



HAL
open science

Toward a digital twin IoT for the validation of AI algorithms in smart-city applications

Hamza Ngadi, Ahcène Bounceur, Madani Bezoui, Laaziz Lahlou, Mohammad Hammoudeh, Kara Nadja, Laurent Tchamnda Nana

► **To cite this version:**

Hamza Ngadi, Ahcène Bounceur, Madani Bezoui, Laaziz Lahlou, Mohammad Hammoudeh, et al..
Toward a digital twin IoT for the validation of AI algorithms in smart-city applications. 6th International Conference on Machine Learning for Networking (MLN'2023), Nov 2023, Paris, France.
hal-04389064

HAL Id: hal-04389064

<https://hal.science/hal-04389064v1>

Submitted on 11 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Toward a digital twin IoT for the validation of AI algorithms in smart-city applications

Hamza Ngadi¹, Ahcene Bounceur², Madani Bezoui³, Laaziz Lahlou⁴,
Mohammad Hammoudeh², Kara Nadjia⁴, and Laurent Nana¹

¹ Lab-STICC, UBO, Brest, France
Hamza.Ngadi@univ-brest.fr

² King Fahd University of Petroleum and Minerals (KFUPM)
Information & Computer Science Department
Dhahran, Saudi Arabia

Ahcene.Bounceur; m.hammoudeh@kfupm.edu.sa

³ CESI LINEACT, UR 7527, Nice, France
mbezoui@cesi.fr

⁴ Ecole de Technologie Superieure, Montreal, Canada
laaziz.lahlou@etsmtl.ca

Abstract. The development of digital twins for road traffic has garnered significant attention within the scientific community, particularly in the realms of virtualization and the Internet of Things (IoT). The implementation of a digital twin for automobiles offers a virtual replica, capable of discerning the precise location, status, and real-time behavior of each vehicle present in the road traffic network. The primary objective of this endeavor is to create an advanced digital twin of cars that can seamlessly navigate through road traffic. To accomplish this, a meticulously selected technical approach involves employing a platform that simulates virtual sensor networks, accompanied by a purpose-built application that facilitates the access and dissemination of car-related data. Furthermore, this undertaking incorporates the utilization of an existing traffic simulator alongside a robust communication protocol to ensure seamless data transfer between the simulation environment and the sensors responsible for data collection from automobiles.

Keywords : Digital twin, Sumo simulator, Internet of Things, CupCarbon, MQTT communication protocol, Smart-city.

1 Introduction

The development of a digital twin aims to duplicate a real object or environment taking into account its behavior over time, and then project it into a virtual environment for the purpose of monitoring or collecting data from the real environment. Data collection offers a means approved by scientific research in this field, to understand and analyze the object or environment targeted, in order to create predictions and scenarios of activities, whether undesirable or not from the real source environment [2]. The main motivations for creating a digital twin in a specific domain [8], can be driven by :

- ✓ Ensuring the reliability of behaviors and actions in an environment.
- ✓ Creating scenarios to guarantee stability in an environment.
- ✓ Guarantee the confidence of interaction with an object or an environment.
- ✓ Reducing and anticipating future failures of an object or environment.

From a technical point of view, a digital twin is a model that represents a description, a behavior and a state in time, updated with the shared data of an object, or an entity to be studied [7]. Several scales of creation of a digital twin are possible, these depend on the complexity of the objects or environment to simulate.

The scales can be varied between objects or equipment, complex systems, districts or cities up to the scale of a whole country, approached in the near future by the progress of scientific research [28].

To succeed in simulating the dynamics of road traffic that consists of at least one or more cars, the most common way in technical implementations is to equip one or more cars with sensors, thus, collecting their geolocation coordinates in real road traffic via these sensors, in order to use them to create or update the coordinates of the simulated cars in an application that implements a geographical map dedicated to visualization.

The technical implementation addressed in this work aims at using a Sumo road traffic simulator [22] [3], to extract geographic data and send it via a communication protocol to an IoT simulator [23], which receives this data and updates the locations of the cars in the geographic area from which the cars are traveling. The presence of IoT sensors brings to life the digital twin of a city-related environment, providing a seamless and transparent means of exchanging data. Connected IoT objects offer the possibility for data to flow between real and virtual environments.

