On the latent space, principal component analysis was applied, resulting in the conclusion that aggressiveness and carefulness are the main factors behind all the characteristics.

It is worth mentioning that even classic traffic simulation models use psychological categories implicitly or explicitly. For example, the well-known MOBIL [17] lane-changing model uses the politeness factor when making decisions. Car-following models also have hyperparameters that can be described using psychological correspondences. However, the human aspects of the traffic simulation models remain to be researched extensively [18].

B. Parametric traffic models

The most common methods for analysing microscopic traffic behaviour are based on various parametric models. These representations determine physical, measurable parameters which influence the drivers' behaviour and decision-making process in certain traffic situations. The most common models were developed to describe behaviour in highway traffic, e.g. changing the car's acceleration based on the speed and distance of the vehicle ahead (car-following models, [19], [20], [21]), or the event of lane changing based on the relative speed and available space in the overtaking lane (lane-changing models, [22]). However, more complex models are being developed to describe more complicated events, such as driver behaviour during unexpected lane closures [23].

Almost all parametric models require the specification of some hyperparameters which describe certain driver properties. For example, in the case of the most well-known car-following model, namely the Intelligent Driver Model (IDM) [21], a couple of hyperparameters are needed to be defined as inputs of simulations, like desired velocity in free driving, maximum acceleration or comfortable braking deceleration.

The correct setting of these hyperparameters is critical for making the simulation models realistic. This can be done using real traffic data during model calibration, when the input hyperparameters are adjusted in a way that each car's behaviour and trajectory mirrors the real-life data as closely as possible. This optimisation process can be done using analytic calculations [24]; however, newer methods also use artificial intelligence and neural networks [25], [26].

As the IDM example shows, the hyperparameters that describe a chosen vehicle's motion dynamics are constants and can be used to determine the response to different events. However, the value of these hyperparameters also bear useful information and, when they are suited for real-life data, can be used for describing real driver traits [27].

As such, the examination of hyperparameter statistics can be promising. The distribution of these properties can provide useful information about the expectable behaviour of drivers in a given situations, as well as the probability and nature of irregular behaviours [28], which can lead to more accurate predictions [28], [27].

Although these models describe certain driver actions in a realistic and comprehensible way, the understanding of intensive, complex and heterogeneous traffic requires more elaborated methodologies. For example, it may be worth comparing the results of these parametric model analyses with the aforementioned psychological models and finding correlations between the psychological driver traits and the hyperparameters of the trajectories. This can lead to more accurate descriptions of how different types of drivers behave on the road.

C. Non-parametric machine learning models

There is a wide literature on the application of non-parametric models to describe driver behaviour in different traffic scenarios. Non-parametric behaviour analysis uses a black-box model. In this machine learning principle, the model learns the driver behaviour from real input and output data. The behaviour itself is the relation between input feature data and the related output response.

The emergence of ADAS (Advanced Driver Assistance System) has emphasised the need and viability of this kind of modelling. In order to provide relevant information to the driver, a system must be capable of understanding what is currently happening at a certain traffic scene. This problem is known as situation assessment [29]. To address this problem, conditional probabilistic Bayesian networks are used to estimate driver intention in road intersections in [30] and [31].

Besides these approaches, machine learning principles are widely used to build non-parametric behaviour models to describe ego driver behaviour. The model input basically is the ego driver measurable relations to its surroundings, the ambient cars, or even static and dynamic signals, signs and lane rules. The common output is the real trajectory of the ego driver's vehicle.

In a typical approach, inputs come from the eight surrounding vehicles, which concept in [32] is extended by the preceding vehicle of the front vehicle resulting in the most reliable behaviour modelling and prediction. Most of the approaches use a full set of input and output data; however, [33] exploited sparse floating car data to predict vehicle trajectories with penetration rate between 2 and 8 percent.

As in many machine learning applications, existing techniques can be split between classification or regression. When applied to motion prediction, classification problems intend to determine a high-level intention, such as lane changing to left or right, lane-keeping or braking during highway driving, turning left, turning right, or going straight in an intersection.

On the other hand, regression problems aim at directly obtaining a predicted future positions of the considered vehicle, which can then be even used for motion planning of target vehicle. Trajectory prediction hypothesises can be used as an input during decision process. ([34])

More recently, the most used approaches for traffic flow prediction are based on deep learning algorithms. The most used methods for traffic prediction are the Long Short-Time Memory (LSTM) and the Generative Adversarial Network (GAN). Because LSTM neural networks are able to keep previous inputs in the memory, they are considered particularly efficient for problems that require learning of patterns in a time series. In [32], authors described a unimodal vehicle trajectory prediction in highway scenes for ADAS systems proposing a Keras framework based LSTM neural network.

However, predicting one manoeuvre is not the best solution. Yuexin et al. [35] showed while studying pedestrian movements that a person has, couple of choices in a given situation is socially acceptable. This is called multimodal behaviour. Multimodality is highly important: if an algorithm predicts one trajectory only, it can be the average of possible trajectories resulting in a mode collapse. Based on this result, a couple of multimodal solutions were introduced for vehicle manoeuvre prediction, like in [36], [37].

Through intensive analysis [33] proved that, although LSTM-based models are superior in unimodal scenarios, generative models perform best in those where the effects of multimodality are higher. Luca et al. [33] shows that the difference between using an LSTM and a GAN for prediction lies in the objective function. The LSTM learns an average behaviour because it minimizes the average error between all the predictions, while GANs learn to produce plausible samples corresponding to specific behaviours as a result of competition between generator and discriminator networks.

The training phase is essential in machine learning models. Under-represented data usually leads to poor prediction, invalid or unseen situations. This problem of cascading errors [38] is well-known in expert knowledge-based Imitation Learning (IL). Motivated by working on alternative IL methods, [39] proposed Inverse Reinforcement Learning (IRL), implemented with Generative Adversarial Imitation Learning (GAIL) [40]. The model assumes that the expert follows an optimal policy with respect to an unknown reward function. When the reward function is discovered, the model can follow the Reinforcement Learning (RL) method and acts identically as the expert. The proposed imitation even extends the model to unseen scenarios.

In order to estimate the influence of various vehicles near the target vehicle, there is a need to jointly reason and predict the future trajectories of all the vehicles involved in a traffic scene, especially in intersections. In the literature, in most of the scenarios, homogeneous traffic is considered, where although the behaviour can be different, each vehicle or driver follows the same rule. A more demanding situation happens when traffic is heterogeneous. In a heterogeneous scenario, different types of vehicles interact with each other with different behaviour and action sets. This scenario is particularly demanding if the traffic is predominated by smaller vehicles that do not follow lane rules, such as scooters, motorcycles, or bicycles, or dominant vehicles that regulate traffic, such as buses.

To handle heterogeneous traffic, mapping different vehicle types to equivalent homogeneous flow is available in [41], [42]. A more sophisticated solution is proposed in [43]. The authors proposed a conditional GAN model to generate multiple trajectory predictions for vehicles at either signalised or non-signalised intersections.

As a result of building predictive models, latent space is built, which represents the driver behaviour. The explanation of latent space is a fundamental task to understand the learned behaviour. The understanding is based on the traffic scene classification, which traditionally is defined by expert knowledge.

In the literature, different approaches are available for automatic latent space exploration. The goal of [44] was to explain the behaviour of any given black-box classifier instead of just reasoning about its individual predictions. Their explanation relies on the small number of compact decision sets. The authors of [45] introduced human-readable instance-wise feature selection as a methodology for model interpretation. The proposed method learns a function to extract a subset of the most informative features for each given example.

[46] introduced two different unsupervised deep learning approaches to understand and classify real traffic scenes from ADAC ego driver point of view. Authors proved that proposed approaches are capable of learning an expressive latent space for a real-world highway dataset and making scene classification from real datasets.

As a result of explanation, we lose information available in latent space, which is the cost of easier interpretation. Various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related to each other and when one method is preferable over another. To address this problem, the authors of [47] present a unified framework for interpreting predictions. Based on the proposed unification, the authors present new methods that show improved computational performance and better consistency with human intuition than previous approaches. Extending the above work, authors show in [44] that a good explanation must follow properties of fidelity, unambiguity, interpretability, and interactivity.

The predictive models' increasing complexity makes it harder to explain or reason the driver behaviour represented in the latent space [48]. It emphasises the need for tools that can explain predictive models' complex behaviour in a faithful and interpretable manner.

D. Mesoscopic analysis

There are two common, well-known description strategies for traffic situations: microscopic and macroscopic descriptions. While the latter assigns aggregated measures to the traffic flow, the former describes each vehicle's behaviour individually. However, microscopic description is egoistic, meaning that it views traffic from the point of one vehicle at a time. The interaction between agents (vehicles) is usually simplified to the followed car, and a couple of neighbours defined *a priori*. The behaviour of the vehicle does not depend directly on the manoeuvres of nearby cars, e.g. the lane change in MOBIL model depends only on the car ahead and the space in the target lane [17]. However, the dependency of the participants is much wider than in the classic microscopic models; in fact, there are a number of vehicles affected by each other mutually. So, we can handle a group of vehicles as a mesoscopic model of traffic situations, where a couple of nearby participants interact with each other continuously. This way, we can analyse the behaviour of this group together, highlighting a deeper relationship between the behaviour of distinct vehicles.

A couple of early works analysed the nature of vehicle groups. In [49] the author found a type of collective behaviour (called „solid block") in the case of increasing density of vehicles resulting in a highly coherent state, in which all the vehicles have the same average velocity with little deviation. Besides, [50] showed the closed stability conditions of vehicle formations. In contrast, [51] searched for the answer to the question: what characterises a „free vehicle”? Based on a large number of recorded vehicle movements, she found that time separation is a better measure of freedom than distance-based metrics. There were attempts to discover the conditions of forming and splitting vehicles into groups. In [52], a Bayesian estimator estimated the state of individual vehicles then a density-based clustering approach was applied to identify vehicle groups and group boundaries. They found that the probability of collision is an applicable metric for the closeness of two vehicles.

The latest and most common research in mesoscopic analysis also tries to solve the task of multimodal trajectory and manoeuvre prediction, meaning that not only one trajectory but a couple of possible trajectories are predicted. The topic is extensively researched in automated driving because the behaviour prediction and possible movements of nearby vehicles is a serious safety issue.

After realising that independent traffic models are not successful enough in trajectory and manoeuvre prediction [30], [31], a new theory called „social force” was taken over from the analysis of pedestrian behaviour. [53] introduced the term „social force” measuring the internal motivations of the individuals to perform certain actions. The social force works like a force in the force field, and is actually a vector, which attracts and distracts: attracts to the desired velocity of motion and keeps a certain distance to other pedestrians. The social force model was improved in order to „learn” the social force from real trajectories which leaded to a so called „social pooling” layer in LSTM networks [54]. Based on these results, [55] introduced the convolutional social pooling term to consider the social context during trajectory prediction. The social pooling basically learns the spatial interdependencies of the tracks based on the output of LSTM encoders, which we call manoeuvre prediction. The same method was applied in [35], using different participant classes like pedestrians, cars and bicycles. It is worth noting that the behaviour prediction model can be static or dynamic: some works predict fixed manoeuvre classes, others predict graph embeddings or other encodings.

The latest research involves state-of-the-art graph neural networks representing traffic situation as an interaction graph. Interestingly enough, these solutions usually employ the same method: they use a behaviour or manoeuvre prediction model and a trajectory prediction model (see e.g. Fig. 4).

For example, [57], [58] applies a graph convolutional model to extract the dynamic graph features from a graph formed by nearby cars. An LSTM network is then used to generate the trajectory predictions. In [59], the authors use same method, but estimate all the movements of every considered car, and use interaction model based on collision estimates. There are a couple of other methods, using attentive graph networks [60], [61], or joint estimations [56]. Not just vehicle-only interactions, but vehicle-pedestrian interactions were analysed in [62].

