Lecture Notes in Electrical Engineering 1104

N. R. Shetty
N. H. Prasad
N. Nalini *Editors*

Advances in Computing and Information

Proceedings of ERCICA 2023, Volume 1

Springer

Lecture Notes in Electrical Engineering

Volume 1104

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

N. R. Shetty · N. H. Prasad · N. Nalini
Editors

Advances in Computing and Information

Proceedings of ERCICA 2023, Volume 1

Springer

Editors
N. R. Shetty
Central University of Karnataka
Kalaburagi, Karnataka, India

N. H. Prasad
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India

N. Nalini
Nitte Meenakshi Institute of Technology
Bengaluru, Karnataka, India

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-99-7621-8 ISBN 978-981-99-7622-5 (eBook)
https://doi.org/10.1007/978-981-99-7622-5

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The Eighth International Conference on "Emerging Research in Computing, Information, Communication and Applications," ERCICA 2023, is an annual event organized at the Nitte Meenakshi Institute of Technology (NMIT), Yelahanka, Bangalore, India, in association with Laghu Udyog Bharati (LUB-Karnataka) and technology partner National Research Development Corporation, Ministry of Science and Technology, Govt. of India.

ERCICA aims to provide an interdisciplinary forum for discussion among researchers, engineers, and scientists to promote research and exchange of knowledge in computing, information, communications and related applications. This conference will provide a platform for networking of academicians, engineers and scientists and also will encourage the participants to undertake high-end research in the above thrust areas.

ERCICA 2023 received more than 400+ papers from all over the world, viz. from China, UK, Africa, Saudi Arabia and India. The ERCICA Technical Review Committee has followed all necessary steps to screen more than 400 papers by going through six rounds of quality checks on each paper before selection for presentation/publication in Springer LNEE (Lecture Notes in Electrical Engineering) proceedings, which is SCOPUS indexed.

The acceptance ratio is 1:4

Kalaburagi, India	N. R. Shetty
Bengaluru, India	N. H. Prasad
Bengaluru, India	N. Nalini
February 2023	

Contents

About the Editors

N. R. Shetty is Chancellor of Central University of Karnataka, Kalaburagi, and Chairman of the Review Commission for the State Private University Karnataka. He is currently serving as Advisor to the Nitte Meenakshi Institute of Technology (NMIT), Bangalore. He is also Founder Vice-President of the International Federation of Engineering Education Societies (IFEES), Washington DC, USA. He served as Vice Chancellor of Bangalore University for two terms and President of the ISTE, New Delhi three terms. He was also Member of the Executive Committee of the AICTE and Chairman of its South West Region Committee.

N. H. Prasad is currently working as Professor and Head of the Department of Master of Computer Applications at Nitte Meenakshi Institute of Technology, Bengaluru. He completed his Ph.D. at Jawaharlal Nehru University, New Delhi, India. He has more than 18 years of experience in different roles in both public and private sector enterprises, including the Ministry of Human Resource and Development, New Delhi, Government of India. He has received the prestigious "Dr. Abdul Kalam Life Time Achievement Award" and also received a "Young Faculty" award at the 2nd Academic Brilliance Awards.

N. Nalini is currently working as Professor at the Department of Computer Science and Engineering at Nitte Meenakshi Institute of Technology, Bangalore. She received her M.S. from BITS, Pilani, in 1999, and her Ph.D. from Visvesvaraya Technological University in 2007. She has more than 17 years of teaching and 10 years of research experience. She has several international publications to her credit and has received the "Bharath Jyothi Award" by Dr. Bhishma Narain Singh, Former Governor of Tamil Nadu and Assam, given out by the India International Friendship Society. She is Lifetime Member of the ISTE, CSI, ACEEE and IIFS.

2D Mapping and Exploration Using Autonomous Robot

N. Shravan, M. Manoj Kumar, Sriraag Jayanth, R. S. Bindu, B. R. Madhu, and K. S. Sreekeshava

Abstract A Light Detection and Ranging (LIDAR) system is a very useful tool in the exploration of sparse environments, such as underground tunnels and during disasters. LIDAR can also be used to measure distances from the sensor to an object by illuminating that object with a laser light and then measuring time for the reflected light to return. The LIDAR is a device that emits and measures light in the form of a laser which reflects from the ground to the lidar. The aim of our research is to make use of LIDAR technology and explore unknown environments with autonomous navigation at ease. In our research, we used LIDAR and made a robot that can generate a 2D map of the surrounding environment and can help the operator to analyse the interior part of it. We have used Raspberry Pi to pre-process the data from the LIDAR and used Robot Operating System (ROS) to interpret the data on the graph to generate the map. This system is almost human independent and is risk free. Hence, generated 2D map is used for further process to inspect if the environment is safe for human to enter. In this research, we have also implemented the autonomous navigation part. Once the map is generated, the robot can move from one point to another point taking the shortest path. Suppose there is new obstacle in the path, the robot is capable of generating the next feasible alternative path to reach the destination Manoj Kumar and Nandakumar (Int J Grid Distrib Comput 13:1622–1627, 2020). SLAM technique is used to predict the position of the robot and simultaneously navigate. In this paper, we have explained in detail about how the data is processed and the functionality.

N. Shravan (✉) · M. Manoj Kumar · S. Jayanth · R. S. Bindu
Department of CSE, Jyothy Institute of Technology, Bangalore, India
e-mail: shravan.nanjunda@gmail.com

M. Manoj Kumar
e-mail: manoj.kumar@jyothyit.ac.in

B. R. Madhu
Department of AI and ML, Jyothy Institute of Technology, Bangalore, India
e-mail: madhu.br@jyothyit.ac.in

K. S. Sreekeshava
Department of Civil Engineering, Jyothy Institute of Technology, Bangalore, India
e-mail: sreekeshava.ks@jyothyit.ac.in

N. R. Shetty et al. (eds.), *Advances in Computing and Information*, Lecture Notes in Electrical Engineering 1104, https://doi.org/10.1007/978-981-99-7622-5_1

Keywords LIDAR (light detection and ranging) · Robot operating system (ROS) · Autonomous · Navigation · SLAM (simultaneous localization and mapping) · Sparse environments

1 Introduction

A robot needs some sensors like camera or obstacle sensor to navigate. For autonomous navigation, we need some computing node which can process several possible routes and choose to travel in the best route possible. The perception sensor that we have used in our robot is LIDAR. Using this we can get odometry data without using encoders for the motor. Odometry gives the distance of travel which is very important for navigation.