The collection and monitoring of data via connected sensors, give a global view of the interactions of an environment belonging to a city [23, 13]. In this case, the construction of a twin with the behavior associated with each object of the targeted environment becomes possible, to simulate and study possible scenarios. The Sumo multimodal traffic simulator used in this work allows the creation and management of a large number of traffic networks, including the modeling and simulation of roads, vehicles and pedestrians. The data extracted from this tool belong to the simulations created from an OpenStreetMap geographical map, describing the behavior of cars in a road network.

OpenStreetMap (OSM) project [10] [24] aims to build an online mapping database using open source geographic data. Common uses of this data are route planning, making available geographic data of real addresses and places, and accessing already collected data of a region, city or country in the form of a map, via map editors, libraries or APIs. The data extracted from a Sumo simulation is intended to be sent to the CupCarbon IoT simulator [5][18] for visualization of the cars in the IoT geographic map generated by this tool.

The following sections of this paper will discuss the creation of the application for data extraction, the implemented means of data sharing, and the receipt of



Fig. 1: Global process map of the developed application.

data via the CupCarbon tool for visualization of the simulated cars, as well as the potential for improvement of the presented work.

2 Traffic Extractor Application

Internet of Things (IoT) simulation within digital twins enhances the efficiency and reliability of intelligent transport systems by enabling real-time data analysis and predictive modeling for optimized traffic management [12]. To establish a comprehensive repository of geographical data intended for utilization in the CupCarbon tool, aiming to construct a digital twin of road traffic relying on Sumo simulations, the imperative lies in crafting an intermediary application. This application shall seamlessly facilitate the exchange of data between the Sumo dataset and the CupCarbon IoT platform, ensuring a smooth and coherent flow of information between these two distinct entities.

Access to the simulation file plays a pivotal role in effectively leveraging the valuable data contained within the Sumo files. In order to achieve seamless data transfer from Sumo to the CupCarbon tool, a carefully considered communication protocol is adopted. This protocol serves as the conduit through which the data is efficiently transmitted towards the CupCarbon platform. To enhance user experience and foster intuitive interactions, a user-friendly graphical interface is meticulously designed and integrated into the application. This interface empowers users with the freedom to customize and fine-tune various simulation parameters, enabling them to execute scenarios according to their specific requirements. The thoughtful incorporation of this graphical interface not only simplifies user interactions but also streamlines the process of scenario simulation, contributing to an optimized and sophisticated workflow.

2.1 Access to the scenario of a Sumo simulation: data extraction

The Sumo simulator serves as a powerful tool capable of generating simulation files that encapsulate scenario-specific parameters for our chosen road traffic setup. The tool's comprehensive documentation offers access to these files through a variety of programming languages, such as Python, C++, and Java.

In the context of this work, Java has been selected as the language of choice for developing the application. To access the simulation files effectively, Sumo provides several libraries, and for this particular application, the Libsumo library [20][16][14] is employed, replacing the TraaS library [21][11], which relies on the TraCI (Traffic Control Interface) API [1][21] for accessing road traffic data from the simulation files. This ensures seamless and efficient retrieval of the required data for further processing within the application, thereby contributing to the successful realization of the project objectives.

The Libsumo library effectively fulfills the role previously performed by the TraCI API, serving as an intermediary interface between the Java application and the simulation. Within the developed Java application, it operates as a client, diligently querying the simulation to retrieve real-time information pertaining to the current states of the cars within the road environment specified by the generated Sumo file.

The Java application comprehensively captures the simulation by targeting the specific Sumo file, and in an iterative manner, it systematically traverses each round of the loop to extract pertinent data from one or more currently simulated cars. Subsequently, this extracted information is seamlessly transmitted to the CupCarbon tool, facilitating the visualization of the cars' movements and activities.

