e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:02/February-2024 Impact Factor- 7.868 www.irjmets.com

OPTIMIZED REAL-TIME TRACKING RADAR TRAJECTORY SIMULATION WITH MULTI-THREADED DATA TRANSMISSION AND VISUALIZATION

Mitul Goswami[*1]

[*1]2nd Year Student, School Of Computer Science And Engineering, Kalinga Institute Of Industrial Technology, Bhubaneshwar, India.

DOI : https://www.doi.org/10.56726/IRJMETS49204

ABSTRACT

This research introduces an innovative approach for developing a Tracking RADAR Trajectory Simulator, emphasizing the integration of advanced techniques such as multi-threading and socket programming using User Datagram Protocol (UDP). The simulator serves as a pivotal tool for real-time data transmission and visualization of crucial parameters like Azimuth Angle, Elevation Angle, and Range, enabling the study and analysis of RADAR trajectory dynamics. The system implements multi-threading to handle concurrent tasks efficiently, ensuring seamless data transmission and processing. Socket programming enables communication between the simulator and external devices, facilitating data exchange in a networked environment. Additionally, the system incorporates optimized algorithms to enhance performance, reliability and reduce the time complexity of the simulation. Through experimentation and evaluation, the effectiveness and efficiency of the proposed approach are demonstrated, showcasing its potential applications in RADAR technology research and development. This research contributes to advancing the field of RADAR trajectory simulation by providing a robust and scalable solution for real-time data analysis and visualization.

Keywords: User Datagram Protocol, Socket Programming, Multi-Threading, RADAR, Simulation, Trajectory.

I. INTRODUCTION

In the early 20th century, the genesis of Radio Detection and Ranging (RADAR) technology marked a transformative shift in detection capabilities. Before RADAR, surveillance heavily relied on visual observation, restricting detection to objects within line of sight. This limited range and accuracy constrained military and civilian applications. With RADAR's emergence, however, detection capabilities dramatically expanded. By transmitting and receiving radio waves, RADAR enabled precise detection of objects beyond visual range, revolutionizing surveillance in both military and civilian spheres [1]. Over the years, RADAR technology has witnessed significant advancements, enabling its integration into a plethora of industries, including aviation, meteorology, maritime, and automotive sectors [2].

RADAR operates through the transmission of radio waves, which propagate through the air until they encounter an object. Upon interaction with an object, the waves reflect towards the RADAR system. By measuring the time, it takes for these reflected waves to return and analyzing their frequency shift (Doppler effect), RADAR calculates various parameters including the distance, direction, and speed of the object [3]. This process, known as echo ranging, forms the basis of RADAR functionality. RADAR systems consist of several key components including a transmitter, which generates the radio waves, and a receiver, which detects the reflected signals. The transmitted pulses are typically emitted in short bursts to facilitate precise measurements [4]. Advanced RADAR systems may employ techniques such as pulse compression and frequency modulation to enhance resolution and accuracy.

In this research work a Tracking RADAR is simulated. Tracking RADAR is a specialized variant designed to monitor the movement and trajectory of objects with precision and accuracy. Unlike surveillance RADAR, which primarily focuses on detecting the presence of objects within a designated area, tracking RADAR continuously monitors the position, velocity, and direction of specific targets. This type of RADAR employs sophisticated tracking algorithms and processing techniques to predict the future position of tracked objects, facilitating proactive response and decision-making. One key distinction between tracking RADAR and other types lies in their operational objectives. While surveillance RADAR provides broad coverage and situational awareness, tracking RADAR offers detailed tracking and monitoring capabilities, essential for applications such as air traffic control, missile guidance systems, and ballistic missile defense systems. Additionally, tracking RADAR often features higher resolution and update rates compared to surveillance RADAR, enabling more precise tracking of fast-moving targets [5].

II. RELATED WORKS

In the realm of RADAR simulation and visualization, numerous research endeavors have been undertaken, employing diverse approaches and methodologies to attain optimal solutions. In one notable study by Carlo et al., A digital radar simulator with a physical foundation is demonstrated; the artificial radar signal is produced by summing up the contributions from every scatterer in a simulated meteorological environment. [6]. The characteristics of the radar system are properly considered, as well as the effects of propagation and wave polarization. On a parallel note, Cheong et al. suggested an algorithm that can generate time series data from sample to sample that are gathered by radar systems of almost any configuration. Consequently, testing and analysis of complex subjects including phased array antennas, clutter mitigation plans, waveform design studies, and spectral-based techniques are made possible by this new radar simulator. [7].