Lidar consists of several components like the laser, optics, photodetector, receiver and Inertial Measurement Unit (IMU). It's a good replacement of camera and Global Positioning System (GPS). It uses light in the form of UV or near infrared to measure distance and other properties of the target within its range. It can sense a wide range of materials including uneven objects like rocks. A narrow laser beam is used to map physical features with high resolution. Sophisticated LIDAR can see through objects such as walls or tress and can create a high-resolution digital elevation model with vertical accuracy with up to 10 centimetres. The light source from the LIDAR hits the object, and the reflected light is detected and measured by sensors. As we know the speed of light which is $3x10^8$ m/s, using the time taken by the laser light to travel back to the receiver the distance can be calculated. The advantage of LIDAR is that it provides high accuracy, fast acquisition and processing with minimum human dependence and is independent of weather and light conditions. But it requires frequent calibration to get high precession.

The main intention is to provide a low-cost reliable solution for mapping and exploration. Lidar is the best feasible solution when it comes to underground and sparse environments where GPS and GPRS signals don't penetrate and due to low lighting conditions using camera will be infeasible. LIDAR sensors are usually used in cleaning robots, evacuation robots, inspection robots, etc.

Robotics has become an increasingly popular topic in recent years due to advancements in technology that have made it possible for robots to do things like go into dangerous environments or aid humans with jobs that are difficult or not safe for humans to do. A robot is a machine or device that performs a complex task for which it has been programmed. The term is also used to describe a type of fiction in which the robot can do anything and the only limitation is that it conforms to the limits of its programming. NASA and other space agencies use them for exploring undersea life, deep sea exploration and Mars exploration. Also, mine exploration can also be done using robots. Sometimes it's risky to send humans to extract minerals; in such cases, robots can perform the task.

Robot Operating System (ROS) is a collection of software, libraries, frameworks and tools used to build various robot applications. It provides functionality for hardware abstraction, device drivers, communication between processes over multiple machines, tools for testing and visualization and much more. The aim of this project is to provide an overview on how to work with ROS capable robots in the real world. It allows developers to assemble a complex system by integrating existing ones. It also provides a platform to simulate and test the functionality before practically working on it. The key feature of this software is the simple approach to connect devices by establishing publish and subscribe technique. It also allows to develop a highly complex software without any knowledge on the hardware part. It provides a platform to connect to a central node where commutations can be performed. Multiple devices can be connected to the same node, and the communication network is simple. Communication can be created by issuing requestable services or by establishing publish and subscribe concept with other nodes. Both methods communicate via specified message types. Peer-to-peer communication is mostly managed by ROS itself. Our goal is to build software for an on-board computer that is Raspberry Pi that allows us to remotely control and monitor a robot, connected to us via Wi-Fi, by using a keyboard on our computer and get the feed from the LIDAR mounted on the robot.

2 Proposed System

The proposed robot system consists of three subsystems; they are sensors, motors and compute node. For the navigation system, we considered three options which are GPS, LIDAR and camera. After considering the advantages and disadvantages, we considered that LIDAR will be the best option. LIDAR is easy to use, cost-effective and does not depend on the lighting conditions and weather conditions.

The robot consists of LIDAR of 8m range which makes up the perception part. The Raspberry Pi 4B+ 8 gb single-board computer (SBC) makes up the computing part. Motors are connected to the L298N 2A H-bridge motor driver which takes up the locomotion part. The power to the whole system is supplied by a li-ion battery pack of 11.1 V 2500 mah. The Raspberry Pi requires 5 V regulated constant power supply; hence, we need to use a buck converter which regulates the 12–5 V. The interconnecting between all the components is shown in Figure 1. The robot and computer communicate with each other via Wi-Fi.

The robot is the master and computer act as slave. The robot publishes and computer subscribes to get the data from the robot. The computer will have ROS software to visualize, interpret and plot the data from the LIDAR sensor.

After few design iterations, we decided how the robot will be and started making a 3D design of it, so we could get an idea how the robot will be in reality and where to place all the components. We used Autodesk Fusion 360 student edition software to design the robot as shown in Figure 2.

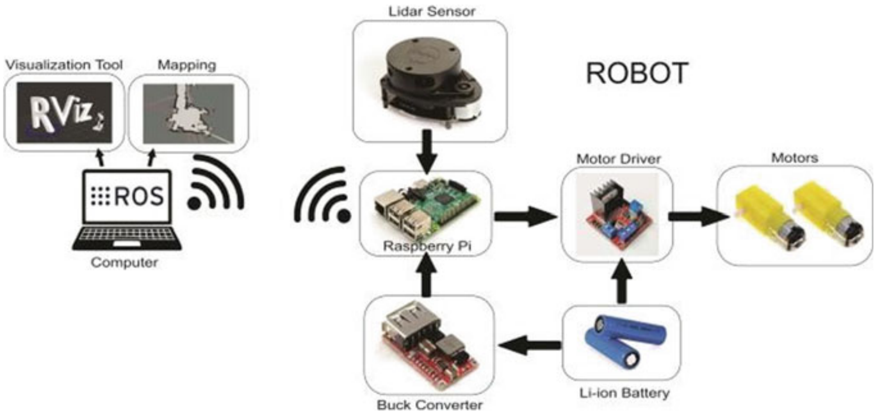
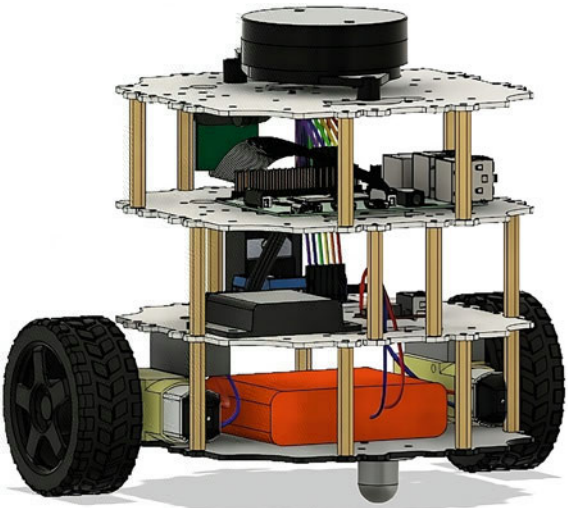


Fig. 1 Interconnection of components

Fig. 2 Robot 3D design in CAD software



Before building the actual prototype model of our robot, we wanted to make sure that it really works as we think so we thought of simulating the project as shown in Figure 3 and explore about Gazebo. It's a popular simulation software for robotics development. It provides a platform to accurately and effectivity simulate populations of robot in complex environments. It's similar to Unity3D or Unreal Engine, but with this we can integrate with ROS. Moreover, advanced features like controlling and data acquisition from the robot can also be interfaced.