Upon completion of the scenario iterations, the Sumo simulation file is safely released, and the simulation itself is gracefully concluded. By effectively coordinating this process, the Java application successfully synchronizes data extraction and transmission, thereby optimizing the generation of the digital twin for road traffic in CupCarbon.

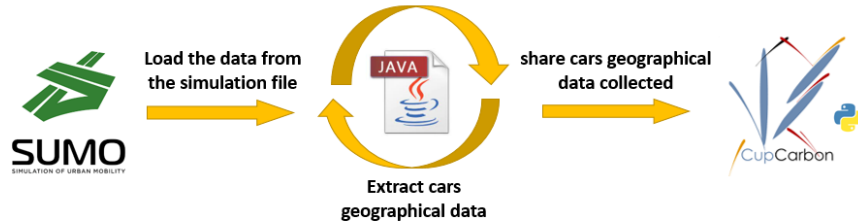


Fig. 2: The process of extracting geographical data from cars, from a Sumo simulation.

2.2 Data sharing: MQTT communication protocol

In our technical implementation, the crucial aspect of data sharing is facilitated by the MQTT protocol [25] [26]. This protocol is specifically designed for machine-to-machine communication, where relevant stakeholders subscribe to a

centralized broker to facilitate a seamless and efficient exchange of data. The decision to adopt MQTT as the communication protocol is well-founded, owing to its user-friendly nature and streamlined implementation process.

One of the primary advantages of MQTT lies in its lightweight design, ensuring minimal energy consumption [15] and bandwidth usage. This inherent efficiency makes it an ideal and well-suited choice for IoT applications, such as our digital twin system for road traffic. The utilization of MQTT significantly contributes to the optimization of data transmission, promoting a highly responsive and agile ecosystem, ultimately enhancing the overall performance and effectiveness of the digital twin implementation.

Ensuring robust communication security is paramount for IoT in digital twin simulations, as it safeguards sensitive data and critical infrastructure, mitigates cyber threats, and ensures the integrity and reliability of the simulation environment [6]. In our project's architecture, the local broker selected for implementation is Mosquitto [19][4][9]. Mosquitto aligns with the inherent lightweight nature of the MQTT protocol, and it stands out for its effortless installation process and user-friendly operation. The advantage of Mosquitto lies in its ability to maintain the efficiency and responsiveness synonymous with MQTT, while simultaneously conserving system resources.

By incorporating Mosquitto as the local broker, we ensure seamless and efficient data communication between the various components of our digital twin system for road traffic. Its stability, ease of setup, and optimal utilization of machine resources make it an apt choice for facilitating reliable data exchange, thereby enhancing the overall performance and robustness of our IoT-based simulation.

A broker serves as an intermediary between clients utilizing protocols like MQTT, ensuring the reliable delivery and reception of messages, with adjustable quality of service in different shared topics. However, public and internet-accessible brokers may encounter challenges in message transmission and reception when facing saturation.

Apart from Mosquitto, there are several other alternatives that provide access to and utilization of the MQTT protocol. These widely used alternatives include: ActiveMQ [17], Apache Kafka [27], EMQX [4][9], VerneMQ [4][9], RabbitMQ [17], and HiveMQ [4][9]. Notably, HiveMQ is recognized for its stability and is often employed for online data sharing purposes. Each of these alternatives offers distinctive features and capabilities, allowing users to choose the most suitable broker based on their specific requirements and preferences.

2.3 The GUI interface of the application

A graphical interface, integrated to the Java application, has been developed to facilitate the management of the parameters dedicated to the examples of the simulations to be executed.

The various parameters available in the graphical interface can be adjusted according to the example scenario of the cars to be simulated, the characteristics of the available field parameters are :

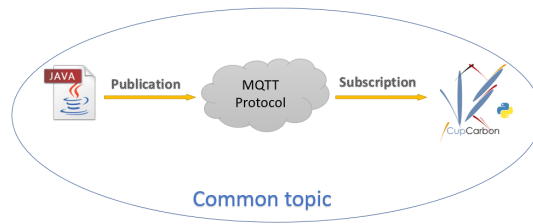


Fig. 3: General operation of the MQTT protocol.