These results aim at the topic of trajectory prediction. However, the methods can be made suitable to realise traffic behaviour analysis. Unfortunately, this topic has been rarely studied, although reasoning in machine learning systems is well-known and widely practiced.

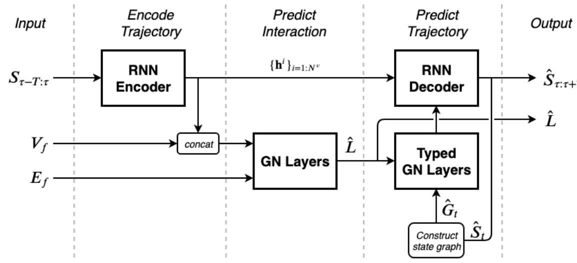


Fig. 4: A typical graph based on trajectory prediction method [56]. These models consist of an interaction prediction part creating features to describe interaction between participants and a trajectory prediction part estimating the trajectories of the participants based on the interaction features. The interac- tion prediction part on this model is readily identifiable which potentially provides internal features to analyse behaviour.

The analysis of the characteristics of traffic behaviour requests the interpretation of the presented machine learning models. A couple of proposed methods can be found in the literature. The main goal is to interpret the behaviour prediction model, which is a graph neural network. Analysing the graph helps to discover the reasons for a selected manoeuvre as a dependence of the neighbouring vehicles. To understand which parts of the graph are contributing to a prediction, [63] proposed an edge classifier that predicts if an edge can be dropped. This way, the most influential interacting parties can be found effectively. Similarly, [64] finds a smaller subgraph and a subset of features that have a crucial role in predictions.

IV. PERSONALITY AND DECISION ANALYSIS

While identifying and predicting behaviour from observations are widely studied topics, analysing the effect of different personalities and drivers’ decisions on traffic is also promising. Personality and decision analysis deals with the interaction of vehicle drivers trying to solve traffic situations mutually. Practical methods apply traffic simulations to resolve the issues of a selected situation; however, they sometimes lack behaviour or personality aspects. Fortunately, we can find game theory models developed to analyse these aspects and select the proper behaviour or action to solve the traffic situation effectively.

The behaviour of the interacting drivers evolves from the personality and the actual situation. This way, the methods typically examined define some traffic situation (e.g. a lane change or a road crossing) and a couple of personality or behaviour patterns. The traffic interactions have a couple of safety, sociological, and communication perspectives. The most complete definition and taxonomy of traffic interaction can be found in [65], which states that traffic interaction is a „situation where the behaviour of at least two road users can be interpreted as being influenced by the possibility that they are both intending to occupy the same region of space at the same time in the near future”. These interactions include not only interactions between vehicle drivers but also, for example, between a vehicle driver and a pedestrian.

Game-theoretic models are widely used in traffic theory, not only in behaviour analysis. For instance, several methods can be found for planning the traffic infrastructure in urban spaces [66] or to optimise traffic throughput [67], [68], [69]. However, the first mention of the game theory applied to traffic behaviour analysis dates back to 1984, when Fisk applied Nash non-cooperative and Stackelberg games in transportation problems like intercity passenger travel optimisation and traffic signal optimisation [70]. A huge family of game-theoretic traffic behavioural models analyses the traffic behaviour in the aspect of cooperation. Cooperation seems to contradict evolution and natural selection; however, Nowak showed that evolution could actually lead to cooperation [71]. The main point in these kinds of analysis is that a group of players (called cooperators) take joint actions resulting in collective payoffs, while defectors do not attach to these groups. The methods are commonly based on the cellular automata paradigm. It is worth to point out that the validation of game theory models is not always provided, i.e. the game theory model may not describe an observable behaviour in reality.

Modelling freeway traffic, [72] investigated the evolution of cooperative and defective behaviour in lane change situations. They showed that as the difference between the velocity of co-operators and defectors decreases, the fraction of cooperators increases, i.e. in high-density traffic the drivers start to behave as cooperators. At a non-signalised crossing, it was shown that cooperative drivers improve traffic quality, while defective drivers worsen it [73]. It was also presented that there is some relationship between the density of cars on the road and the desire for drivers to go defective. However, defectiveness is not always a bad thing: using Weibull distribution for describing the generation of defectors, it can be shown that while the existence of defectors reduces the safety of the crossing, it is also beneficial for the capacity of the side road [74]. An interesting defective-cooperative model was developed in [75], where the non-ego vehicles switch between cooperative and defective behaviour in the process of interaction. The main idea is that while the non-ego vehicles' core behaviour is cooperative, they have a short duration of defective behaviour (so-called adversarial time), making the ego vehicle behave carefully, e.g. the driver becomes attentive when sees a pedestrian at the edge of the road. Other interesting methods in this topic can be found in [76], [77], [78], [79].

Of course, there are game-theoretic models beyond cooperative games. For example, [80] presents the Game-Theoretic Social Force Model based on the well-known social model of pedestrians or [81] models 2-to-1 lane junctions as an N prisoner's dilemma game. The [82] review cites 10 game-theoretic models showing why informal norms of behaviour develop, e.g. giving way to cyclists who do not have priority. There is also some criticism regarding the application of game-theoretic models, e.g. that the Nash equilibrium is not always properly applicable in traffic behaviour [83].

While it is a benefit for a human driver to identify the behaviour of other participants, nowadays, even autonomous vehicles (AV) need to be able to predict the behaviour of human drivers. In the near future, mixed traffic situations are going to be widespread, so action planning in AVs requires behaviour prediction to be executed for safety and efficiency. The main framework of the behaviour analysis of AV game theory models is the dynamic game model, which essentially is a dynamic model of the steps of the game. The model's state is typically composed of the position, the velocity, and the vehicle's heading [84]. These models' most notable decision policy is the so-called level-k reasoning, where all level-k players make strategic decisions by assuming that all of the other players are level-$(k-1)$. A level-0 player makes no strategic decision, e.g. she reflexively brakes if the followed vehicle is breaking. These models commonly handle a colossal state space and make complex optimisations to produce decisions. Level-k models were successfully applied to freeway situations [85] and also to uncontrolled intersections [86]. However, [87] showed on two large naturalistic datasets that even level-0 behaviour can capture most human driving behaviour. Finally, it is worth mentioning that to help behaviour planning research, [88] created a taxonomy of human interactions in traffic conflicts, including vehicle-vehicle and vehicle-pedestrian conflicts.

V. CONCLUSION

The current paper outlined different methods for driver behaviour analysis, even in traffic situations with complex interactions (see Table I.). However, classical behaviour analysis – which finds reasons and motivations behind actions – is committed by only psychological models while most research topics target trajectory and interaction prediction. While state-of-the-art machine learning models, latent space exploration techniques and game theory methods are successfully applied to numerous problems (e.g. image processing), it is hard to find a solution which addresses entirely the traffic behaviour analysis. It is tempting to improve the presented methods to be applicable for exploring the behaviour behind visible actions, moreover, to reason the behaviour of groups of vehicles.

The consideration of traffic swarms is really promising, where the interactions among drivers are not limited to the surrounding vehicles and are not bound to kinematic properties, so one can analyse the actions as a result of complex behaviour instead of only trajectories or visible interactions of the vehicles. However, mesoscopic modelling provides a reasonable starting point to this research. Beyond that, further research on behaviour interpretation is a promising direction, but modelling, analysing, and extracting the latent space requires more elaborated measures for driver behaviour evaluation.

ACKNOWLEDGMENT

The research was supported by the Institute of Transport Sciences (KTI) within the Innovative Mobility Program.

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program.

TABLE I
THE SUMMARY OF METHODS FOR TRAFFIC BEHAVIOUR ANALYSIS. THE „METHODOLOGIES” COLUMN SHOWS A COUPLE OF KEYWORDS (THEORIES, TECHNOLOGIES, METHODS, ETC.) WHICH MOSTLY, BUT NOT EXCLUSIVELY, DESCRIBE THE METHOD.

Method		Description	Methodologies	References
Personality and decision analysis	Dynamic game models	The models define the states of the game and the state transition functions and analyse the dynamic progression of the states.	dynamic game theory models, level-k model	[84] [85] [86] [87] [88]
	Cooperative models	Cooperative models find the consequences of some discrete behaviours in actual traffic situations using game theory methods.	game theory, cooperative and non-cooperative games (zero-sum games, Stackelberg games)	[65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83]
Behaviour identification and description	Psychology models	Psychology models determine a priori behaviour categories and maps visible actions to these categories.	psychology categories, surveys, latent space analysis (e.g. PCA)	[8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18]
	Parametric traffic models	Parametric traffic models are simulation models used by traffic engineers. The latent space of hyperparameters can be analysed as a behaviour space.	traffic simulation models (e.g. Wiedemann, IDM), latent (hyperparameter) space analysis	[19] [20] [21] [22] [23] [89] [24] [25] [26] [28] [27]
	Non parametric models	Non parametric models are machine learning models which are mapping visible actions to latent behaviour spaces. The methodologies are typically based on generative deep learning methods or non-supervised learning methods.	trajectory prediction, machine learning models, deep learning models (GAN, LSTM), reinforcement learning, model explanation	[29] [30] [31] [32] [33] [34] [35] [38] [39] [40] [43] [44] [45] [46] [47] [48]
	Mesoscopic models	Mesoscopic models are between microscopic and macroscopic models. They are commonly using graph based solutions to analyse complex interactions among participants.	interaction prediction, social force models, social pooling, graph neural networks, model explanation	[17] [49] [50] [51] [52] [30] [31] [54] [55] [35] [53] [57] [58] [59] [60] [61] [56] [62] [63] [64] [44] [90] [45]

REFERENCES

[1] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine, “Self-organization in social insects,” *Trends in Ecology and Evolution*, vol. 12, p. 188, 05 1997, DOI: 10.1016/S0169-5347(97)01048-3.

[2] P. Karlson and M. Lüscher, “Pheromones: a new term for a class of biologically active substances,” *Nature*, vol. 183, pp. 55–56, 01 1959, DOI: 10.1038/183055a0.

[3] G. Theraulaz, E. Bonabeau, S. C. Nicolis, R. V. Solé, V. Fourcassié, S. Blanco, R. Fournier, J.-L. Joly, P. Fernández, A. Grimal, P. Dalle, and J.-L. Deneubourg, “Spatial patterns in ant colonies,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 15, pp. 9645–9649, 2002. [Online]. Available: https://www.pnas.org/content/99/15/9645. DOI: 10.1073/pnas.152302199

[4] K. von Frisch, “The dance language and orientation of bees,” 1967,” DOI: 10.4159/harvard.9780674418776.

[5] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’87. New York, NY, USA: Association for Computing Machinery, 1987, p. 25–34, DOI: 10.1145/37401.37406.

[6] P.-P. Grasse, “La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. la theorie de la stigmergie: Essai d’interpretation du comportement des termites constructeurs.” *Insectes Sociaux*, vol. 6, pp. 41–81, 1959.

[7] G. Theraulaz and E. Bonabeau, “A brief history of stigmergy,” *Artificial life*, vol. 5, pp. 97–116, 02 1999, DOI: 10.1162/106454699568700.

[8] E. Cheung, A. Bera, E. Kubin, K. Gray, and D. Manocha, “Identifying driver behaviors using trajectory features for vehicle navigation,” 2018,” DOI: 10.1109/IROS.2018.8594348.

[9] E. R. Boer and D. M. Hoedemaeker, “Modeling driver behaviour with different degrees of automation: a hierarchical decision framework of interacting mental models,” 1998.