After tinkering with design of the robot and simulation, we decided to source the components and try it out practically [8]. So, we bought all the necessary components and make a custom acrylic laser cut chassis and got some 3D-printed parts which were

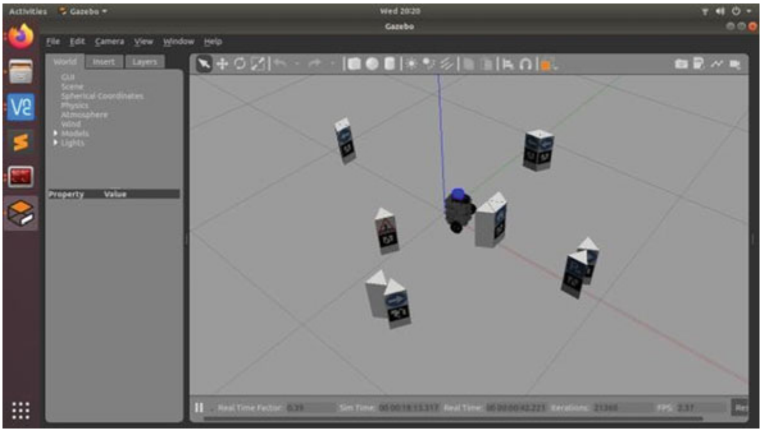


Fig. 3 Robot simulation in gazebo

necessary to build the frame of the robot. After assembling we started interfacing with the LIDAR sensor. We studied in detail about it and took references from the datasheet provided by the manufacturer and developed a simple Python program which can stream the raw LIDAR data as shown in Figure 4.

We also tried to plot the LIDAR data on a graph using matplotlib library as shown in Figure 5. Using this we were able to visualize how the LIDAR can sense the surrounding objects.

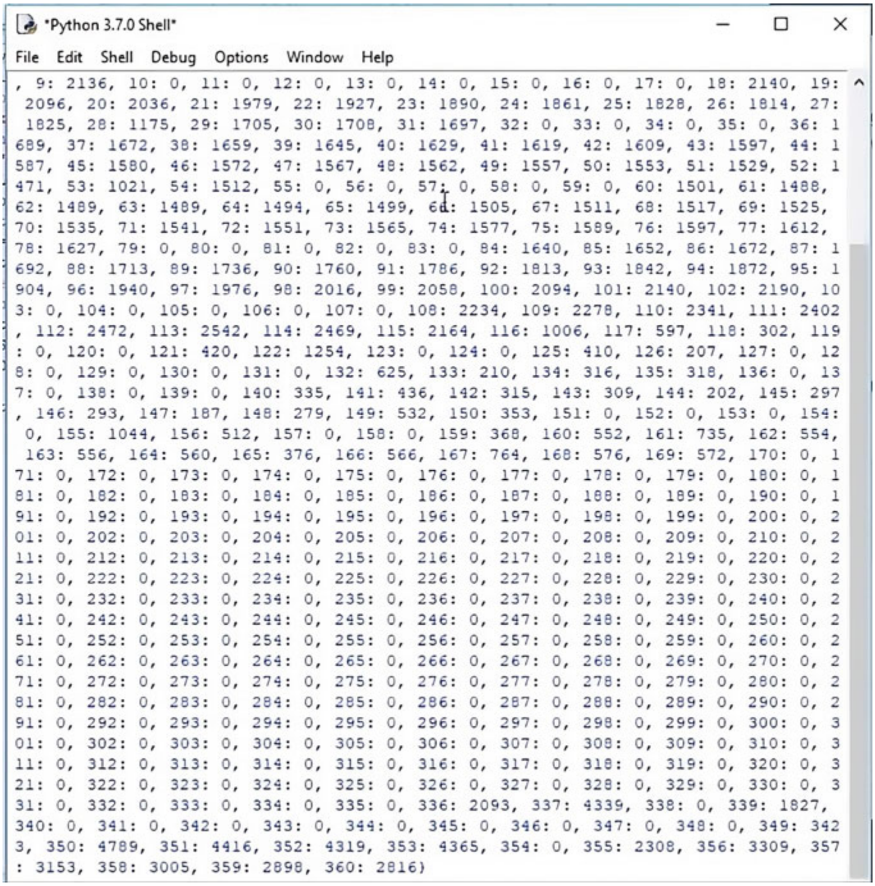
Later we developed a program using ROS using which the robot was able to communicate to the computer via Wi-Fi by steaming the pre-processed LIDAR data. The navigation commands can be sent to the robot using the arrow keys on the keyboard. This way the map of the environment can be generated and saved. After that the robot can move to the specified location in the environment autonomously choosing the shortest path. If any obstacle is detected in its path and is not able to proceed to the destination in the calculated path, then it can find the next feasible alternative path to reach the destination (Fig. 6).

We conducted some experiments at different environments; the Figure 7 represents the point cloud inside a cave, and Figure 8 represents point cloud inside a tunnel.

3 Data Processing

The raw data from LIDAR contains noise, and it has to be filtered before processing it. Hence, the pre-processing is performed by Raspberry Pi on the robot, and it follows the approach mentioned in the Fig. 9.