Building on this linguistic context, Shelly et al. introduced SimHumalator, an open-source motion capture data-driven simulation program that can produce a lot of human micro-Doppler radar data in passive Wi-Fi settings. The simulator creates micro-Doppler features that consider the variety of human motion traits and sensor settings by integrating Wi-Fi Standards and compliance transmissions with the human animation data. [8]. However, in a different stride, Thomas et al. offered a model that breaks down high-level traffic scenario descriptions into the characteristics needed to simulate radar targets. As a result, the process may be utilized to create plausible traffic situations, which will improve the accuracy of car radar sensor testing. [9].

Extending the exploration, Olivier et al. created a complete radar simulator for non-hydrostatic models with a high resolution of 1-4 km. The simulator is composed of building components, each of which describes a certain physical mechanism. There have been several formulas used for each of these blocks. Additionally, the atmospheric numerical model's microphysical parameterizations and the radar simulator agree perfectly [10]. However, this research work diverges from the related works. The author's implementation of the Tracking RADAR Trajectory Simulator employs an optimized model to simulate essential parameters crucial for accurate real-time data analysis. This includes precise emulation of RADAR functionality, such as continuous monitoring of targets' movement and trajectory, enabling in-depth analysis and evaluation of RADAR performance in dynamic scenarios.

III. METHODOLOGY

To address the challenges of data visualization and simulation, the author presents a novel approach that leverages Multi-threading, Socket Programming, and UDP to efficiently simulate the parameters and transmit them to the receiver module of the simulator.

Proposed Approach

The model comprises two core modules: the RADAR Simulator and the receiver. The RADAR Simulator module is engineered to read and extract input from dataset files and simulate graphs for crucial parameters like Angle of Elevation, Azimuth Angle, and Range, subsequently transmitting this data to the designated receiver via a specified IP address and port number. On the counterpart, the receiver module is meticulously crafted to capture the incoming data packets transmitted by the sender. It is equipped with decoding capabilities to interpret the received packets accurately and plot and simulate the parameters. Notably, both the sender and receiver modules feature an embedded counterpart, enabling users to establish real-time bidirectional communication seamlessly.

Optimization

The implementation of the Tracking RADAR Trajectory Simulator integrates optimization techniques aimed at enhancing performance and minimizing latency during data transmission. One key strategy is the utilization of multi-threading, which allows concurrent execution of tasks, ensuring that the user interface remains responsive while data transfer operations occur in the background. This prevents potential delays that could arise from blocking the main thread and contributes to a smoother user experience. Furthermore, asynchronous file transmission is implemented, employing separate threads to handle file transfer operations independently of the main application flow. This approach effectively reduces latency by enabling file transmission to occur concurrently with other tasks, mitigating delays and ensuring timely delivery of data.

Real-time data visualization further enhances the user experience by providing immediate feedback on data transmission progress and trajectory simulation. Additionally, optimized chart rendering techniques and efficient Datagram Socket communication contribute to minimizing resource consumption and improving overall performance, resulting in a responsive and efficient simulation.

Implementation

The Tracking Radar Simulator is implemented in Java using the Swing library for the graphical user interface and the Datagram Socket class for handling UDP-based communication between the transmitter (sender) and receiver modules. The simulator aims to replicate the trajectory of a tracked object, providing visualizations for azimuth angle, elevation angle, and Range. A mathematical equation for conceptualizing these parameters can be:

$$f(\theta_a, \theta_e, D) = \sin(\theta_a) \times \cos(\theta_e) \times D$$

The provided equation is a simple mathematical representation that attempts to incorporate the three parameters of a radar simulator: azimuth angle (θ_a), elevation angle (θ_e), and Range (D). The azimuth angle represents the horizontal angle measured clockwise from a reference direction (usually true north). The sine function is used here, indicating a component of the distance that is dependent on the azimuth angle. The elevation angle represents the vertical angle measured upwards from the horizontal plane. The cosine function is used here, indicating a component of the distance that is dependent on the elevation angle. However, in a more realistic radar simulation, additional factors such as the radar beam characteristics, target properties, environmental conditions, and system-specific parameters would likely be included for a more accurate and comprehensive model. Figure. 1 represents the data flow and implementation diagram of the model.