[10] P. Harris, J. Houston, J. Vazquez, J. Smither, A. Harms, J. Dahlke, and D. Sachau, “The prosocial and aggressive driving inventory (padi): A self-report measure of safe and unsafe driving behaviors,” *Accident Analysis and Prevention*, vol. 72, p. 1–8, 11 2014, DOI: 10.1016/j.aap.2014.05.023.

[11] F. Zhong-xiang, L. Yang-yang, L. Jing, and Z. Wei-hua, “Selectedmodel and sensitivity analysis of aggressive driving behavior,” *China Journal of Highway and Transport*, vol. 25, no. 2, pp. 106–112, 2012.

[12] A. Aljaafreh, N. Alshabatat, and M. S. Najim Al-Din, “Driving style recognition using fuzzy logic,” in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, 2012, pp. 460–463, DOI: 10.1109/ICVES.2012.6294318.

[13] R. Vasudevan, V. Shia, Y. Gao, R. Cervera-Navarro, R. Bajcsy, and F. Borrelli, “Safe semi-autonomous control with enhanced driver modeling,” in *2012 American Control Conference (ACC)*, 2012, pp. 2896–2903, DOI: 10.1109/ACC.2012.6315654.

[14] D. Sadigh, K. Driggs-Campbell, A. Puggelli, W. Li, V. Shia, R. Bajcsy, A. Vincentelli, S. Sastry, and S. Seshia, “Data-driven probabilistic modeling and verification of human driver behavior,” 03 2014.

[15] L. M. Bergasa, D. Almería, J. Almazán, J. J. Yebes, and R. Arroyo, “Drivesafe: An app for alerting inattentive drivers and scoring driving behaviors,” in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 240–245, DOI: 10.1109/IVS.2014.6856461.

[16] A. Bera and D. Manocha, “Behavior modeling for autonomous driving,” in *Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy*, 2018, pp. 16–21.

[17] A. Kesting, M. Treiber, and D. Helbing, “General lane-changing model mobil for car-following models,” *Transportation Research Record*, vol. 1999, pp. 86–94, 01 2007, DOI: 10.3141/1999-10.

[18] M. Saifuzzaman and Z. Zheng, “Incorporating human-factors in car-following models: A review of recent developments and research needs,” *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 379–403, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X14002551. DOI: 10.1016/j.trc.2014.09.008

[19] R. Wiedemann, “Simulation des strassenverkehrsflusses.” 1974.

[20] P. Gipps, “A behavioural car-following model for computer simulation,” *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0191261581900370. DOI: 10.1016/0191-2615(81)90037-0

[21] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Phys. Rev. E*, vol. 62, pp. 1805–1824, Aug 2000, DOI: 10.1103/PhysRevE.62.1805.

[22] M. Brackstone, M. McDonald, and J. Wu, “Lane changing on the motorway: factors affecting its occurrence, and their implications,” *IET Conference Proceedings*, pp. 160–164(4), January 1998, DOI: 10.1049/cp:19980175.

[23] B. Mahmood and J. Kianfar, "Driver behavior models for heavy vehicles and passenger cars at a work zone," *Sustainability*, vol. 11, no. 21, 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/21/6007. **doi**: 10.3390/su11216007

[24] C. Chen, L. Li, J. Hu, and C. Geng, "Calibration of mitsim and idm car-following model based on ngsim trajectory datasets," in *Proceedings of 2010 IEEE International Conference on Vehicular Electronics and Safety*, 2010, pp. 48–53, **doi**: 10.1109/ICVES.2010.5550943.

[25] I. I. Otković, T. Tollazzi, and M. Šraml, "Calibration of microsimulation traffic model using neural network approach," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5965–5974, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0957417413002893. **doi**: 10.1016/j.eswa.2013.05.003

[26] P. Fan, J. Guo, H. Zhao, J. S. Wijnands, and Y. Wang, "Car-following modeling incorporating driving memory based on autoencoder and long short-term memory neural networks," *Sustainability*, vol. 11, no. 23, 2019. [Online]. Available: https://www.mdpi.com/2071-1050/11/23/6755. **doi**: 10.3390/su11236755

[27] B. Hegde, M. O'Keefe, S. Muldoon, J. Gonder, and C.-F. Chang, "Real-world driving features for identifying intelligent driver model parameters," 4 2021. [Online]. Available: https://www.osti.gov/biblio/1779054. **doi**: 10.4271/2021-01-0436

[28] M. Pourabdollah, E. Bjärkvik, F. Fürer, B. Lindenberg, and K. Burgdorf, "Calibration and evaluation of car following models using real-world driving data," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6, **doi**: 10.1109/ITSC.2017.8317836.

[29] M. Endsley, "Theoretical underpinnings of situation awareness: A critical review," *Situation awareness analysis and measurement*, pp. 3–32, Jan 2000, **doi**: 10.1201/B12461-7.

[30] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Exploiting map information for driver intention estimation at road intersections," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 583–588, **doi**: 10.1109/IVS.2011.5940452.

[31] J. Firl, H. Stu ¨bing, S. A. Huss, and C. Stiller, "Predictive ma- neuver evaluation for enhancement of car-to-x mobility data," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 558–564, **doi**: 10.1109/IVS.2012.6232217.

[32] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Oct 2017. **doi**: 10.1109/itsc.2017.8317913.

[33] L. Rossi, A. Ajmar, M. Paolanti, and R. Pierdicca, "Vehicle trajectory prediction and generation using lstm models and gans," *PLOS ONE*, vol. 16, no. 7, pp. 1–28, Jul 2021, **doi**: 10.1371/journal.pone.0253868.

[34] D. Dolgov, "Google self-driving car project monthly report," Google Inc., Tech Report, Feb 2016.

[35] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," 2019," arXiv:1811.02146.

[36] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun 2018, **doi**: 10.1109/ivs.2018.8500493.

[37] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," 2019," **doi**: 10.1109/ICRA.2019.8793868.

[38] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *13th International Conference on Artificial Intelligence and Statistics*, ser. Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9, no. 5. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 661–668.

[39] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Intelligent Vehicles Symposium (IV)*. IEEE, Jun 2017. **doi**: 10.1109/ivs.2017.7995721.

[40] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *30th International Conference on Neural Information Processing Systems (NIPS)*. ACM, Oct 2016, pp. 4572–4580, arXiv: 1606.03476.

[41] X. Liang, Z. Liu, and K. Qian, "Capacity analysis of signalized intersections under mixed traffic conditions," *Journal of Transportation Systems Engineering and Information Technology*, vol. 11, no. 2, pp. 91–99, Apr 2011, **doi**: 10.1016/S1570-6672(10)60116-X.

[42] N. Cao and K. Sano, "Estimating capacity and motorcycle equivalent units on urban roads in hanoi, vietnam," *Journal of Transportation Engineering*, vol. 138, no. 6, pp. 776–785, 2012, **doi**: 10.1061/(ASCE)TE.1943-5436.0000382.

[43] D. Roy, T. Ishizaka, C. K. Mohan, and A. Fukuda, "Vehicle trajectory prediction at intersections using interaction based generative adversarial networks," in *Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 2318–2323, **doi**: 10.1109/ITSC.2019.8916927.

[44] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec, "Interpretable & explorable approximations of black box models," 2017," arXiv:1707.01154.

[45] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," 2018," arXiv:1802.07814.

[46] N. Harmening, M. Bilosˇ, and S. Günnemann, "Deep representation learning and clustering of traffic scenarios," 2020," arXiv:2007.07740.

[47] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *31st International Conference on Neural Information Processing Systems*. ACM, Dec 2017, pp. 4768–4777, arXiv:1705.07874.

[48] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016, pp. 1135–1144, **doi**: 10.1145/2939672.2939778.

[49] D. Helbing and B. Huberman, "Coherent moving states in highway traffic," Nature, vol. 396, pp. 738–740, 12 1998, **doi**: 10.1038/25499.

[50] J. A. Fax and R. M. Murray, "Graph laplacians and stabilization of vehicle formations," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 55–60, 2002, 15th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015385116. **doi**: 10.3182/20020721-6-ES-1901.00090

[51] K. Vogel, "What characterizes a "free vehicle" in an urban area?" *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 5, no. 1, pp. 15–29, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369847802000037. **doi**: 10.1016/S1369-8478(02)00003-7

[52] Q. Wang and B. Ayalew, "A probabilistic framework for tracking the formation and evolution of multi-vehicle groups in public traffic in the presence of observation uncertainties," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 560–571, 2018, **doi**: 10.1109/TITS.2017.2750073.

[53] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, p. 4282–4286, May 1995, **doi**: 10.1103/physreve.51.4282.

[54] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971, **doi**: 10.1109/CVPR.2016.110.

[55] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," 2018," **doi**: 10.1109/CVPRW.2018.00196.

[56] D. Lee, Y. Gu, J. Hoang, and M. Marchetti-Bowick, "Joint interaction and trajectory prediction for autonomous driving using graph neural networks," 2019," arXiv:1912.07882.

[57] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interactionaware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3960–3966, **doi**: 10.1109/ITSC.2019.8917228.

[58] ——, "Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving," 2020," **doi**: 10.13140/RG.2.2.16886.63044.

[59] E. Jo, M. Sunwoo, and M. Lee, "Vehicle trajectory prediction using hierarchical graph neural network for considering interaction among multimodal maneuvers," *Sensors*, vol. 21, no. 16, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/16/5354. **doi**: 10.3390/s21165354

[60] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara, "Dag-net: Double attentive graph neural network for trajectory forecasting," 2020," **doi**: 10.1109/ICPR48806.2021.9412114.8

[61] J. Li, H. Ma, Z. Zhang, and M. Tomizuka, "Social-wagdat: Interaction-aware trajectory prediction via wasserstein graph double-attention network," 2020," **DOI**: 10.13140/RG.2.2.25253.04320.

[62] Z. Li, J. Gong, C. Lu, and Y. Yangtian, "Interactive behavior prediction for heterogeneous traffic participants in the urban road: A graph-neural-network-based multitask learning framework," *IEEE/ASME Transactions on Mechatronics*, vol. PP, p. 1, 03 2021, **DOI**: 10.1109/TMECH.2021.3073736.

[63] M. S. Schlichtkrull, N. D. Cao, and I. Titov, "Interpreting graph neural networks for nlp with differentiable edge masking," 2021," arXiv:2010.00577.

[64] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," 2019," **DOI**: 10.5555/3454287.3455116.

[65] G. Markkula, R. Madigan, D. Nathanael, E. Portouli, Y. M. Lee, A. Dietrich, J. Billington, A. Schieben, and N. Merat, "Defining interactions: a conceptual framework for understanding interactive behaviour in human and automated road traffic," *Theoretical Issues in Ergonomics Science*, vol. 21, no. 6, pp. 728–752, 2020, **DOI**: 10.1080/1463922X.2020.1736686.

[66] S. Sakaue and K. Nakamura, "Differentiable equilibrium computation with decision diagrams for stackelberg models of combinatorial congestion games," 2021," arXiv:2110.01773.

[67] V. Morandi, "Bridging the user equilibrium and the system optimum in static traffic assignment: how the cooperation among drivers can solve the congestion problem in city networks," 2021," arXiv:2105.05804.

[68] H. Heaton, D. McKenzie, Q. Li, S. W. Fung, S. Osher, and W. Yin, "Learn to predict equilibria via fixed point networks," 2021," arXiv:2106.00906.

[69] L. Cianfanelli, G. Como, A. Ozdaglar, and F. Parise, "Optimal intervention in traffic networks," 2021," arXiv:2102.08441.