Raw data from the LIDAR is taken, and the outliers are removed, and if any null values are obtained, then it represents that its open space and the robot can go there to discover the environment. After this the classification can be done by continuous



```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
, 9: 2136, 10: 0, 11: 0, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 2140, 19:
 2096, 20: 2036, 21: 1979, 22: 1927, 23: 1890, 24: 1861, 25: 1828, 26: 1814, 27:
 1825, 28: 1175, 29: 1705, 30: 1708, 31: 1697, 32: 0, 33: 0, 34: 0, 35: 0, 36: 1
689, 37: 1672, 38: 1659, 39: 1645, 40: 1629, 41: 1619, 42: 1609, 43: 1597, 44: 1
587, 45: 1580, 46: 1572, 47: 1567, 48: 1562, 49: 1557, 50: 1553, 51: 1529, 52: 1
471, 53: 1021, 54: 1512, 55: 0, 56: 0, 57: 0, 58: 0, 59: 0, 60: 1501, 61: 1488,
62: 1489, 63: 1489, 64: 1494, 65: 1499, 66: 1505, 67: 1511, 68: 1517, 69: 1525,
70: 1535, 71: 1541, 72: 1551, 73: 1565, 74: 1577, 75: 1589, 76: 1597, 77: 1612,
78: 1627, 79: 0, 80: 0, 81: 0, 82: 0, 83: 0, 84: 1640, 85: 1652, 86: 1672, 87: 1
692, 88: 1713, 89: 1736, 90: 1760, 91: 1786, 92: 1813, 93: 1842, 94: 1872, 95: 1
904, 96: 1940, 97: 1976, 98: 2016, 99: 2058, 100: 2094, 101: 2140, 102: 2190, 10
3: 0, 104: 0, 105: 0, 106: 0, 107: 0, 108: 2234, 109: 2278, 110: 2341, 111: 2402
, 112: 2472, 113: 2542, 114: 2469, 115: 2164, 116: 1006, 117: 597, 118: 302, 119
: 0, 120: 0, 121: 420, 122: 1254, 123: 0, 124: 0, 125: 410, 126: 207, 127: 0, 12
8: 0, 129: 0, 130: 0, 131: 0, 132: 625, 133: 210, 134: 316, 135: 318, 136: 0, 13
7: 0, 138: 0, 139: 0, 140: 335, 141: 436, 142: 315, 143: 309, 144: 202, 145: 297
, 146: 293, 147: 187, 148: 279, 149: 532, 150: 353, 151: 0, 152: 0, 153: 0, 154:
 0, 155: 1044, 156: 512, 157: 0, 158: 0, 159: 368, 160: 552, 161: 735, 162: 554,
 163: 556, 164: 560, 165: 376, 166: 566, 167: 764, 168: 576, 169: 572, 170: 0, 1
71: 0, 172: 0, 173: 0, 174: 0, 175: 0, 176: 0, 177: 0, 178: 0, 179: 0, 180: 0, 1
81: 0, 182: 0, 183: 0, 184: 0, 185: 0, 186: 0, 187: 0, 188: 0, 189: 0, 190: 0, 1
91: 0, 192: 0, 193: 0, 194: 0, 195: 0, 196: 0, 197: 0, 198: 0, 199: 0, 200: 0, 2
01: 0, 202: 0, 203: 0, 204: 0, 205: 0, 206: 0, 207: 0, 208: 0, 209: 0, 210: 0, 2
11: 0, 212: 0, 213: 0, 214: 0, 215: 0, 216: 0, 217: 0, 218: 0, 219: 0, 220: 0, 2
21: 0, 222: 0, 223: 0, 224: 0, 225: 0, 226: 0, 227: 0, 228: 0, 229: 0, 230: 0, 2
31: 0, 232: 0, 233: 0, 234: 0, 235: 0, 236: 0, 237: 0, 238: 0, 239: 0, 240: 0, 2
41: 0, 242: 0, 243: 0, 244: 0, 245: 0, 246: 0, 247: 0, 248: 0, 249: 0, 250: 0, 2
51: 0, 252: 0, 253: 0, 254: 0, 255: 0, 256: 0, 257: 0, 258: 0, 259: 0, 260: 0, 2
61: 0, 262: 0, 263: 0, 264: 0, 265: 0, 266: 0, 267: 0, 268: 0, 269: 0, 270: 0, 2
71: 0, 272: 0, 273: 0, 274: 0, 275: 0, 276: 0, 277: 0, 278: 0, 279: 0, 280: 0, 2
81: 0, 282: 0, 283: 0, 284: 0, 285: 0, 286: 0, 287: 0, 288: 0, 289: 0, 290: 0, 2
91: 0, 292: 0, 293: 0, 294: 0, 295: 0, 296: 0, 297: 0, 298: 0, 299: 0, 300: 0, 3
01: 0, 302: 0, 303: 0, 304: 0, 305: 0, 306: 0, 307: 0, 308: 0, 309: 0, 310: 0, 3
11: 0, 312: 0, 313: 0, 314: 0, 315: 0, 316: 0, 317: 0, 318: 0, 319: 0, 320: 0, 3
21: 0, 322: 0, 323: 0, 324: 0, 325: 0, 326: 0, 327: 0, 328: 0, 329: 0, 330: 0, 3
31: 0, 332: 0, 333: 0, 334: 0, 335: 0, 336: 2093, 337: 4339, 338: 0, 339: 1827,
340: 0, 341: 0, 342: 0, 343: 0, 344: 0, 345: 0, 346: 0, 347: 0, 348: 0, 349: 342
3, 350: 4789, 351: 4416, 352: 4319, 353: 4365, 354: 0, 355: 2308, 356: 3309, 357
: 3153, 358: 3005, 359: 2898, 360: 2816}
```

Fig. 4 Raw LIDAR data

data readings. Later the point cloud's resolution can be increased by LIDAR intensity normalization and resampling with different radii [4]. The distance of an obstacle from two different points can give precise measurements. If the projected light does not return the LIDAR, then it's considered that there is no obstacle and the robot can move there to analyse the environment. Proceeding that the data can be subjected to regression analysis and reflectance adjustment which can then determine the optimal radius and accuracy.

The robot perception, obstacle measurement, pose estimation, feature extraction and mapping are explained in the Fig. 10.

First the LIDAR is initialized, and it starts gathering the data from the surround obstacles in the environment. The readings are then analysed by the computer, and then matching takes place. If the result is true, then the graph will be refined and if any temporary obstacle is detected on the map, then it will slowly disappear. The resolution of the graph will be increased. But if the matching turns out to be false,

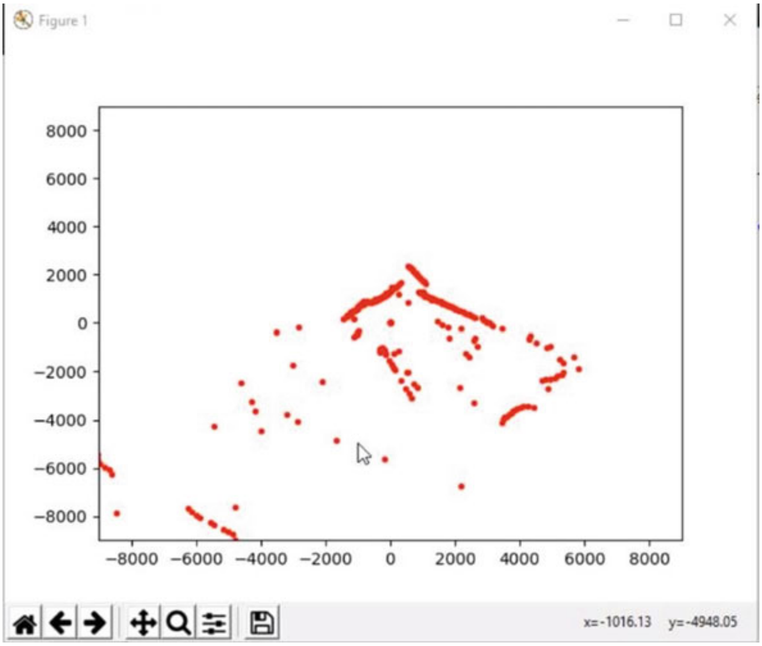


Fig. 5 LIDAR data plotted on a graph

Fig. 6 Robot prototype

then it’s classified based on the observation. If its unexpected observation, then it is ignored as offensive features. If it’s expected observation then the new features will be added to the map and the ROS will help to stich the map and generate a large-scale map.