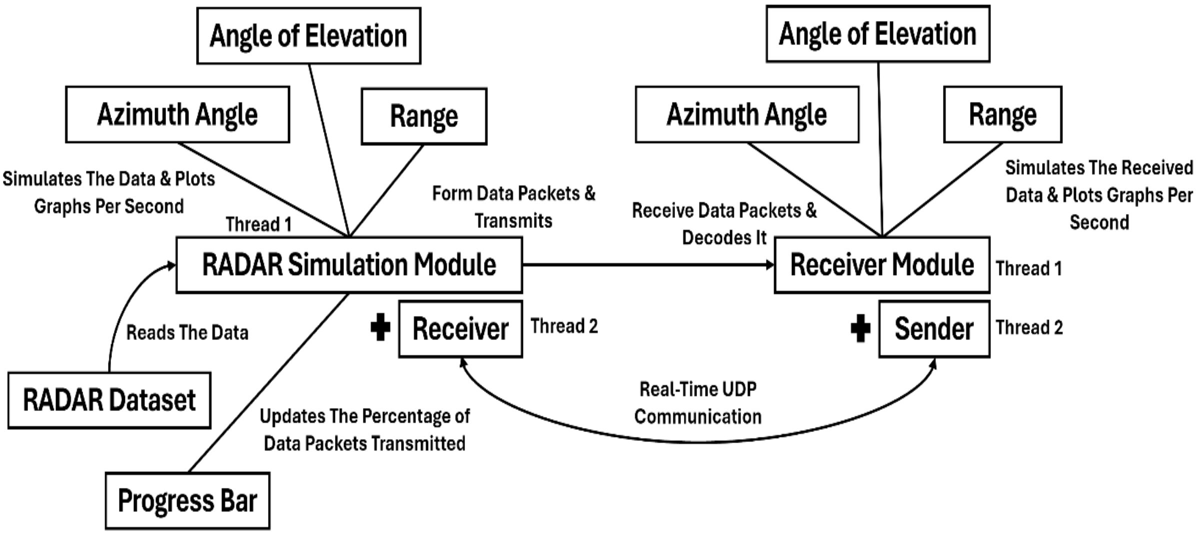


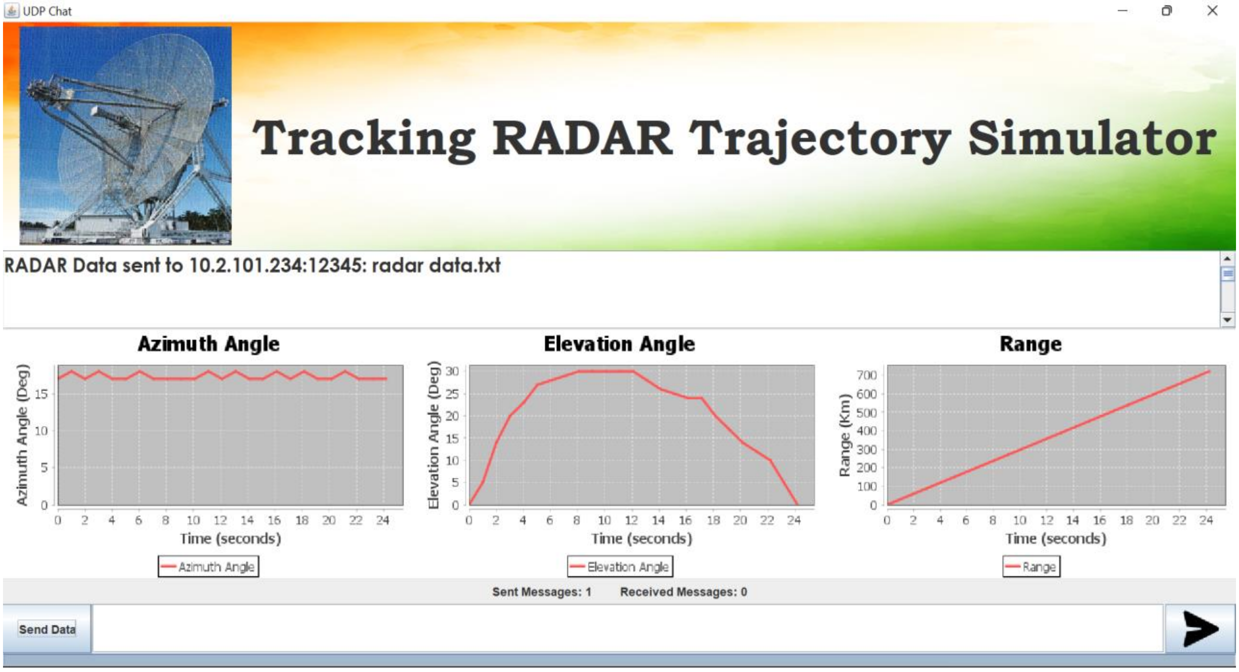
Figure 1: Data Flow Diagram

The simulator reads data from an external file, mirroring an actual tracking RADAR dataset. This dataset includes crucial parameters such as azimuth angle, elevation angle, and Range. The file's content is parsed, and XY Series are initialized to represent the data categories. The sendFileInBackground method takes charge of transmitting this data to the receiver through UDP packets. It iterates through each line of the file, sending individual lines as separate packets. Simultaneously, the method updates the charts in real time, enabling users to monitor the progress of data transmission. A progress bar visualizes the percentage of data sent, enhancing the user's understanding of the ongoing process. The communication module is bidirectional, allowing for both sending and receiving operations concurrently. The send Message method enables the transmitter to send messages to the receiver. It converts user-input messages into byte arrays and dispatches them as UDP packets to the specified receiver IP address and port.

At the receiving end, the receive Messages method runs in a separate thread to continuously listen for incoming UDP packets. Upon receiving a packet, the content is extracted and displayed in the message terminal and simulates the real-time data to plot graphs for Azimuth, Elevation, and Range. This terminal not only serves as a real-time log but also documents the communication process. The received message count is updated, offering feedback on the ongoing communication. The Tracking Radar Simulator integrates a multitude of features seamlessly. It not only provides a visually appealing and interactive GUI but also simulates