[70] C. Fisk, "Game theory and transportation systems modelling," *Transportation Research Part B: Methodological*, vol. 18, no. 4, pp. 301– 313, 1984. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/0191261584900134. **DOI**: 10.1016/0191-2615(84)90013-4

[71] M. A. Nowak, "Five rules for the evolution of cooperation, "*Science*, vol. 314, no. 5805, pp. 1560–1563, 2006, **DOI**: 10.1126/science.1133755.

[72] L. E. Cortés-Berrueco, C. Gershenson, and C. R. Stephens, "Traffic games: Modeling freeway traffic with game theory," *PLOS ONE*, vol. 11, no. 11, pp. 1–34, 11 2016, **DOI**: 10.1371/journal.pone.0165381.

[73] K. J. Szajowski and K. Włodarczyk, "Drivers' skills and behavior vs. traffic at intersections," *Mathematics*, vol. 8, no. 3, p. 433, Mar 2020, **DOI**: 10.3390/math8030433.

[74] H. Fan, B. Jia, J. Tian, and L. Yun, "Characteristics of traffic flow at a non-signalized intersection in the framework of game theory," *Physica A: Statistical Mechanics and its Applications*, vol. 415, pp. 172– 180, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437114006001. **DOI**: 10.1016/j.physa.2014.07.031

[75] C.-Y. Chiu, D. Fridovich-Keil, and C. J. Tomlin, "Encoding defensive driving as a dynamic nash game," 2021," arXiv:2011.04815.

[76] L. Sun, M. Cai, W. Zhan, and M. Tomizuka, "A game-theoretic strategy-aware interaction algorithm with validation on real traffic data," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11038–11044, **DOI**: 10.1109/IROS45743.2020.9340988.

[77] P. Geiger and C.-N. Straehle, "Learning game-theoretic models of multiagent trajectories using implicit layers," 2021," arXiv:2008.07303.

[78] X. Liao, X. Zhao, G. Wu, M. Barth, Z. Wang, K. Han, and P. Tiwari, "A game theory based ramp merging strategy for connected and automated vehicles in the mixed traffic: A unity-sumo integrated platform," 2021," arXiv:2101.11237.

[79] L. Wastavino, B. Toledo, J. Rogan, R. Zarama, V. Muñoz, and J. Valdivia, "Modeling traffic on crossroads," *Physica A: Statistical Mechanics and its Applications*, vol. 381, pp. 411–419, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0378437107003330. **DOI**: 10.1016/j.physa.2007.03.052

[80] F. T. Johora, D. Yang, J. P. Müller, and Ümit Özgüner, "On the generalizability of motion models for road users in heterogeneous shared traffic spaces," 2021," arXiv:2101.06974.

[81] A. Yamauchi, J. Tanimoto, A. Hagishima, and H. Sagara, "Dilemma game structure observed in traffic flow at a 2-to-1 lane junction," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 79, p. 036104, 04 2009, **DOI**: 10.1103/PhysRevE.79.036104.

[82] R. Elvik, "A review of game-theoretic models of road user behaviour," *Accident Analysis & Prevention*, vol. 62, pp. 388–396, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0001457513002443. **DOI**: 10.1016/j.aap.2013.06.016

[83] J. Wright and K. Leyton-Brown, "Beyond equilibrium: Predicting human behavior in normal-form games," vol. 2, 07 2010, **DOI**: 10.1145/1807406.1807449.

[84] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," 2018," arXiv:1810.05766.

[85] N. Li, D. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. Girard, "Game-theoretic modeling of driver and vehicle interactions for verifi- cation and validation of autonomous vehicle control systems," 2016," arXiv:1608.08589.

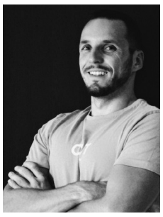
[86] R. Tian, N. Li, I. Kolmanovsky, Y. Yildiz, and A. Girard, "Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation," 2020," arXiv:1910.07141.

[87] A. Sarkar, K. Larson, and K. Czarnecki, "Generalized dynamic cognitive hierarchy models for strategic driving behavior," 2021," arXiv: 2109.09861.

[88] ——, "A taxonomy of strategic human interactions in traffic conflicts," 2021," arXiv:2109.13367.

[89] M. Treiber and A. Kesting, *Traffic flow dynamics*. Springer, 2013. **DOI**: 10.1007/978-3-642-32460-4.

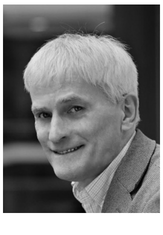
[90] D. Neil, J. Briody, A. Lacoste, A. Sim, P. Creed, and A. Saffari, "Interpretable graph convolutional neural networks for inference on noisy knowledge graphs," 2018," arXiv:1812.00279.



Gergely Hollósi is a researcher at Dept. of Telecommunications and Media Informatics (TMIT) of Budapest University of Technology and Economics (BME). Gergely received his M.Sc. in the Budapest University of Technology and Economics (BME) in 2009. He is actively working and researching on computer vision, image processing, machine learning algorithms and indoor localization systems.

Csaba Lukovszki is a researcher at Budapest University of Technology and Economics (BME). He is technical leader of RnD projects, has courses in areas of telecommunications, radio- and vision-based indoor localization, and machine learning algorithms. He is author of 10 journals and 25 conference papers. He is a member of Scientific Association for Infocomm. in Hungary, from he received one Silver and two Gold medals. For a significant publication he is a proud owner of Pollák-Virág Prize.

Máté Bancsics is a researcher Dept. of Telecommunications and Media Informatics (TMIT) at Budapest University of Technology and Economics (BME). Máté got his Master's Degree in Electrical Engineering at the same institute in 2021. He is actively working and doing research on Indoor Positioning Systems, Internet of Things and Machine Learning Algorithms.



Gábor Magyar is Associated Professor at BME Dept. of Telecommunications and Media Informatics. Head of Data Management Working Group in the Hungarian Mobility Platform, Head of Data Management and Communications in Autonomous Systems National Laboratory, Head of Data Management and Communications at AutomatedDrive Lab. Adviser to the National Council for Infocommunications to the Hungarian Government. Honorary President of the Scientific Association for Infocommunications (HTE), and President of the Hungarian Oracle User Group (HOUG). Served as the Director of Stategic Affaires in BME, and ministerial commissioner for European Institute of Technology and Innovation. He is the recipient of the Kalmár László, Pollák-Virág and Puskás Tivadar awards.

A Survey on Machine Learning based Smart Maintenance and Quality Control Solutions

Attila Frankó[1] and Pál Varga[2]

Abstract—**Machine learning aided tasks and processes have key roles in smart manufacturing, especially in controlling production and assembly lines, as well as smart maintenance and intelligent quality control. The last two ones are those tasks that nowadays are still performed manually by employees; however, there are numerous machine learning-based solutions that can automate these fields to optimize cost and performance. In this paper, we present an overview of smart manufacturing ecosystem and define the roles of maintenance and quality control in it. Up-to-date machine learning-based smart solutions will also be detailed while addressing current challenges and identifying hot research topics and possible gaps.**

Index Terms—**machine learning, smart maintenance, intelligent quality control, diagnostics, health management, computer vision**

I. Introduction

Over the last decade, the number of artificial intelligence (AI) based applications significantly raised, as certain emerging machine learning algorithms and technologies, such as deep learning (DL), gained popularity. Since then, machine learning has matured enough to enable its usage in various fields, from computer vision to machine-type communications. Its ability to solve complex, high-dimensional problems in a wide range made it one of the major drivers of the Industry 4.0 [1] movement besides industrial Internet of Things. As digitalization efforts are increasing within the industry to transform traditional manufacturing into highly automated smart factories, machine learning methods become the pillar of automating, not only manufacturing products within shop floors but maintenance tasks, logistical processes across the whole supply chain, warehouse management, automated quality management, production control – to mention the most popular ones [2].

Several global companies with sufficient technical and financial background have already applied machine learning-based solutions with success to automate certain tasks within warehouses or shop-floors, e.g., autonomous vehicles that can relocate assets without any manual interactions by operators.

Regardless of that, this area is still considered to be relatively new because of two factors: first, many areas within this field – such as automating the aforementioned processes and tasks – are not considered hot topics, despite that few of them (e.g. self-driving vehicles) dominate current research trends. Second, most market players – predominantly Small and Medium-sized Enterprises (SMEs) – have not adopted such state-of-the-art solutions (e.g., robots for logistical tasks) yet, since it would not be beneficial financially in the short term currently. Nonetheless, SMEs can implement new technologies more rapidly than bigger enterprises, therefore they will remain as the backbone of manufacturing supply chains after the digital transition too. Nonetheless, it is more challanging to adopt Industry 4.0 technologies for SMEs often due to financial reasons, therefore it is crucial to choose strategy, since an effective implementation can offer tremendous benefits [3] [4] [5]. Numerous researches investigated methods, strategies and roadmaps for adopting industry 4.0 technologies efficiently for SMEs, thus each related work – including this current article – can relevant for SMEs too [6] [7]. Due to this, there are ongoing researches that investigate the opportunities of utilizing machine learning in smart factories.

These researches mainly focus on (i) automating and optimizing maintenance tasks by monitoring machinery to provide intelligent fault detection and root cause analysis, (ii) predicting remaining useful life of a piece of equipment or tool, and (iii) automating quality management supported by computer vision and machine learning-based classification, creating automated production lines. This paper aims to compile a state-of-the-art survey on machine learning-based methods and solutions regarding smart maintenance and intelligent quality control as core tasks of smart manufacturing. It also describes potential challenges and research gaps of these fields to be solved, as well as challenges that are already addressed and hot research topics.

The paper is organized as follows: Section II gives a general overview of the role of machine learning in smart manufacturing and introduces maintenance and quality control as important tasks within this ecosystem. Section III and section IV investigate machine learning approaches, methods and algorithms regarding smart maintenance (diagnostics and prognostics) and intelligent quality control (non-visual and computer vision aided), respectively. Section V concludes the paper.

II. Overview of smart manufacturing ecosystem

Manufacturing operations include numerous processes and tasks which have always been critical in terms of safety, reliability, and cost-efficiency. Hence, increasing control over these operations is a focal point in industrial digitization trends. Traditionally there are few fields considered the targets of industrial automation – such as assembly or manufacturing generally [8] – but nowadays, as smart factories emerging, these traditional automation targets have been started to share

---

[1,2] Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Hungary

Email: {franko, pvarga}@tmit.bme.hu

DOI: 10.36244/ICJ.2021.4.4

the spotlight with other fields such as maintenance, quality control, and many others. In this paper, the focus is on maintenance and quality control applications, however, these two fields cannot be fully discussed without mentioning related fields, which provide the input for these tasks, and which digitalization also has a significant impact on.

In the last few years, machine learning applications, especially deep learning for smart manufacturing, were investigated in order to discover the areas well-worth improving [10]. Generally, machine learning offers great potentials due to the intense presence of big data and thus opens new ways to develop advanced, data-driven applications [11]. Moreover, machine learning methods such as convolutional and recurrent neural networks, auto encoder, etc. [12] [13] allow us (i) to model complex systems with limited domain knowledge due to the feature extraction and also (ii) to optimize specific processes and tasks based on this model. It is often required to estimate performance and make decisions based on available input data. Authors of [14] presented the concept of a machine learning-based Design Support System (DSS) to address some sort of optimization problems in design processes regarding manufacturing, logistics, and warehouse management. Nonetheless, deep expert knowledge is still required since manufacturing data cannot be utilized in its raw form due to its high dimensions [15], thus regarding smart planning and design, the usage of knowledge-based models is still one of the most important methodologies.