- ✓ The possibility to choose the host server of the MQTT broker.
- ✓ The choice of the global topic for data sharing via MQTT.
- ✓ The number of cars identifiable to the MQTT topics to be generated.
- ✓ The possibility to choose between the default simulation file and another available Sumo file.
- ✓ The maximum number of iterations of a simulation.

In addition, the graphical interface offers the possibility of launching a simulation or stopping it, with two distinct buttons, dedicated to these functions. A descriptive text, located after the buttons, indicates the current state of the simulation that is being executed.

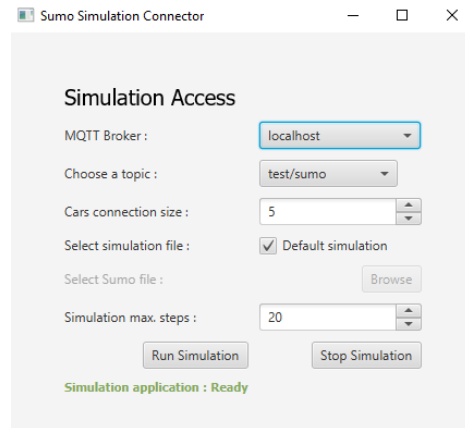


Fig. 4: Graphical interface of the application dedicated to accessing and sharing data of Sumo cars.

The possibility of choosing a maximum number of cars by the user gives us the interest to generate as many topics as cars to simulate. The objective of the creation of these topics is to assign, for each generated topic, a unique car which is in the course of simulation, thus, the data sharing of each car is ensured with their respective topics.

The CupCarbon tool chosen to process or visualize the coordinates of the cars shared via MQTT, needs to identify the geographical area, presented and covered by the simulation. The Libsumo library provides methods to retrieve and convert the boundary coordinates of the map proposed by Sumo, into latitude and longitude coordinates, which will allow it to be located in a geographical map of the CupCarbon IoT tool.

3 Digital twin construction with CupCarbon

The CupCarbon tool serves as an IoT simulator, integrating visualization capabilities through OpenStreetMap (OSM) maps to depict the geographical layout of a city. It employs both Python and the official SenScript.com language for scripting purposes.

By utilizing this tool, we gain access to a visualization interface for our MQTT-shared data. It facilitates the creation and manipulation of networks comprising nodes and virtual sensors. Furthermore, we can associate behaviors with these nodes through scripts, enabling seamless data accessibility via subscription to the relevant topic within our developed application.

The mechanism for data manipulation and treatment, as well as the subscription process, are effectively implemented to efficiently utilize the received data and provide real-time visualization of cars along with their updated coordinates.

3.1 Receiving geographical data and display of cars

The data retrieved from the topic shared with the Java application greatly facilitates data access. The Python scripts required for the retrieval and processing of the MQTT data, make the display of the geographical points of one or more cars, more dynamic as the data is extracted from the Sumo simulation, thus giving life to a digital twin of a road traffic whose source data to be collected belongs to the Sumo simulation.

The purpose of using Python in this tool is to retrieve the data emitted by the car using the MQTT protocol and then display the location of the car, in real time, on the virtual platform of CupCarbon. The data received can be saved to create complex road traffic scenarios exploring all possibilities to avoid collisions between vehicles and/or with pedestrians, or to reduce the load of road traffic in a traffic jam.

Fig. 5: Process of receiving data with the CupCarbon tool, for the display of cars with their respective coordinates.

4 Conclusion and future work

In this article, we have introduced a novel approach to creating a digital twin of a car by harnessing data from an existing traffic simulator. The core objective of

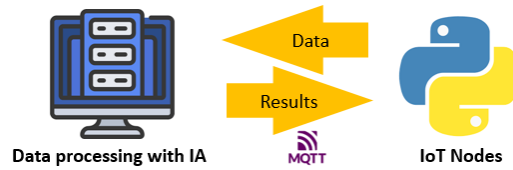


Fig. 6: Data processing integration scheme using artificial intelligence algorithms in future work.

this manoeuvre is to develop a Java application capable of extracting geographical data from simulated cars at specific time intervals. This extracted data is then seamlessly shared with the CupCarbon tool, enabling efficient processing and visualization of the available cars.