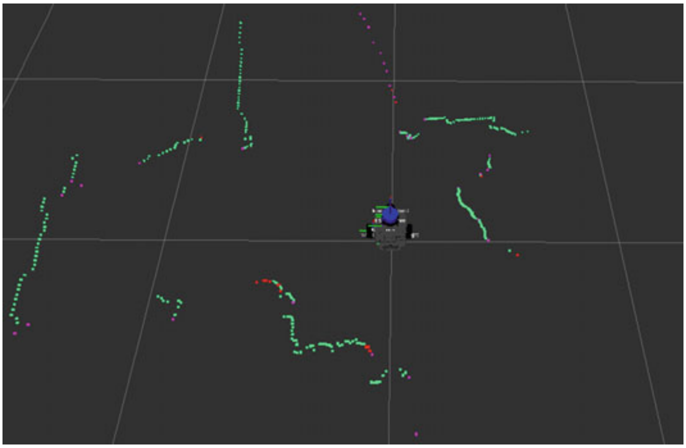


Fig. 7 Point cloud inside a cave

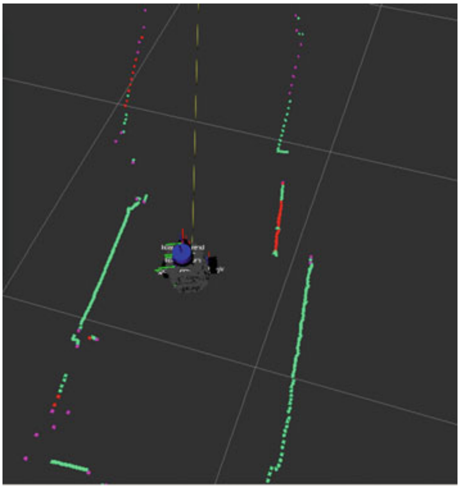


Fig. 8 Point cloud inside a tunnel

Navigation is the science of determining the position of an object which is moving to the destination. The robot navigation and path planning process are described in Fig. 11.

First the perception sensor and motor driver get initialized, after that the first waypoint is given to the robot. If the current position itself is the first waypoint, the control will proceed else it will reach the first waypoint and then the control will proceed. The current robot facing direction is noted, and the LIDAR data is taken as reference and is checked if the next waypoint is reached. If not, then the navigation route is calculated and the motors are controlled to navigate the robot to the waypoint. When the robot reaches the waypoint, the control checks if there

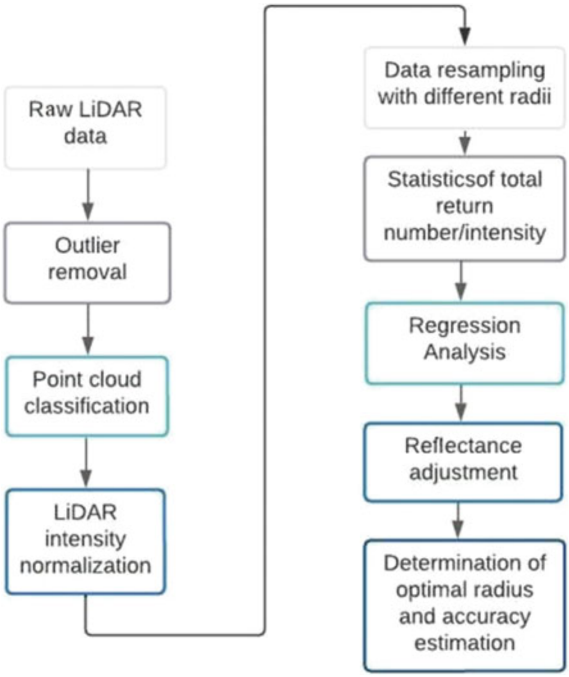


Fig. 9 LIDAR data pre-processing

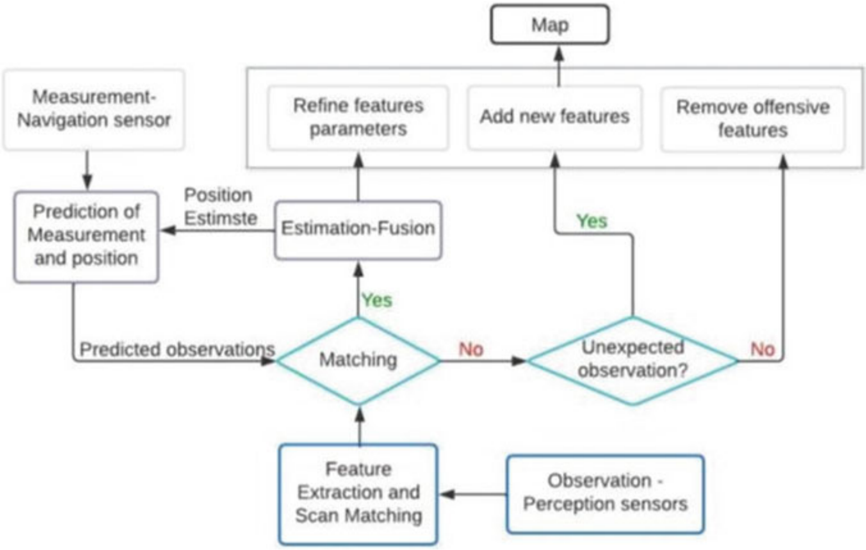


Fig. 10 Robot working block diagram

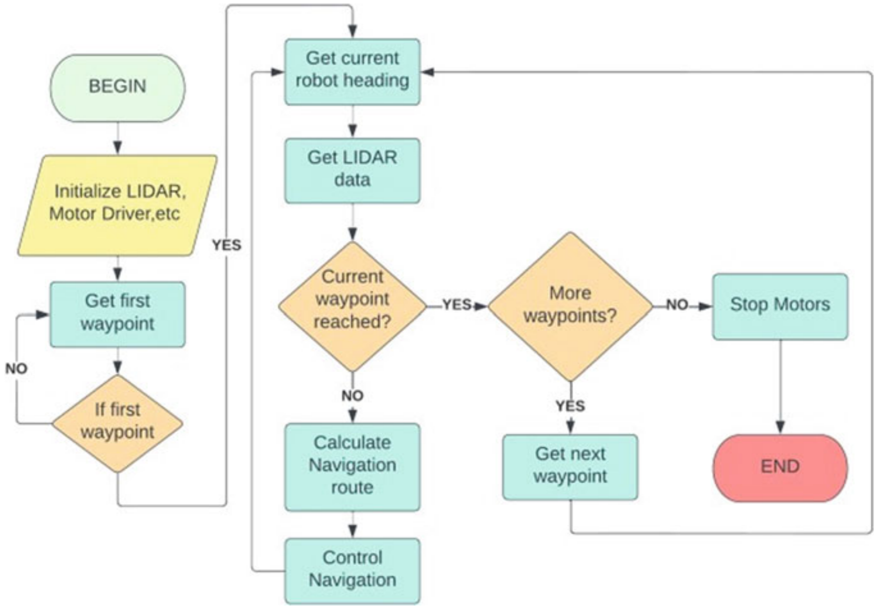


Fig. 11 Robot working block diagram

are any other waypoints. If no other waypoints are there, then the motors will stop rotating and the program will end. If there are more waypoints, the previous waypoint is considered as temporary waypoint and again control loops from calculating robot heading and gathering LIDAR data and navigating the robot until it reaches the destination.