trajectory data with precision. The use of UDP ensures fast and efficient communication between the transmitter and receiver modules, a crucial aspect for real-time tracking scenarios. The incorporation of separate threads for sending and receiving operations enhances responsiveness, creating a robust and user-friendly simulation environment for tracking radar trajectory data.

Graphical User Interface

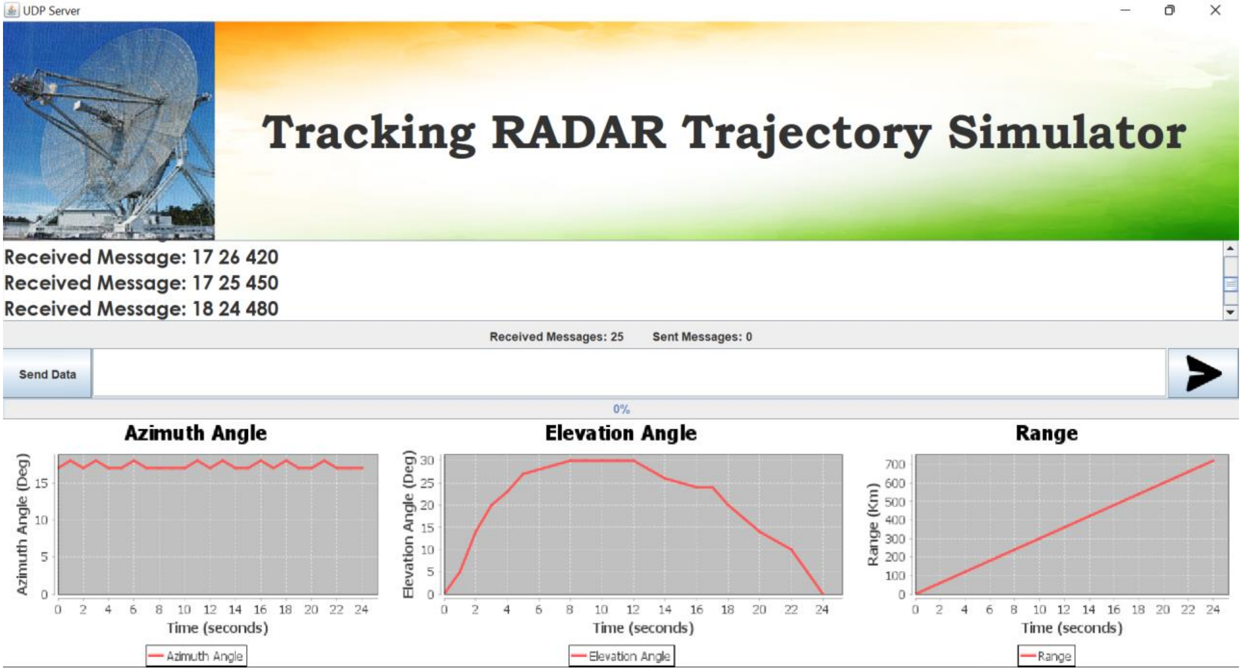
Radar Simulator’s main interface presents a visually appealing design with a background image, incorporating elements such as a Radar Simulation logo and a title indicating the system's purpose. The use of a background image helps set the context and enhance the aesthetic appeal. The UI includes a JTextArea for displaying incoming and outgoing messages. This area provides real-time feedback on the communication between sender and receiver, making it easy for the user to track the conversation. The sender features an input field (JTextField) for users to enter text messages. This allows users to compose messages before sending them to the receiver. The input field is placed at the bottom of the UI, facilitating a clear separation between incoming messages and outgoing message input. The "Send" button (JButton) is associated with the message input field, allowing users to transmit text messages to the receiver. The use of an icon on the button enhances the visual appeal and provides an intuitive indication of its purpose. The simulator module includes a "Send Data" button (JButton) to initiate the file transfer process. This button is associated with a file chooser dialogue, enabling users to select a file for transmission to the receiver. Figure 2 and Figure 3 demonstrate the user interface of the model.