In our forthcoming research, we aim to advance our methodology by integrating artificial intelligence (AI) methods into the processing and display of cars in road traffic scenarios. From a technical perspective, we plan to implement AI and machine learning algorithms using the Python interpreter under Java. The Python interpreter provides us with the flexibility to harness the functionalities and libraries available in Python, making it an ideal choice for this work. The extensive range of stable and regularly updated libraries dedicated to AI/ML in Python, validated and supported by the scientific community and developers, further strengthens this decision.

To enable AI algorithms to efficiently process and manipulate car data, we intend to utilize the MQTT protocol for sharing the data with a supercomputer equipped with a graphical processing unit. Once the processing is done, the modified data will be sent back to the Python program. This will allow for in-depth analysis and organization of the newly acquired geographic car data.

The purpose of implementing this data processing layer is to impart intelligent and autonomous behaviors to the fleet of cars in various road traffic situations. This involves exploring ways to influence the overall behavior of the road traffic system. Through these efforts, our research endeavors to enhance traffic management strategies, promoting the evolution of smarter and more efficient road traffic systems.

References

1. Ahmed Abdo, Guoyuan Wu, and Nael Abu-Ghazaleh. Cyber-security oriented co-simulation platform for connected and autonomous driving. page 9.
2. T Alsoufi, Mohammad Hammoudeh, Zuhair Bandar, and Andy Nisbet. An overview and classification of approaches to information extraction in wireless sensor networks. In *Proceedings of the 5th International Conference on Sensor Technologies and Applications (SENSORCOMM'11)*, volume 255, 2011.
3. Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. SUMO – Simulation of Urban MObility. page 6.

4. Melvin Bender, Erkin Kirdan, Marc-Oliver Pahl, and Georg Carle. Open-Source MQTT Evaluation. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4, Las Vegas, NV, USA, January 2021. IEEE.
5. Ahcene Bounceur, Laurent Clavier, Pierre Combeau, Olivier Marc, Rodolphe Vauzelle, Arnaud Masserann, Julien Soler, Reinhardt Euler, Taha Alwajeeh, Vyas Devendra, Umber Noreen, Emilie Soret, and Massinissa Lounis. CupCarbon: A new platform for the design, simulation and 2D/3D visualization of radio propagation and interferences in IoT networks. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4, Las Vegas, NV, January 2018. IEEE.
6. Andrew Carlin, Mohammad Hammoudeh, and Omar Aldabbas. Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. *International Journal of Advanced Computer Science and Applications*, 6(6), 2015.
7. Gregory Epiphaniou, Mohammad Hammoudeh, Hu Yuan, Carsten Maple, and Uchenna Ani. Digital twins in cyber effects modelling of iot/cps points of low resilience. *Simulation Modelling Practice and Theory*, 125:102744, 2023.
8. Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*, 8:108952–108971, 2020.
9. Sten Gruener, Heiko Koziolok, and Julius Ruckert. Towards Resilient IoT Messaging: An Experience Report Analyzing MQTT Brokers. In *2021 IEEE 18th International Conference on Software Architecture (ICSA)*, pages 69–79, Stuttgart, Germany, March 2021. IEEE.
10. I. Alihan Hadimlioglu and Scott A. King. City Maker: Reconstruction of Cities from OpenStreetMap Data for Environmental Visualization and Simulations. *ISPRS International Journal of Geo-Information*, 8(7):298, July 2019.
11. Abeer Hakeem, Reza Curtmola, Xiaoning Ding, and Cristian Borcea. DFPS: A Distributed Mobile System for Free Parking Assignment. *IEEE Transactions on Mobile Computing*, pages 1–1, 2021.
12. Mohammad Hammoudeh, Gregory Epiphaniou, Sana Belguith, Devrim Unal, Bamidele Adebisi, Thar Baker, ASM Kayes, and Paul Watters. A service-oriented approach for sensing in the internet of things: Intelligent transportation systems and privacy use cases. *IEEE Sensors Journal*, 21(14):15753–15761, 2020.
13. Mohammad Hammoudeh and Robert Newman. Information extraction from sensor networks using the watershed transform algorithm. *Information Fusion*, 22:39–49, 2015.
14. Fin Malte Heuer. Scenario Generation for Testing of Automated Driving Functions based on Real Data. page 119.
15. Sara Holm and Mohammad Hammoudeh. A comparative analysis of iot protocols for resource constraint devices and networks. In *Proceedings of the 6th International Conference on Future Networks & Distributed Systems*, pages 616–625, 2022.
16. Thanapapas Horsuwan and Chaodit Aswakul. Reinforcement Learning Agent under Partial Observability for Traffic Light Control in Presence of Gridlocks. In *SUMO User Conference 2019*, pages 29–9.
17. Valeriu Manuel Ionescu. The analysis of the performance of RabbitMQ and ActiveMQ. In *2015 14th RoEduNet International Conference - Networking in Education and Research (RoEduNet NER)*, pages 132–137, Craiova, Romania, September 2015. IEEE.