4 Mathematical Analysis

Mathematical analysis is important in research to understand the basic working principle. It helps to modularize the process to easily achieve the desired target. Figure 12 describes how the robot calculates the angle it has to turn to reach its waypoint or destination given.

- α = Heading Angle
- β = Current Heading
- γ = Target Heading
- d = Distance

$$\boxed{\alpha = \text{Target Heading} - \text{Current Heading}}$$

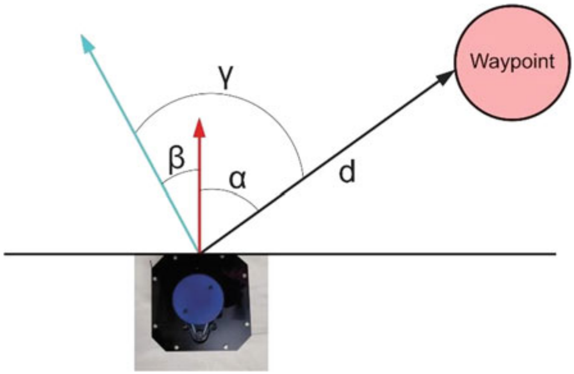


Fig. 12 How the robot reaches the waypoint

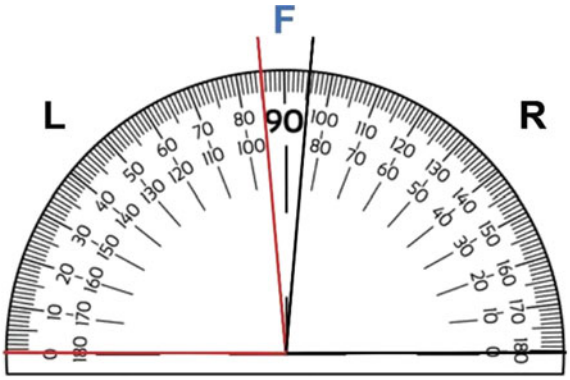


Fig. 13 Classification of direction based on angle

Based on the heading angle, the robot can decide if it has to either take left or right turn or move forward. The classification is represented in Figure 13.

If the heading angle (α) is between − 180° and − 5° then it's considered that the robot should turn left. If α is between + 5° and + 180° then it's considered that the robot should turn right. Else if the α is between −5° and + 5° then it's considered that the robot should move forward.

5 Results

See (Figs. 14 and 15).

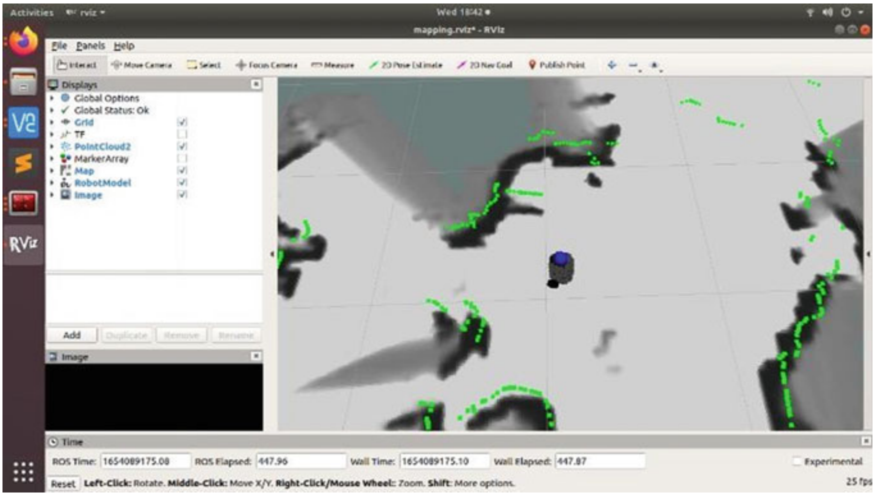


Fig. 14 Robot mapping in space environments

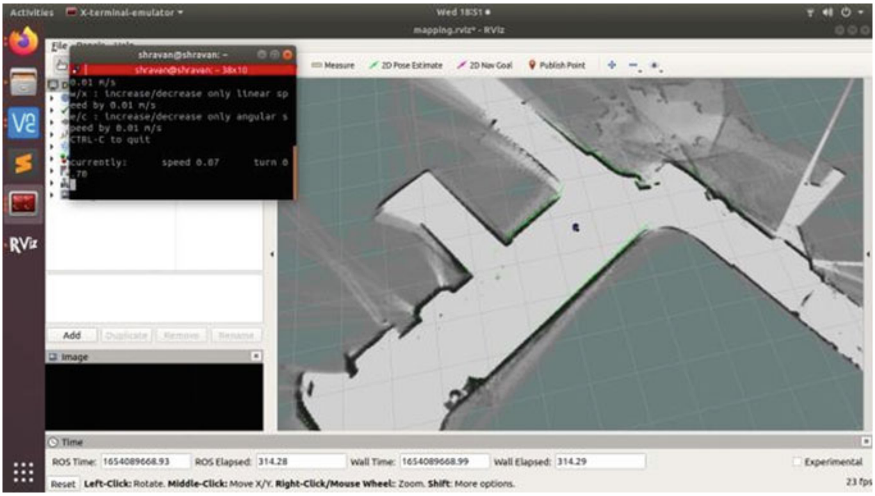


Fig. 15 Robot mapping in underground tunnel

6 Conclusion

Using LIDAR sensor, we explored about ROS and were able to build a prototype robot which can generate map of any critical environment. This is built in such a way that it is almost human independent. During emergency situations like earthquake when underground structure collapses, it's very risky for human to enter inside. Instead, this robot can be sent to analyse or for rescue operation. The robot can explore the surrounding using the LIDAR perception sensor and in future the LIDAR which we are using can be upgraded to a 3D LIDAR. This opens new possibilities and capabilities that this robot can perform. During the research, we learnt about the LIDAR technology, ROS, 3D designing, simulation, autonomous navigation and other aspects.