As it is shown in Figure 1, both maintenance and quality control has strong connections with related subfields and tasks within smart manufacturing, such as the aforementioned Smart design and Smart planning. Due to being versatile, machine learning is commonly used within smart industry; therefore, the general research gaps and challenges are the same for these smart tasks. However, specific applications raise new problem sets as well as new solutions; thus, there are unique challenges and research gaps the can be identified. In the following chapters, machine learning applications will be investigated from the aspect of maintenance and quality control to give a comprehensive overview of the identified challenges, gaps, and hot research topics. The figure lists energy optimization or energy efficiency management as one of the goals of maintenance. However, it is a focal point of smart manufacturing; it is a specific task, therefore; in this article, this concept is not presented.

III. MACHINE LEARNING IN SMART MAINTENANCE

One of the most important tasks in each segment of the industry is maintenance. Unexpected malfunctions can lead to undesirable consequences such as stopping assembly lines or reschedule logistical processes, which cause economic loss directly or indirectly e.g., due to delays in operations.

The mechanism of maintenance seems to be simple; however, smart and efficient maintenance involves numerous tasks and each of them improves the efficiency of the overall mechanism. In its purest form, maintenance is reactive; thus, keeping certain tools, machines, or pieces of equipment in good condition is not a part of maintenance. So, in the case of reactive maintenance, these machines and tools will be repaired only after a breakdown and not earlier. Nevertheless, in certain cases, the occurrence of malfunctions is not obvious; therefore, the condition of machinery continues to deteriorate; thus, fault detection techniques are required to indicate the need for maintenance. Additionally, applying diagnostics and root cause analysis can enhance the quality of maintenance, especially when the source of failure is unknown and must be investigated.

Modern maintenance paradigms such as preventive, predictive, and proactive maintenance use different approaches than the reactive one. Preventive maintenance – as its name suggests – focuses on keeping the tools and machinery in good condition, which requires frequent examination of wear-out level. Optimizing the efficiency of this paradigm by eliminating unnecessary examination and repairing is achieved by using telemetry, external sensors, and other condition monitoring techniques to collect data for diagnostics. Predictive maintenance also utilizes the tools above; however, it has a different purpose: to estimate the date of a breakdown so the whole process can be planned. For precise estimations, certain models of the monitored asset are used to calculate its remaining useful life (RUL) [17]. As proactive maintenance unified these two paradigms – predictive and preventive –the used toolsets were combined too, and with the rise of machine learning and Industrial Internet of Things (IIoT), it became a dynamic, data-driven approach.

Being data-driven also means that a massive amount of data is required, often provided by related smart tasks, processes, systems or even logs [18]. For example, the growing tendency in adapting technologies such MES as Plant floor automation and information systems (PES) can give prognostics-based maintenance a boost by generating a huge amount of data [19]. As an essential concept of smart manufacturing, the Digital Twin as a high-level, data-driven, abstract model of systems also eases data aggregation and analytics. It also improves the overall effectiveness of proactive maintenance, thus reducing the potential of failure [20].

A. Diagnostics

Diagnostics in smart maintenance usually consist of two main parts: fault detection (FD) and root cause analysis (RCA). Fault detection aims to monitor the machinery and indicate that if components or parts of the machine wear out or deteriorate, RCA investigates the malfunction to find the root cause.

Fault detection has always been a difficult task for learning systems due to the overly biased training data. Moreover, despite the numerous available high-performance learning algorithms, prior domain knowledge is required to implement effective fault detection and diagnosis (FDD) systems [21]. In this case, the aim of learning is to make the learning system detect anomalies successfully, and the obvious bottleneck is not having a sufficient number of samples that represent faults and malfunctions, since, in a well-maintained environment, such things are avoided as much as possible. In [22] Fathy et al. presented a comprehensive analysis of different techniques to deal with imbalanced data. Although their results showed

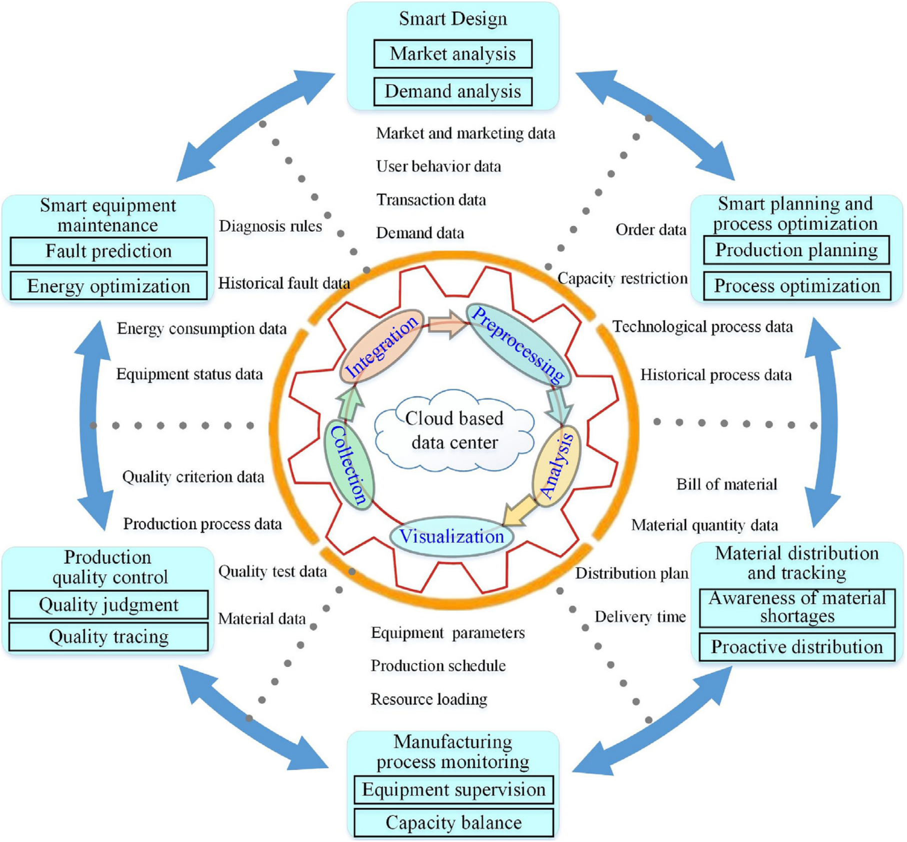


Fig. 1. The role of smart maintenance and smart quality control in a data-driven smart manufacturing ecosystem [9].

no one-solution-fits-all in binary classification, applying synthetic oversampling or generative adversarial networks (GAN) highly improves fault detection performance. Authors of [23] presented a deep learning-based FDD method that utilizes GAN to generate fake samples in order to balance and expand the training dataset and showed the performance of different methods for especially rotating machinery as a common type of equipment in manufacturing. Authors of [24] also sucessfully utilized a customzied GAN to handle imbalanced-class problem for fault detection diagnosis of chillers. In [25] an FDD method was introduced towards imbalanced data as well. However, the proposed new WMODA technique – based on synthetic oversampling – is limited to binary-class imbalanced data problems; it was demonstrated that WMODA performs better than several other imbalanced data learning methods and the traditional data-driven methods.

Another major drawback of ML-based fault detection applications is that a trained system can be used only for one specific machine, which is represented by the collected dataset [21]. It would be promising if similar systems with the same properties and characteristics by default could be trained by a common, unified dataset consisting of each collected sample from all involved systems. Transfer learning aims to solve this problem set. Lei et al. [26] address transfer learning as one of the most important research topics within ML-based, and big data enabled fault detection in the future. It also highlights the importance of deep transfer learning, GAN, and Transfer factor analysis as methodologies capable of being used in transfer scenarios where data is collected from identical machines (TIM) as well as from different machines (TDM). In [27], authors propose a deep generative transfer learning system for fault diagnostics on new hard disk, and showed that this method outperfomed other ones in this application due to the low number of real faulty samples.

The other important task in diagnostics is the aforementioned root cause analysis, which is more of a collection of models and techniques that share the same basic principles and workflow as one straight algorithm or method. The chosen model depends on the problem set and the observed system since domain, or system knowledge may be required. Authors of [28] presented an exhaustive and detailed survey of applied RCA approaches, methods, and technologies, and a broad set of these can be seen in Figure 2. In [29] address two major issues regarding RCA, (i) the lack of supervised datasets and (ii) the need for interpretability to foster trust are addressed as major obstacles of intensive use of machine learning in anomaly detection; therefore, the authors propose random forest technique to enable using unlabeled data in root cause analysis.

However, there is much ongoing research in this field; these works mainly discuss topics from a general perspective rather than presenting actual applications – in contrast to the previous

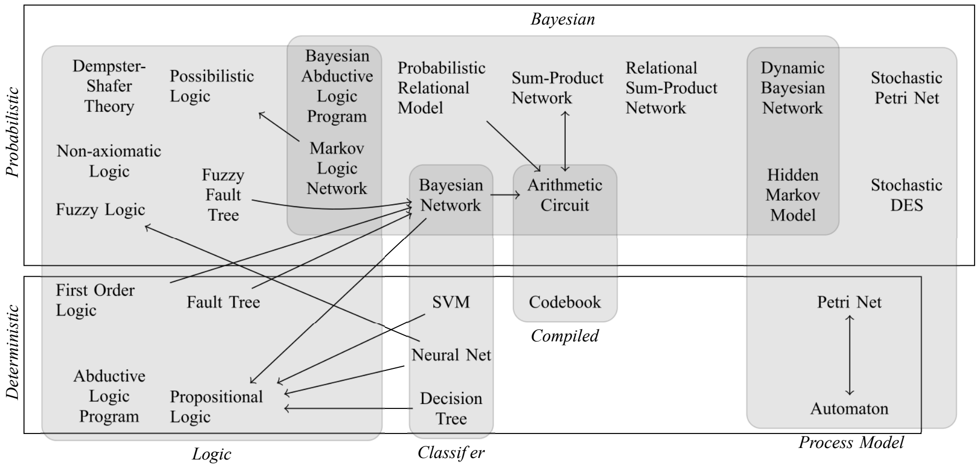


Fig. 2. Classification of models and methods that are used in Root Cause Analysis. Directed arrows indicate possible conversions between different models [16].

decade. Nonetheless the number of articles that utilize the aforementioned methods such as GANs or transfer learning in industrial use-cases is increasing as it is shown in table I.

Citation	Application	Used method
[30]	Power plant thermal system	transfer learning (HDDA)
[31]	Rolling element bearings (train bogie)	transfer learning (many)
[32]	Laboratory/Locomotive bearings	transfer learning (deep generative)
[33]	rolling bearings	GAN
[34]	Wind Turbines Gearbox	GAN

TABLE I
CUTTING-EDGE SMART MAINTENANCE APPLICATIONS

B. Prognostics, predictions and health management

Prognostics and predictions – or as often called: predictions and health management (PHM) – play essential roles in smart maintenance since they enable precise and accurate scheduling maintenance tasks in an optimal manner. The basic concept includes constructing a model of the machine based on historical data and monitoring it in real-time, and predicting its future behavior by using real-time data as the input of the established model. This concept's main goal is to estimate the remaining useful life (RUL) of a tool or machinery as precisely as possible.

Making predictions is beneficial in such a field where machine learning techniques perform very well traditionally [35], since most predictions include different hidden factors that are challenging or impossible to investigate by other methods. Certain applications and researches involving prognostics to calculate RUL are related to medical sciences, healthcare, or health management, where the main targets of the estimation are patients or course of disease. Nevertheless, these results also come in handy for smart maintenance and often directly applicable in such systems.

Traditionally, there are three approaches that are widely used in a prognostic manner: physics/model-based, data-driven, and hybrid approaches [36]. In the last decade, the data-driven approach obviously dominated this field [37] [38] with mostly stochastic algorithms – most notably Bayesian networks – which and are still efficient in related fields or in safety use-case where the uncertainty of a solution has to be known [39] [40].