18. Cristina Lopez-Pavon, Sandra Sendra, and Juan F. Valenzuela-Valdes. Evaluation of CupCarbon Network Simulator for Wireless Sensor Networks. *Network Protocols and Algorithms*, 10(2):1, June 2018.
19. Paul Macheso, Tiwonge D. Manda, Sylvester Chisale, Nelson Dzupire, Justice Mlatho, and Didacienne Mukanyiligira. Design of ESP8266 Smart Home Using MQTT and Node-RED. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 502–505, Coimbatore, India, March 2021. IEEE.
20. Arthur Müller, Vishal Rangras, Georg Schnittker, Michael Waldmann, Maxim Friesen, Tobias Ferfers, Lukas Schreckenber, Florian Hufen, Jürgen Jasperneite, and Marco Wiering. Towards Real-World Deployment of Reinforcement Learning for Traffic Signal Control, January 2022.
21. Cristina Olaverri-Monreal, Javier Errea-Moreno, Alberto Díaz-Álvarez, Carlos Biurrún-Quel, Luis Serrano-Arriezu, and Markus Kuba. Connection of the SUMO Microscopic Traffic Simulator and the Unity 3D Game Engine to Evaluate V2X Communication-Based Systems. *Sensors*, 18(12):4399, December 2018.
22. Mustapha Saidallah, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. A Comparative Study of Urban Road Traffic Simulators. *MATEC Web of Conferences*, 81:05002, 2016.
23. Teodora Sanislav, George Dan Mois, and Silviu Folea. Digital Twins in the Internet of Things Context. In *2021 29th Telecommunications Forum (TELFOR)*, pages 1–4, Belgrade, Serbia, November 2021. IEEE.
24. Guclu SENYURDUSEV, Ahmet Ozgur DOGRU, and N. Necla ULUGTEKIN. Exploring the opportunities of open source data use in creation 3d procedural city models. 1:9, 2020.
25. Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. Performance evaluation of MQTT and CoAP via a common middleware. In *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, Singapore, April 2014. IEEE.
26. Nguyen Quoc Uy and Vu Hoai Nam. A comparison of AMQP and MQTT protocols for Internet of Things. In *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 292–297, Hanoi, Vietnam, December 2019. IEEE.
27. Shubham Vyas, Rajesh Kumar Tyagi, Charu Jain, and Shashank Sahu. Performance Evaluation of Apache Kafka – A Modern Platform for Real Time Data Streaming. In *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, pages 465–470, Gautam Buddha Nagar, India, February 2022. IEEE.
28. Shuzhen Zou, Xiaoxiao Tao, Biaohong Tao, and Guoxu Wu. A Preliminary Study on the Development and Application of Digital Twin Landscape Architectures in the Context of Smart City. In *2022 Global Conference on Robotics, Artificial Intelligence and Information Technology (GCRAIT)*, pages 322–327, Chicago, IL, USA, July 2022. IEEE.