References

1. Barbosa AW, Bianchi RE, Romero RA (2005) Control of mobile robots via internet. In: Proceedings of COBEM–18th international congress of mechanical engineering, novovember,(cd-rom), Ouro Preto, Minas Gerais, Brazil
2. Manoj Kumar M, Nandakumar AN (2020) Conceptual multilateral security framework for authentication of cloud computing services. Int J Grid Distrib Comput 13(1):1622–1627, ISSN: 2005-4262, Web of Science
3. Siegwart R (2011) Illah Reza Nourbakhsh, and Davide Scaramuzza. MIT press, Introduction to autonomous mobile robots
4. Manoj Kumar M, Nandakumar AN, Comprehension of the architectural entropy in the cloud computing ecosystem. J Crit Rev, ISSN 2394-5125. https://doi.org/10.31838/jcr.07.18.198, Scopus indexed
5. Vanitha M, Selvalakshmi M, Selvarasu R (2016) Monitoring and controlling of mobile robot via internet through raspberry Pi board. In: 2016 second international conference on science technology engineering and management (ICONSTEM). IEEE, pp 462–466
6. Dawadee A, Chahl J, Nandagopal N (2015) A method for autonomous navigation of uavs using landmarks. In: Proceedings of the 16th Australian aerospace congress, p 022015
7. Al-khawaldah M, Nüchter A (2012) Multi-robot exploration and mapping with a rotating 3D scanner. IFAC Proceedings 45(22):313–318
8. Manoj Kumar M, Nandakumar AN (2017) Exploring multilateral cloud computing security architectural design debt in terms of technical debt. In: Smart innovation, systems, and technologies, (SCI) 2017, vol 2. Springer Publications, Scopus indexed
9. Cho YK (2019) UAV-UGV cooperative 3D environmental mapping. ASCE Int Conf Comput Civ Eng
10. Nex F, Remondino F (2013) UAV for 3D mapping applications: a review. Appl Geomatics 6(1):1–15. https://doi.org/10.1007/S12518-013-0120-X
11. Ahmad A (2011) Digital mapping using low altitude UAV. Pertanika J Sci Technol 19(S):51–58
12. Rivai M, Hutabarat D, Nafis ZM (n.d.) 2D mapping using omni-directional mobile robot equipped with LiDAR. TELKOMNIKA Telecommun Comput Electron Control 18(3):1467. https://doi.org/10.12928/TELKOMNIKA.V18I3.14872

Prediction of Fake Twitters Using AdaBoost-Based Neuro-Evolution of Augmenting Topologies Algorithm

V. Suhasini and N. Vimala

Abstract Knowledge dissemination had never before been hampered in the history of humanity until the World Wide Web's development and the rapid adoption of social media outlets. As a result of the growing usage of social media platforms, fake news is increasingly common in all kinds of circumstances. After the internet evolved, most of the people are utilizing Internet for their personal purpose only at the same time they are uncontrolled to read many of fake news, also. Automated classification of a text article as real or fake is a challenging task. In this situation, to detect such types of fake news and to provide well verified news to our society, the machine learning (ML) techniques such as support vector machine, linear regression, K-nearest neighbor, neuro-evolution of augmenting topologies (NEAT) and boosting NEAT are applied in this research. After preprocesses over the actual dataset methods effectively identify the fake news with collected dataset and evaluated by the metrics such as accuracy, precision, recall and F1-score.

Keywords NEAT · AdaBoost · Twitter dataset · Fake news

1 Introduction

Social media for news is unclear. People consume news via social media due to its low cost, ease of access and speedy dissemination. On the other hand, it aids the propagation of "false news." Fake news may harm people and society. Detecting fake news on social media has been a popular study topic [1]. This paper focuses on ML classification algorithms for fake news detection in social media, as well as related research areas, unresolved challenges and future research routes (Twitter data). The classification algorithms used in the present work are Naïve Bayes (NB), K-nearest neighbors (KNN), logistic regression (LR), random forest (RF), neuro-evolution of augmenting topologies (NEAT) and boosting-based NEAT. In addition, this work

V. Suhasini (✉) · N. Vimala
Department of Computer Science, L.R.G Government Arts College (W), Tirupur, India
e-mail: suhasinikanishk2811@gmail.com

N. R. Shetty et al. (eds.), *Advances in Computing and Information*, Lecture Notes in Electrical Engineering 1104, https://doi.org/10.1007/978-981-99-7622-5_2

tested several measures with given data in order to a priori determine the expected performance of the compared classifiers.

While considering the fake news, it is always providing a wrong message to the society. This should be properly detected. Few organizations are involved with confirming author with their quality of news in order to publish the same. Those are following the manual detection about the fake, but huge numbers of data cannot follow manually to accept or else remove the same. So, an automatic system should be required to qualify the news content. The proposed method possible creates a method to detect about a specific news is comes under fake or original with respect to its words, phrases, corresponding sources with title.

Contribution of the Research Work: The state-of-the-art ML approaches, solutions and validation with data from online social networks are presented in this study. This paper analyzes the essential advantages and disadvantages of the various existing methodologies on a large amount of Twitter data in order to pinpoint the current and upcoming trends of the suggested research. This paper also offers solution for uncertain problems by proposed a new technique that is based on NEAT method.

2 Literature Review

Granik et al. [2] used a naive Bayesian classifier to detect bogus news. This method uses Facebook news post data. They claim 74% accuracy. This model's accuracy is good, but many other works use better classifiers [2]. Hadeer Ahmed et al. [3] proposed a false news recognition process using n-gram and ML approaches. Experiments demonstrate that features extraction yields the greatest results (TF-IDF). The LSVM classifier gives 92% accuracy. This model employs LSVM for two linearly separated classes [3].

Florian Sauvageau et al. [4] explained how social media users can verify information. They also detail the validation methods and the participation of journalists, scholars and official entities. People who read this material are more likely to question the veracity of social media news and hold selective beliefs [4]. A new public dataset of authentic news articles was created by Vivek Singh et al. [5], who also recommended a text-processing-based machine learning solution with an accuracy rate of 87 percent. The text's emergent sentiments, not its content, are the subject of this piece [5]. The detection of fake news may be done automatically using a new dataset called LIAR. Political NLP research, position categorization, argument mining, issue modeling, and rumor identification can all be done using this corpus. In this industry, this is the norm. Unlike the others, which also feature information from other industries, this one is only political stuff.

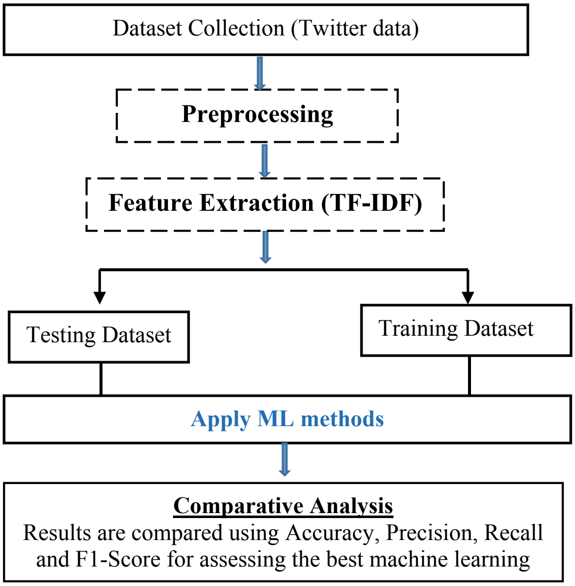


Fig. 1 Steps in recommended system

3 System Overview

The proposed system is implemented on Python program and used a convenient interface. In this system, twitter dataset are used for the experimental results. The proposed model can be described in below figure.