Regarding the applied technologies that are shown in Figure 3, it is safe to say that a wide range of machine learning techniques and algorithms are used in prognostics that cover specific use-cases [41], depending on the environment, the input data, the level of domain or expert knowledge. Therefore in this topic, the main challenges of machine learning-based techniques are usually related to these aspects, such as data acquisition and handling big data in terms of scalability, latency, and network bandwidth, moreover finding machine learning applications that are suitable for multiple use-cases [42].

It is also worth mentioning that some of the issues addressed in other sections of this paper, such as dealing with unlabeled data [43] or flexibility of a solution [44] are also valid challenges for prognostics and health management. It goes for the same for future researches regarding transfer learning, generative algorithms, and physics-induced machine learning as well [45] [46].

IV. INTELLIGENT QUALITY CONTROL

While proper maintenance ensures that the production line behaves and works as it should, quality control ensures that the final product meets different (inner and outer) standards. Quality control is as important, if not more so, than maintenance, since if quality control fails, poor or inadequate products will be shipped for the customers.

The first big, revolutionary step in quality control was when the application of statistical methods, such as control charts and acceptance sampling, had begun. Since then, the theory of quality management and quality control has been evolving by creating new approaches and paradigms that fuel the automating intentions within the industry towards quality control processes. In this topic, the machine learning applications are very heterogenous since they include regular

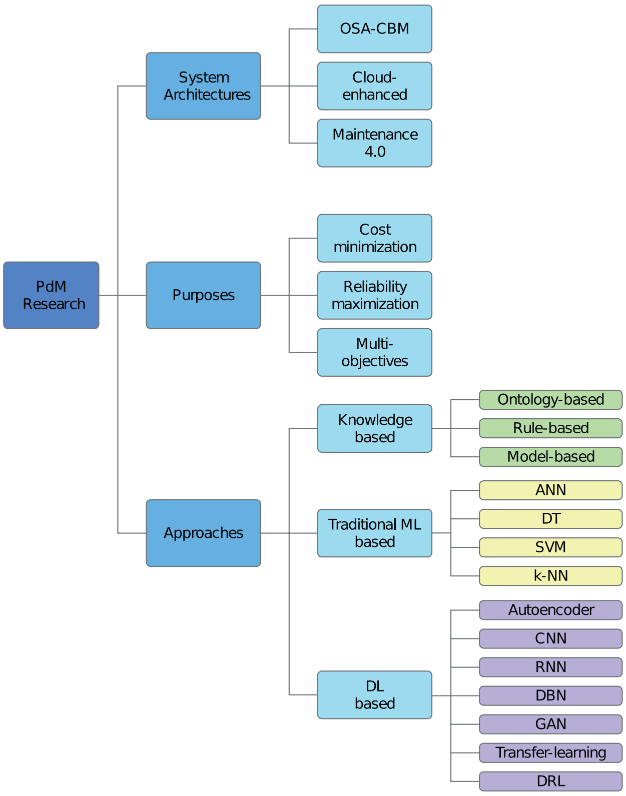


Fig. 3. Taxonomy of predictive maintenance including the applied technologies and models [47]

classification based on collected data, but also computer vision and image recognition.

As Cyber-physical production systems (CPPS) and IIoT became prevalent paradigms within the industry, they enabled such data-intensive applications like the aforementioned digital twin concept. In [48] the authors proposed a new Digital Twin modeling method based on Multi-agent System (MAS) and semantic models that provide improved quality control during the manufacturing phase of the product. Authors of [49] also introduced a digital twin-based quality control solution while emphasizing that it is strongly coupled to assembly and production line optimization.

From the technical aspect, smart maintenance and smart quality control might seem analogous, so in most cases, problems can be formulated in the same way: e.g., monitoring and quality inspection of machinery can be implemented by using the same approach and technologies that would be used in quality insurance or quality control of products. This would imply that all the solutions above for smart maintenance could be applied for quality control; however, these tasks have different requirements and characteristics than maintenance tasks.

While smart maintenance deals mostly with inspecting and monitoring a static, single piece of equipment, quality inspection covers the investigation and classification of great quantities of products that change dynamically. Consequently, the cited smart maintenance design approaches and technologies are also frequently used for quality control applications, but this field must be discussed separately since its unique requirements. Moreover, it is worth noting that quality control applications usually rely upon domain knowledge-based defect detection architectures, design approaches, and solutions.

A. *Non-visual quality control applications*

This approach of quality control, which involves machine learning applications – mostly deep neural networks, but not computer vision – relies upon telemetry data and datasets that are measured and gathered by sensors across the whole assembly or production line [50]. As with all similar machine learning solutions, this one also requires a huge amount of data in the training phase; therefore, the implemented system will be rigid since each time the production line changes, the model has to be retrained again. While computer vision-based solutions offer more flexibility, this approach is fast, reliable, and considered lightweight. Moreover, techniques like Lifelong Learning with Deep neural networks (L-DNN) [51] can enhance the flexibility of such a quality control system, creating an alternative to image recognition-based solutions that can be considered lightweight.

Most of the quality control application that uses machine learning formalize the problem as binary classification, e.g., pass or deny. However, certain cases require more complex formalization; therefore, the designed architecture, the applied learning algorithms, and pattern recognition strategies often vary. In [52] an l_1 regularized logistic regression is used as a learning algorithm in order to create a system that can successfully detect rare quality events to achieve defect-free processes. Authors of [52] also presented two use-cases, where different model selection criteria were applied to find the optimal one among multiple ones, which were created as a result of manually defining features. It has to be emphasized that regarding machine learning-based quality control applications, there are no one-size-fits-all approaches, algorithms, or technologies because the characteristics of the production line have a huge impact on the system to be implemented.

The properties and characteristics of the proposed solutions can be very diverse depending on the applied architecture, algorithm, approaches, and other building blocks of such systems. However, the accuracy of classification is one of the most important properties; availability, reliability, performance, response time, and resource requirements play key roles in choosing the appropriate solution for a specific task, as always.

Authors of [53] investigate several binary classification models for quality control of Multistage Manufacturing Process (MMP) and measured the performance of different algorithms, namely: Gaussian Naive Bayes (GNB), K-Nearest Neighbours (KNN), XGBOOST, Random Forest, and Support Vector Machine (SVM). Since in MMP an early stage defect or failure will propagate, it is essential to detect these defects as soon and fast as possible, however; the quality inspection problem of each stage is less complex than the final one. In [53] it was shown, the aforementioned algorithms performed very well in an MMP; while they are considered lightweight in contrast to any neural networks, therefore they can be efficiently applied at each stage of MMP to inspect production quality.

In [54] a Bi-LSTM (Long-Short Term Memory) based solution was proposed, where the implemented system uses control charts and histograms as input data to find patterns; therefore, it is an indirect approach, which uses statistical learning. In this case, it is easier to define feature sets for recognizing histogram and control chart patterns than using raw data as input; therefore, this solution can be considered lightweight.

When it comes to performance – and scalability in most cases – where lightweight solutions can not be utilized, cloud computing is an effective and usually cost-efficient way to deal with heavy resource issues [55]. Nowadays, almost every cloud provider offers services for machine learning applications; moreover, these can be connected to other cloud-based applications such as warehouse management, etc. [56]. It can often be the optimal solution if the performance must be improved, although in this case, the quality of the connection will be the bottleneck. Authors of [57] introduce a combined edge-cloud-based quality inspection system that trains the model in the cloud to enable high-performance in training phase, but it deploys the model in the edge; thus, the response time will be much faster compared to an only cloud-based one.

B. Computer vision-aided quality control

Another technique regarding machine learning-aided quality control is the computer vision (CV) based quality inspection. However, CV-enabled feature extraction is not a fresh field; it is still a hot topic since it has not been adopted by industrial players as fast as it was expected. Nevertheless, this technology is used more widely than the one mentioned above since it is suitable for quality control of non-manufactured products (e.g., food) that can not be monitored one by one via sensors.

This method is much more flexible than the previous one since it can handle complex, complicated, non-binary classification problems while being robust. Moreover, it can be applied in dynamic environments that often change because it is contactless; while numerous methods based on sensing require some sort of physical interaction, therefore they are less flexible or not flexible at all. Nonetheless, image processing is almost always slower than applying binary classification (including inference and training time) purely based on sensor or telemetric data; therefore, the applicability of this technique in the case of fast, high-throughput assembly or production lines is limited [58], [59].

The flexibility and performance of such a system depend on the applied technology and the followed approaches while designing the architecture. The most important factors are usually the features that will be examined during the investigation of an object to find defects; therefore, defining features is a crucial task of designing an optical quality control (OQC) system. Another approach to design fast and reliable OQC systems is applying Convolutional Neural Networks (CNN) that provide automatic feature extraction instead of the aforementioned manual one. In [58] Weimer et al. showed that CNN-based architectures are capable of identifying defects with high accuracy; therefore, it is possible to create reliable OQC systems with limited prior knowledge on the problem domain. Authors of [60] presented a CNN-based OQC system with unique architecture to find the optimal accuracy and training time. Therefore they limited the number of the used convolutional layer. They showed that an optimum exists for this specific task, which has better (or similar) accuracy than other CNN architecture while its training time is considerably faster.

However, image processing is usually a computation heavy task; therefore, in certain cases, it can be beneficial to apply prior knowledge to increase the system's performance. The authors of [59] designed and implemented a Convolutional Neural Network-based quality control system for PET bottle caps, which system is supported by image calibration to enable using a custom, lightweight architecture. In spite of that, not so customizable CV frameworks that enable fast prototyping are extremely popular in OQC systems: e.g., a well-known one is the ”You only look once” (YOLOv3) [61]. YOLOv3 provides impressive performance with appropriate detection accuracy and very fast inference time, although it is not the best-performing choice compared to the previous one.

V. CONCLUSION

In this paper, we investigated machine learning-based applications within smart maintenance and intelligent quality control that are core tasks of smart manufacturing while identifying the most important application-specific challenges and research gaps.

Regarding smart maintenance applications: (i) synthetic oversampling, generative adversarial networks, and (ii) transfer learning have been identified as hot research topics that address the problem of (i) imbalanced, overly biased data and (ii) using the same dataset for other different or identical machines. The lack of supervised datasets is a common issue for RCA, IDF, prognostics, and many fields, which may lose its importance as smart manufacturing becomes more mature. However, right now, the best practice is using such an approach the can easily deal with unlabeled data, e.g., random forest. So, the field of applications with unlabeled data is identified as a hot topic as well, since in this early-mid phase of smart factory development, using unlabeled data set is a common scenario.

In the field of quality control, both non-visual and computer vision-aided applications have a great reputation. Non-visual solutions are lightweight CV-based solutions that are much more flexible, while both of them perform very well. The main challenges for non-visual quality control applications are: enabling high-data-rate and enhance flexibility to enable quality control in the ever-changing smart factories. The main challenges for CV-based applications are: improve the inference time and decrease the required computational capacity to enable image recognition-based quality inspection in high-throughput assembly and production lines.

REFERENCES

[1] DIN Std., *Reference Architecture Model Industrie 4.0 (RAMI4.0)*, DIN SPEC 91345, 04 2016.

[2] J. P. Usuga Cadavid, S. Lamouri, B. Grabot, R. Pellerin, and A. Fortin, "Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0," *Journal of Intelligent Manufacturing*, 01 2020, **DOI**: 10.1007/s10845-019-01531-7.

[3] D. Matt, V. Modrák, and H. Zsifkovits, *Industry 4.0 for SMEs Challenges, Opportunities and Requirements: Challenges, Opportunities and Requirements*, 01 2020, **DOI**: 10.1007/978-3-030-25425-4.