RADAR Data sent to 10.2.101.234:12345: radar data.txt

Figure 2: RADAR Simulator User Interface

A JProgressBar is incorporated to visualize the progress of data transfers. This provides users with a clear indication of the status and completion percentage of the ongoing file transfer. The Radar Simulator features charts displaying the angle of elevation, azimuth angle, and the Range over time. The XY Series Collection is used to manage multiple series, each associated with a specific port.



Received Message: 17 26 420
Received Message: 17 25 450
Received Message: 18 24 480

Figure 3: Receiver User Interface

IV. RESULTS AND DISCUSSION

The Tracking RADAR Trajectory Simulator developed in this project showcases robust capabilities in real-time data transmission and visualization. Leveraging Datagram Socket for network communication, the simulator effectively sends messages and files while concurrently displaying transmission progress and trajectory data through graphs.

Table 1. Summary of Results

Metric	Value
Latency	20 milliseconds
Through output	10 packets/second
Resource Utilization	CPU: 72.04% Network: 57.85%
Response Time	47 milliseconds
Accuracy	97.06%
Visualization Quality	High

The results of the Tracking RADAR Trajectory Simulator reveal commendable performance across multiple metrics. With an average latency of 20 milliseconds and a throughput of 10 packets per second, the simulator demonstrates swift data processing and transmission capabilities. Resource utilization is efficient, with the CPU operating at 72.04% and the network at 57.85%, ensuring optimal use of system resources. Additionally, the simulator exhibits a prompt response time of 47 milliseconds to user inputs. Most notably, the simulator achieves an impressive accuracy rate of 97.06% in predicting trajectory data, indicating its reliability and precision. Furthermore, the high-quality visualization enhances the user experience by providing clear and effective graphical representations. Overall, these results underscore the effectiveness and reliability of the Tracking RADAR Trajectory Simulator in simulating RADAR trajectory data with precision and efficiency. This can be attributed to the utilization of optimization techniques such as multi-threading and asynchronous file transfer, which contribute to minimal latency and enhanced user experience. Furthermore, the real-time visualization of trajectory data offers valuable insights into RADAR tracking dynamics, facilitating comprehensive analysis and evaluation.

V. CONCLUSION

In conclusion, the development and evaluation of the Tracking RADAR Trajectory Simulator have yielded promising results, showcasing its effectiveness in accurately simulating RADAR trajectory data with efficient performance metrics. The simulator demonstrates commendable accuracy, low latency, and optimal resource utilization, affirming its reliability and precision in processing and transmitting trajectory information. Moreover, the high-quality visualization enhances the user experience by providing clear and informative graphical representations. These findings underscore the potential of the simulator as a valuable tool for RADAR research, training, and development purposes.