A Python script utilizing the Scikit-learn package was created to analyze a given dataset using ML classification techniques, which successfully distinguished between "genuine" and "fake" data. One of the most widely used programming languages for scientific computing is Python, which has a sizable scientific library [6] (Fig. 1).

4 ML Algorithms for Classification

ML play a vital role in classification. This section will discuss the some ML classification algorithms such as NB [10], KNN [11], LR [12] and RF [13] and demonstrate all classification algorithm's characteristics and working methodology.

Neuro-Evolution of Augmenting Topologies (NEAT): This section describes the NEAT algorithm. NEAT is comparable to the evolution of organisms' genomes, where genomes symbolize neural networks [7].

Figure 2 shows an NEAT network. W-indicates connection weight. Neuro-evolution searches for network topologies using evolutionary algorithms (EAs). EA optimizes key parameters of networks, specifically neuron weights and interconnections, in each stage, called a generation (see Fig. 3a).

NEAT variations can contain different NNs. Randomly selected structures classify NNs. Each generation has subpopulations with comparable structure (i.e., the way that the nodes are connected to each other). Different classes have different-sized genomes. Each class develops in accordance with its level of fitness while learning NN weights. NN weights are learned without back propagation. Figure 4 demonstrates NEAT's major steps [7].

Node Genes

Node 1	Node 2	Node 3	Node 4	Node 5
Input	Input	Input	Output	Hidden

Connection Genes

In1	In3	In2	In5	In1	In4
Out4	Out4	Out5	Out4	Out5	Out5
w1	w2	w3	w4	w6	w7

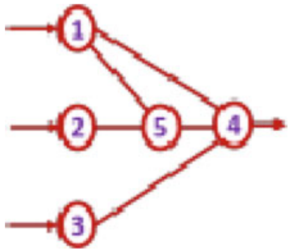


Fig. 2 A typical network in NEAT

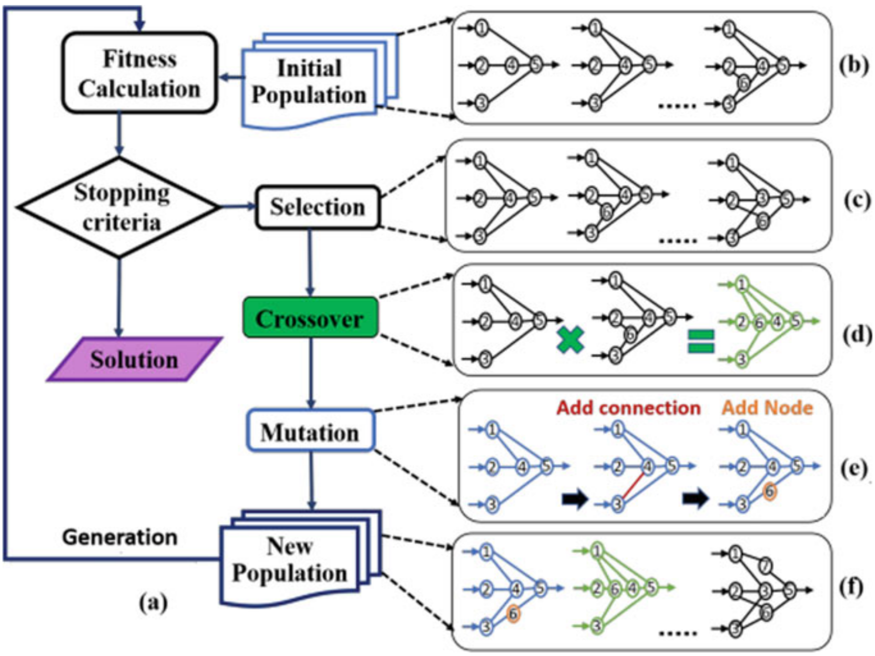


Fig. 3 General overview of the NEAT

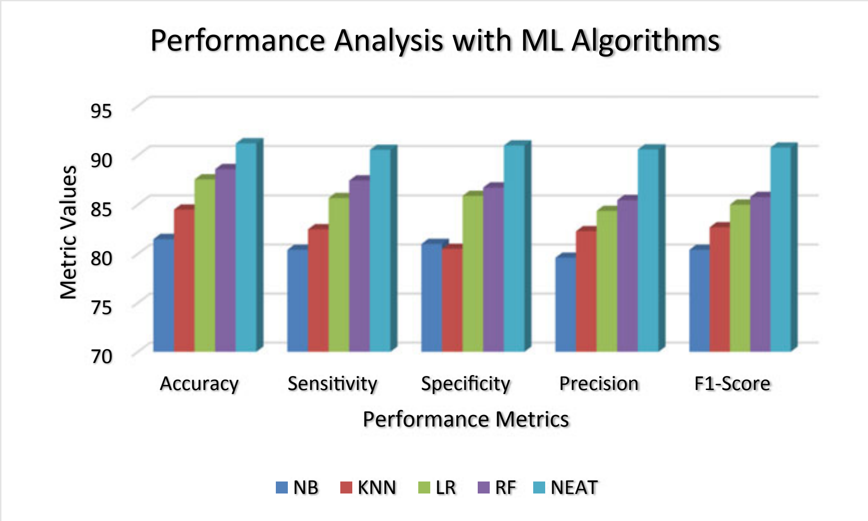


Fig. 4 Performance analysis with ML algorithms

Initial population: NEAT involves a population of neural networks, referred to as genomes, represented by "N^j" to signify the layer "*j*." In this context, "*t*" and "*i*" denote the source and the specific neural network index within the classes. In the initial generation ($t = 1$), N total neural networks span all species, with each species comprising Ns networks. To simplify explanation while maintaining generality, we'll focus on one class's learning/evolution phase. Each genome's input features is denoted as "*X*," and its output class labels as "*Y*." Evolution begins with basic neural networks. In the first generation, each network within classes comprises a single hidden layer ($j = 1$). Layer outputs are computed using the input "*X*," the weight matrix "*Wj*" between the *j*th and $(j + 1)^{\text{th}}$ layers and the bias vector "$bj + 1$" for the $(j + 1)$-th layer (where $1 \leq j \leq \ell$). This involves applying the function "*f*" to the weights, biases and inputs, generating output for the $(j + 1)$th layer as "$Nj + 1 = f(jWj + bj + 1)$." Here, "*f*" represents both the estimated label and the activation function [8]. Through this process, NEAT facilitates the evolution and enhancement of networks over generations.

Fitness calculation and selection: All NNs in a class are evaluated based on a fitness function in NEAT, a neuro-evolutionary method. This probabilistic evaluation determines which NNs will succeed in future generations. Each genome's fitness value does this. A selection operator, such as tournament selection, then considers fitness (Fig. 3c).

Crossover: Figure 3 shows the crossover operation, also called recombination, which combines the genetic information (e.g., NN parameters) of two selected individuals Fig. 3d.