[4] S. Menon and S. Shah, "Are smes ready for industry 4.0 technologies: An exploratory study of i 4.0 technological impacts," in *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, 2020, pp. 203–208, **DOI**: 10.1109/ICCAKM46823.2020.9051550.

[5] E. Sezer, D. Romero, F. Guedea, M. Macchi, and C. Emmanouilidis, "An industry 4.0-enabled low cost predictive maintenance approach for smes," in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2018, pp. 1–8, **DOI**: 10.1109/ICE.2018.8436307.

[6] A. Cotrino, M. A. Sebastián, and C. González-Gaya, "Industry 4.0 roadmap: Implementation for small and medium-sized enterprises," *Applied Sciences*, vol. 10, no. 23, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/23/8566. **DOI**: 10.3390/app10238566

[7] T. Taurino and A. Villa, "A method for applying industry 4.0 in small enterprises," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 439–444, 2019, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319310407. **DOI**: 10.1016/j.ifacol.2019.11.099

[8] R. Seidel, A. Mayr, F. Schäfer, D. Kißkalt, and J. Franke, "Towards a smart electronics production using machine learning techniques," in *2019 42nd International Spring Seminar on Electronics Technology (ISSE)*, 2019, pp. 1–6, **DOI**: 10.1109/ISSE.2019.8810176.

[9] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 157–169, 2018, special Issue on Smart Manufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612518300062. **DOI**: 10.1016/j.jmsy.2018.01.006

[10] M. Sharp, R. ak, and T. Hedberg, "A survey of the advancing use and development of machine learning in smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, 03 2018, **DOI**: 10.1016/j.jmsy.2018.02.004.

[11] C. Y. Park, J. W. Kim, B. Kim, and J. Lee, "Prediction for manufacturing factors in a steel plate rolling smart factory using data clustering-based machine learning," *IEEE Access*, vol. 8, pp. 60 890–60 905, 2020, **DOI**: 10.1109/ACCESS.2020.2983188.

[12] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018, special Issue on Smart Manufacturing, **DOI**: 10.1016/j.jmsy.2018.01.003.

[13] R. Iqbal, T. Maniak, F. Doctor, and C. Karyotis, "Fault detection and isolation in industrial processes using deep learning approaches," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3077–3084, 2019, **DOI**: 10.1109/TII.2019.2902274.

[14] L. Romeo, J. Loncarski, M. Paolanti, G. Bocchini, A. Mancini, and E. Frontoni, "Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0," *Expert Systems with Applications*, vol. 140, p. 112869, 08 2019, **DOI**: 10.1016/j.eswa.2019.112869.

[15] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2018, **DOI**: 10.1109/ACCESS.2017.2783682.

[16] M. Solé, V. Muntés-Mulero, A. I. Rana, and G. Estrada, "Survey on models and techniques for root-cause analysis," 2017.

[17] J. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modeling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing*, vol. 25, pp. 1803–1836, 12 2011, **DOI**: 10.1016/j.ymssp.2010.11.018.

[18] M. Maurer, A. Festl, B. Bricel, G. Schneider, and M. Schmeja, "Automl for log file analysis (alfa) in a production line system of systems pointed towards predictive maintenance," *Infocommunications Journal*, vol. 13, no. 3, pp. 76 – 84, 2021, **DOI**: 10.36244/ICJ.2021.3.8.

[19] S. T. Bukkapatnam, K. Afrin, D. Dave, and S. R. Kumara, "Machine learning and ai for long-term fault prognosis in complex manufacturing systems," *CIRP Annals*, vol. 68, no. 1, pp. 459–462, 2019, **DOI**: 10.1016/j.cirp.2019.04.104.

[20] A. Oluwasegun and J.-C. Jung, "The application of machine learning for the prognostics and health management of control element drive system," *Nuclear Engineering and Technology*, vol. 52, no. 10, pp. 2262–2273, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1738573319308654. **DOI**: 10.1016/j.net.2020.03.028

[21] S. R. Saufi, Z. A. B. Ahmad, M. S. Leong, and M. H. Lim, "Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review," I*EEE Access*, vol. 7, pp. 122644–122662, 2019, **DOI**: 10.1109/ACCESS.2019.2938227.

[22] Y. Fathy, M. Jaber, and A. Brintrup, "Learning with imbalanced data in smart manufacturing: A comparative analysis," *IEEE Access*, vol. PP, 12 2020, **DOI**: 10.1109/ACCESS.2020.3047838.

[23] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, p. 107377, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119312412. **DOI**: 10.1016/j.measurement.2019.107377

[24] K. Yan, A. Chong, and Y. Mo, "Generative adversarial network for fault detection diagnosis of chillers," *Building and Environment*, vol. 172, p. 106698, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360132320300561. **DOI**: 10.1016/j.buildenv.2020.106698

[25] Y. Zhang, X. Li, L. Gao, L. Wang, and L. Wen, "Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning," *Journal of Manufacturing Systems*, vol. 48, pp. 34–50, 2018, special Issue on Smart Manufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027861251830044X. **DOI**: 10.1016/j.jmsy.2018.04.005

[26] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327019308088. **DOI**: 10.1016/j.ymssp.2019.106587

[27] C. Shi, Z. Wu, X. Lv, and Y. Ji, "Dgtl-net: A deep generative transfer learning network for fault diagnostics on new hard disks," *Expert Systems with Applications*, vol. 169, p. 114379, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S0957417420310551. **DOI**: 10.1016/j.eswa.2020.114379

[28] M. Solé, V. Muntés-Mulero, A. I. Rana, and G. Estrada, "Survey on models and techniques for root-cause analysis," 2017. [Online]. Available: https://arxiv.org/abs/1701.08546

[29] M. Carletti, C. Masiero, A. Beghi, and G. A. Susto, "Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 21–26, **DOI**: 10.1109/SMC.2019.8913901.

[30] X. Wang, H. He, and L. Li, "A hierarchical deep domain adaptation approach for fault diagnosis of power plant thermal system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5139–5148, 2019, **DOI**: 10.1109/TII.2019.2899118.

[31] X. Li, W. Zhang, and Q. Ding, "Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5525–5534, 2019, **DOI**: 10.1109/TIE.2018.2868023.

[32] B. Yang, Y. Lei, F. Jia, and S. Xing, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," *Mechanical Systems and Signal Processing*, vol. 122, pp. 692–706, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327018308367. **DOI**: 10.1016/j.ymssp.2018.12.051

[33] F. Zhou, S. Yang, H. Fujita, D. Chen, and C. Wen, "Deep learning fault diagnosis method based on global optimization gan for unbalanced data," *Knowledge-Based Systems*, vol. 187, p. 104837, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705119303120. **DOI**: 10.1016/j.knosys.2019.07.008

[34] X. Zhang, P. Wu, J. He, S. Lou, and J. Gao, "A gan based fault detection of wind turbines gearbox," in *2020 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, 2020, pp. 271–275, **DOI**: 10.1109/ICCSS52145.2020.9336899.

[35] C. Cheng, G. Ma, Y. Zhang, M. Sun, F. Teng, H. Ding, and Y. Yuan, "A deep learning-based remaining useful life prediction approach for bearings," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1243–1254, 2020, doi: 10.1109/TMECH.2020.2971503.

[36] Y. Wang, Y. Zhao, and S. Addepalli, "Remaining useful life prediction using deep learning approaches: A review," Procedia Manufacturing, vol. 49, pp. 81–88, 2020, proceedings of the 8th International Conference on Through-Life Engineering Services – TESConf 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978920316528. doi: 10.1016/j.promfg.2020.06.015

[37] R. K. Singleton, E. G. Strangas, and S. Aviyente, "Extended kalman filtering for remaining-useful-life estimation of bearings," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1781–1790, 2015, doi: 10.1109/TIE.2014.2336616.

[38] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2276–2285, 2017, doi: 10.1109/TIE.2016.2623260.

[39] C. HU, H. PEI, Z. WANG, X. SI, and Z. ZHANG, "A new remaining useful life estimation method for equipment subjected to intervention of imperfect maintenance activities," *Chinese Journal of Aeronautics*, vol. 31, no. 3, pp. 514–528, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936118300256. doi: 10.1016/j.cja.2018.01.009

[40] M. Benker, L. Furtner, T. Semm, and M. F. Zaeh, "Utilizing uncertainty information in remaining useful life estimation via bayesian neural networks and hamiltonian monte carlo," *Journal of Manufacturing Systems*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520301928. doi: 10.1016/j.jmsy.2020.11.005

[41] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustainability*, vol. 12, no. 19, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/19/8211. doi: 10.3390/su12198211

[42] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, and J. Barbosa, "Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges," *Computers in Industry*, vol. 123, p. 103298, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361520305327. doi: 10.1016/j.compind.2020.103298

[43] A. Listou Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliability Engineering & System Safety*, vol. 183, pp. 240–251, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832018307506. doi: 10.1016/j.ress.2018.11.027

[44] M. Kraus and S. Feuerriegel, "Forecasting remaining useful life: Interpretable deep learning approach via variational bayesian inferences," *Decision Support Systems*, vol. 125, p. 113100, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923619301290. doi: 10.1016/j.dss.2019.113100

[45] O. Fink, Q. Wang, M. Svensen, P. Dersin, W.-J. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103678, 06 2020, doi: 10.1016/j.engappai.2020.103678.

[46] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0," *Information Fusion*, vol. 50, 10 2018, doi: 10.1016/j.inffus.2018.10.005.

[47] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," 2019.

[48] X. Zheng, F. Psarommatis, P. Petrali, C. Turrin, L. Jinzhi, and D. Kiritsis, "A quality-oriented digital twin modelling method for manufacturing processes based on a multi-agent architecture," vol. 51, 06 2021, doi: 10.1016/j.promfg.2020.10.044.

[49] Y. Ma, H. Zhou, H. He, G. Jiao, and S. Wei, "A digital twin-based approach for quality control and optimization of complex product assembly," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, 2019, pp. 762–767, doi: 10.1109/AIAM48774.2019.00157.

[50] M. Miškuf and I. Zolotová, "Comparison between multi-class classifiers and deep learning with focus on industry 4.0," in *2016 Cybernetics Informatics (K I)*, 2016, pp. 1–5, doi: 10.1109/CYBERI.2016.7438633.

[51] G. Parisi, R. Kemker, J. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 02 2019, doi: 10.1016/j.neunet.2019.01.012.

[52] C. Escobar and R. Morales-Menendez, "Machine learning techniques for quality control in high conformance manufacturing environment," *Advances in Mechanical Engineering*, vol. 10, p. 168781401875551, 02 2018, doi: 10.1177/1687814018755519.

[53] R. S. Peres, J. Barata, P. Leitao, and G. Garcia, "Multistage quality control using machine learning in the automotive industry," *IEEE Access,* vol. 7, pp. 79 908–79 916, 2019, doi: 10.1109/ACCESS.2019.2923405.

[54] T. Zan, Z. Liu, Z. Su, M. Wang, X. Gao, and D. Chen, "Statistical process control with intelligence based on the deep learning model," *Applied Sciences*, vol. 10, no. 1, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/1/308

[55] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018, doi: 10.1109/MNET.2018.1700202.

[56] "Aws smart factory - manufacturing operations in the cloud," 2021.[Online]. Available: https://aws.amazon.com/manufacturing/smart-factory/

[57] J. Schmitt, J. Bönig, T. Borggräfe, G. Beitinger, and J. Deuse, "Predictive model-based quality inspection using machine learning and edge cloud computing," *Advanced Engineering Informatics*, vol. 45, p. 101101, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034620300707. doi: 10.1016/j.aei.2020.101101

[58] "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016, doi: 10.1016/j.cirp.2016.04.072.