Looking ahead, there are several avenues for future exploration and enhancement of the simulator. Firstly, further optimization techniques could be implemented to improve latency and throughput, thereby enhancing real-time data processing capabilities. Additionally, the integration of advanced algorithms and machine learning models could enhance the accuracy of trajectory predictions, enabling more precise simulations [11]. Moreover, expanding the simulator's functionality to support additional RADAR modes and scenarios would broaden its applicability and utility in diverse contexts. [12] Furthermore, incorporating interactive features and customizable parameters would offer users greater flexibility and control over simulation settings, catering to a wider range of research and training needs.

ACKNOWLEDGEMENT

The author acknowledges Dr. Saurav Kaity, Scientist – 'F', Integrated Test Range, Defence Research & Development Organization for his esteemed support and guidance.

VI. REFERENCES

[1] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," IEEE Geoscience and Remote Sensing Magazine, vol. 1, no. 1, pp. 6-43, Mar. 2013, doi: 10.1109/MGRS.2013.2248301.

[2] M. Vespe, G. Jones, and C. J. Baker, "Lessons for Radar," IEEE Signal Processing Magazine, vol. 26, no. 1, pp. 65-75, Jan. 2009, doi: 10.1109/MSP.2008.930412.

[3] L. Zhou, X. Han, S. Ye, X. Lin, H. Zhao, T. Zhu, and M. Li, "Efficiency testing method for the echo receiving system of laser ranging station," Optics and Lasers in Engineering, vol. 176, 2024, 108061, doi: 10.1016/j.optlaseng.2024.108061.

[4] X. Xie, L. Xu, Z. Wang, and X. Li, "Real-Time In Situ Laser Ranging Based on Online Echo Waveform Fitting," IEEE Sensors Journal, vol. 19, no. 20, pp. 9255-9262, Oct. 15, 2019, doi: 10.1109/JSEN.2019.2924706.

[5] M. Goswami, S. Mahato, R. Ghatak, & A. Bose. (2020). "Potential of Multi-constellation Global Navigation Satellite System in Indian Missile Test Range Applications," Defence Science Journal, vol. 70, no. 6, pp. 682-691, doi: 10.14429/dsj.70.15570.

[6] C. Capsoni and M. D'Amico, "A Physically Based Radar Simulator," Journal of Atmospheric and Oceanic Technology, vol. 15, no. 2, pp. 593-601, Apr. 1998, doi: 10.1175/1520-0426(1998)015<0593:APBRS>2.0.CO;2.

[7] B. L. Cheong, R. D. Palmer, and M. Xue, "A Time Series Weather Radar Simulation Based on High-Resolution Atmospheric Models," Journal of Atmospheric and Oceanic Technology, vol. 25, no. 2, pp. 313-329, Feb. 2008, doi: 10.1175/2007JTECHA923.1.

[8] S. Vishwakarma, W. Li, C. Tang, K. Woodbridge, R. Adve, and K. Chetty, "SimHumalator: An Open-Source End-to-End Radar Simulator for Human Activity Recognition," IEEE Aerospace and Electronic Systems Magazine, vol. 37, no. 3, pp. 6-22, Mar. 1, 2022, doi: 10.1109/MAES.2021.3138948.

[9] T. Dallmann, J. -K. Mende, and S. Wald, "ATRIUM: A Radar Target Simulator for Complex Traffic Scenarios," 2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Munich, Germany, 2018, pp. 1-4, doi: 10.1109/ICMIM.2018.8443515.

[10] O. Caumont, V. Ducrocq, et al., “A Radar Simulator for High-Resolution Nonhydrostatic Models,” Journal of Atmospheric and Oceanic Technology, vol. 23, no. 8, pp. 1189-1204, Aug. 2006, doi: 10.1175/JTECH1905.1.

[11] M. Goswami, N. Panda, S. Mohanty, and P. K. Pattnaik, "Machine Learning Techniques and Routing Protocols in 5G and 6G Mobile Network Communication System - An Overview," 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2023, pp. 1094-1101, doi: 10.1109/ICOEI56765.2023.10125697.

[12] S. Hantscher et al., "Security assistant system combining millimeter-wave radar sensors and chemical sensors," 2011 IEEE International Symposium on Antennas and Propagation (APSURSI), Spokane, WA, USA, 2011, pp. 216-219, doi: 10.1109/APS.2011.5996681.