[59] M. Malesa and P. Rajkiewicz, "Quality control of pet bottles caps with dedicated image calibration and deep neural networks," *Sensors*, vol. 21, p. 501, 01 2021, doi: 10.3390/s21020501.

[60] J. Villalba-Diez, D. Schmidt, R. Gevers, J. Ordieres-Meré, M. Buchwitz, and W. Wellbrock, "Deep learning for industrial computer vision quality control in the printing industry 4.0," *Sensors*, vol. 19, no. 18, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/18/3987. doi: 10.3390/s19183987

[61] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

Attila Frankó received his M.Sc. degree in 2019 in Electrical Engineering from the Budapest University of Technology and Economics, Hungary. He is currently working on his Ph.D. degree at the Department of Telecommunications and Media Informatics. His research interests include Industrial Internet of Things – especially machine learning based smart maintenance, logistic applications, cooperative Systems and, low-powered security for wireless applications.

Pal Varga received his Ph.D. degree from the Budapest University of Technology and Economics, Hungary. He is currently an Associate Professor at the Budapest University of Technology and Economics and also the Director at AITIA International Inc. His main research interests include communication systems, Cyber-Physical Systems and Industrial Internet of Things, network traffic analysis, end-to-end QoS and SLA issues – for which he is keen to apply hardware acceleration and artificial intelligence, machine learning techniques as well.
Besides being a member of HTE, he is a senior member of IEEE, where he is active both in the IEEE ComSoc (Communication Society) and IEEE IES (Industrial Electronics Society) communities. He is Editorial Board member of the Sensors (MDPI) and Electronics (MDPI) journals, and the Editor-in-Chief of the Infocommunications Journal.

Guidelines for our Authors

Format of the manuscripts

Original manuscripts and final versions of papers should be submitted in IEEE format according to the formatting instructions available on
https://journals.ieeeauthorcenter.ieee.org/
Then click: "IEEE Author Tools for Journals"
- "Article Templates"
- "Templates for Transactions".

Length of the manuscripts

The length of papers in the aforementioned format should be 6-8 journal pages.
Wherever appropriate, include 1-2 figures or tables per journal page.

Paper structure

Papers should follow the standard structure, consisting of *Introduction* (the part of paper numbered by “1”), and *Conclusion* (the last numbered part) and several *Sections* in between.
The Introduction should introduce the topic, tell why the subject of the paper is important, summarize the state of the art with references to existing works and underline the main innovative results of the paper. The Introduction should conclude with outlining the structure of the paper.

Accompanying parts

Papers should be accompanied by an *Abstract* and a few *Index Terms (Keywords)*. For the final version of accepted papers, please send the short cvs and *photos* of the authors as well.

Authors

In the title of the paper, authors are listed in the order given in the submitted manuscript. Their full affiliations and e-mail addresses will be given in a footnote on the first page as shown in the template. No degrees or other titles of the authors are given. Memberships of IEEE, HTE and other professional societies will be indicated so please supply this information. When submitting the manuscript, one of the authors should be indicated as corresponding author providing his/her postal address, fax number and telephone number for eventual correspondence and communication with the Editorial Board.

References

References should be listed at the end of the paper in the IEEE format, see below:

a) Last name of author or authors and first name or initials, or name of organization
b) Title of article in quotation marks
c) Title of periodical in full and set in italics
d) Volume, number, and, if available, part
e) First and last pages of article
f) Date of issue
g) Document Object Identifier (DOI)

[11] Boggs, S.A. and Fujimoto, N., “Techniques and instrumentation for measurement of transients in gas-insulated switchgear,” IEEE Transactions on Electrical Installation, vol. ET-19, no. 2, pp.87–92, April 1984. DOI: 10.1109/TEI.1984.298778

Format of a book reference:

[26] Peck, R.B., Hanson, W.E., and Thornburn, T.H., Foundation Engineering, 2nd ed. New York: McGraw-Hill, 1972, pp.230–292.

All references should be referred by the corresponding numbers in the text.

Figures

Figures should be black-and-white, clear, and drawn by the authors. Do not use figures or pictures downloaded from the Internet. Figures and pictures should be submitted also as separate files. Captions are obligatory. Within the text, references should be made by figure numbers, e.g. “see Fig. 2.”
When using figures from other printed materials, exact references and note on copyright should be included. Obtaining the copyright is the responsibility of authors.

Contact address

Authors are requested to submit their papers electronically via the following portal address:

https://www.ojs.hte.hu/infocommunications_journal/about/submissions

If you have any question about the journal or the submission process, please do not hesitate to contact us via e-mail:

Editor-in-Chief: Pál Varga – pvarga@tmit.bme.hu

Associate Editor-in-Chief:
Rolland Vida – vida@tmit.bme.hu
László Bacsárdi – bacsardi@hit.bme.hu

IEEE GLOBECOM®

IEEE GLOBAL COMMUNICATIONS CONFERENCE

4-8 December 2022 – Rio de Janeiro, Brazil

CALL FOR PAPERS AND PROPOSALS

Accelerating the Digital Transformation through Smart Communications

The 2022 IEEE Global Communications Conference (GLOBECOM) will be held in the warm and exciting city of Rio de Janeiro, Brazil, from 4 to 8 December 2022. Themed "Accelerating the Digital Transformation through Smart Communications," this flagship conference of the IEEE Communications Society will feature a comprehensive high-quality technical program including 13 symposia and a variety of tutorials and workshops. IEEE GLOBECOM 2022 will also include an attractive industry program aimed at practitioners, with keynotes and panels from prominent research, industry and government leaders, business and industry panels, and technological exhibits.

TECHNICAL SYMPOSIA

- Cognitive Radio & AI-Enabled Networks
- Communication & Information Systems Security
- Communication QoS, Reliability & Modeling
- Communications Software & Multimedia
- Communication Theory Symposium
- Green Communication Systems & Networks
- IoT & Sensor Networks
- Mobile & Wireless Networks Symposium
- Next-Generation Networking & Internet
- Optical Networks & Systems
- Signal Processing for Communications
- Wireless Communications
- Select Areas in Communications
  - *Smart Grid Communications*
  - *Aerial Communications*
  - *Big Data*
  - *Cloud Computing & Networking*
  - *e-Health*
  - *Machine Learning for Communications*
  - *Molecular, Biological, & Multi-Scale Communications*
  - *Satellite & Space Communications*
  - *Backhaul / Fronthaul Communications*
  - *Full Duplex Communications*
  - *Quantum Communications*
  - *Reconfigurable Intelligent Surfaces*
  - *Social Networks*

INDUSTRY FORUMS AND EXHIBITION PROGRAM

Proposals are sought for forums, panels, demos, seminars, and presentations specifically related to issues facing the broader communications and networking industries.

TUTORIALS

Proposals are invited for half- or full-day tutorials in all communications and networking topics.

WORKSHOPS

Proposals are invited for half- or full-day workshops in all communications and networking topics.

IEEE ComSoc® | IEEE

IMPORTANT DATES

PAPER SUBMISSION
15 April 2022

ACCEPTANCE NOTIFICATION
25 July 2022

CAMERA-READY TECHNICAL
1 September 2022

TUTORIAL PROPOSALS
15 April 2022

WORKSHOP PROPOSALS
15 March 2022

PANEL PROPOSALS
1 June 2022

Full details of submission procedures are available at globecom2022.ieee-globecom.org

ORGANIZING COMMITTEE

Honorary General Chair
J. Roberto de Marca (Catholic University, Brazil)

General Chair
Nelson Fonseca (UNICAMP, Brazil)

Vice General Chair
Stefano Bregni (Polytechnic University of Milan, Italy)

Executive Chair
Lisandro Zambenedetti Granville (UFRGS, Brazil)

Technical Program Chair
Fabrizio Granelli (University of Trento, Italy)

Technical Program Vice-Chairs
Christos Verikoukis (University of Barcelona, Spain)
Shiwen Mao (Auburn University, USA)

Tutorials Co-Chairs
Abbas Jamalipour (University of Sydney, Australia)
Stefano Giordano (University of Pisa, Italy)

Keynotes Chair
Khaled. B. Letaief (The Hong Kong University of Science and Technology, Hong Kong)

Workshops Co-Chairs
Mohammed Atiquzzaman (Univ. Oklahoma, USA)
Ying-Dar Lin (National Chiao Tung University, Taiwan)
Peter Mueller (IBM Zurich, Switzerland)

Industrial Forum & Exhibition Co-Chairs
Ashutosh Dutta (John Hopkins University, USA)
Demi Getschko (CGI.br/NIC.br, Brazil)

Sponsorship Co-Chairs
Tim Weil (Security Feeds, USA)
Harry Sauberman, P.E.

Awards Co-Chairs:
Octavia A. Dobre (Memorial University, Canada)
Ekram Hossain (University of Manitoba, Canada)

GITC Advisor:
Sonia Aissa, INRS, Canada

GIMS Advisor:
Deep Medhi, Univ. of Missouri-Kansas, USA

SCIENTIFIC ASSOCIATION FOR INFOCOMMUNICATIONS

hte

FOUNDED IN 1949

SCIENTIFIC
ASSOCIATION FOR
INFOCOMMUNICATIONS
HUNGARY

Who we are

Founded in 1949, the Scientific Association for Infocommunications (formerly known as Scientific Society for Telecommunications) is a voluntary and autonomous professional society of engineers and economists, researchers and businessmen, managers and educational, regulatory and other professionals working in the fields of telecommunications, broadcasting, electronics, information and media technologies in Hungary.

Besides its 1000 individual members, the Scientific Association for Infocommunications (in Hungarian: Hírközlési és Informatikai Tudományos Egyesület, HTE) has more than 60 corporate members as well. Among them there are large companies and small-and-medium enterprises with industrial, trade, service-providing, research and development activities, as well as educational institutions and research centers.

HTE is a Sister Society of the Institute of Electrical and Electronics Engineers, Inc. (IEEE) and the IEEE Communications Society.

What we do

HTE has a broad range of activities that aim to promote the convergence of information and communication technologies and the deployment of synergic applications and services, to broaden the knowledge and skills of our members, to facilitate the exchange of ideas and experiences, as well as to integrate and harmonize the professional opinions and standpoints derived from various group interests and market dynamics.

To achieve these goals, we…

- contribute to the analysis of technical, economic, and social questions related to our field of competence, and forward the synthesized opinion of our experts to scientific, legislative, industrial and educational organizations and institutions;
- follow the national and international trends and results related to our field of competence, foster the professional and business relations between foreign and Hungarian companies and institutes;
- organize an extensive range of lectures, seminars, debates, conferences, exhibitions, company presentations, and club events in order to transfer and deploy scientific, technical and economic knowledge and skills;
- promote professional secondary and higher education and take active part in the development of professional education, teaching and training;
- establish and maintain relations with other domestic and foreign fellow associations, IEEE sister societies;
- award prizes for outstanding scientific, educational, managerial, commercial and/or societal activities and achievements in the fields of infocommunication.

Contact information

President: **FERENC VÁGUJHELYI** • *elnok@hte.hu*
Secretary-General: **ISTVÁN MARADI** • *istvan.maradi@gmail.com*
Operations Director: **PÉTER NAGY** • *nagy.peter@hte.hu*
International Affairs: **ROLLAND VIDA, PhD** • *vida@tmit.bme.hu*

Address: H-1051 Budapest, Bajcsy-Zsilinszky str. 12, HUNGARY, Room: 502
Phone: +36 1 353 1027
E-mail: *info@hte.hu*, Web: *www.hte.hu*