IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

Volume 10, Issue 5, May 2021

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.122

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|

|| Volume 10, Issue 5, May 2021 ||

DOI:10.15662/IJAREEIE.2021.1005036

Data Validation and Quality Assurance in IOT: Issues, Challenges and Needs

Dr. Archana Verma

Assistant Professor, Computer Science & Engineering, Bipin Tripathi Kumaon Institute of Technology, Dwarahat, Uttarakhand, India

ABSTRACT: Things Board provides the ability to assign custom validation to your entities and manage these validation. Those validation are stored in the database and may be used for data visualization and data processing.Validation are treated as key-value pairs. Flexibility and simplicity of the key-value format allow easy and seamless integration with almost any IoT device on the market. Key is always a string and is basically an attribute name, while the attribute value can be either string, boolean, double, integer or JSON. Emergence of IoT as one of the key data contributors in a big data application has presented new data quality challenges and has necessitated for an IoT inclusive data validation ecosystem. Standardized data quality approaches and frameworks are available for data obtained for a variety of sources like data warehouses, webblogs, social media, etc. in a big data application. Since IoT data differs significantly from other data, challenges in ensuring the quality of this data are also different and thus a specially designed IoT data testing layer paves its way in.

KEYWORDS: data validation, quality assurance, IoT, ecosystem, social media, testing, attribute, value

I. INTRODUCTION

Attribute names

As a platform user, you can define any attribute name. However, we recommend to use camelCase. This make it easy to write custom JS functions for data processing and visualization.[1]

Attribute types

There are three types of validation. Let's review them with examples:

Server-side validation

This type of attribute is supported by almost any platform entity: Device, Asset, Customer, Tenant, User, etc. Server-side validation are the ones that you may configure via Administration UI or REST API. The device firmware can't access the server-side attribute.

Let's assume you would like to build a building monitoring solution and review few examples:

1. The latitude, longitude and address are good examples of server-side attribute you may assign to assets that represent building or other real estate. You may use this validation on the Map Widget in your dashboard to visualize location of the buildings.[2]
2. The floorPlanImage may contain a URL to the image. You may use this attribute to visualize floor plan on the Image Map Widget.[3]
3. The maxTemperatureThreshold and temperatureAlarmEnabled may be used to configure and enable/disable alarms for a certain device or asset.

Administration UI

- Go to Devices. Click on the particular device row to open device details. Select "Validation" tab. Choose "Server validation" scope. Click "+" Icon.

- Input new attribute name. Select attribute value type and input attribute value.
- Sort using "Last update time" to quickly locate the newly created attribute.[4]

REST API

Use REST API documentation to get the value of the JWT token. You will use it to populate the 'X-Authorization' header and authenticate your REST API call request.

Shared validation

This type of validation is available only for Devices. It is similar to the Server-side validation but has one important difference. The device firmware/application may request the value of the shared attribute(s) or subscribe to the updates of the attribute(s). The devices which communicate over MQTT or other bi-directional communication protocols may subscribe to attribute updates and receive notifications in real-time. The devices which communicate over HTTP or other request-response communication protocols may periodically request the value of shared attribute.[5]

The most common use case of shared validation is to store device settings. Let's assume the same building monitoring solution and review few examples:

1. The targetFirmwareVersion attribute may be used to store the firmware version for particular Device.
2. The maxTemperature attribute may be used to automatically enable HVAC if it is too hot in the room.

The user may change the attribute via UI. The script or other server-side application may change the attribute value via REST API.[6]

REST API

Use REST API documentation to get the value of the JWT token. You will use it to populate the 'X-Authorization' header and authenticate your REST API call request.
As an alternative to curl, you may use Java or Python REST clients.[7]

API for device firmware or applications:

- request shared validation from the server: MQTT API, CoAP API, HTTP API, LwM2M API;
- subscribe to shared attribute updates from the server: MQTT API, CoAP API, HTTP API, LwM2M API;.[8]

Client-side validation

This type of validation is available only for Devices. It is used to report various semi-static data from Device (Client) to ThingsBoard (Server). It is similar to shared validation, but has one important difference. The device firmware/application may send the value of the validation from device to the platform.[9]

The most common use case of client validation is to report device state. Let's assume the same building monitoring solution and review few examples:

1. The currentFirmwareVersion attribute may be used to report the installed firmware/application version for the device to the platform.
2. The currentConfiguration attribute may be used to report current firmware/application configuration to the platform.
3. The currentState may be used to persist and restore current firmware/application state via network, if device does not have the persistent storage.

The user and server-side applications may browser the client-side validation via UI/REST API but they are not able to change them. Basically, the value of the client-side attribute is read-only for the UI/REST API.[10]

Fetch client-side validation via REST API

Use REST API documentation to get the value of the JWT token. You will use it to populate the ‘X-Authorization’ header and authenticate your REST API call request.

Validation persistence

ThingsBoard stores latest value of the attribute and last modification time in the SQL database. This enables use of entity filters in the dashboards. Changes to the validation initiated by the user are recorded in the audit logs.[11]

II. DISCUSSION

Data visualization

We assume you have already provisioned device validation. Now you may use them in your dashboards. We recommend dashboards overview to get started. Once you are familiar how to create dashboards and configure data sources, you may use digital and analog gauges to visualize temperature, speed, pressure or other numeric values. You may also use cards to visualize multiple validation using card or entities table.
You may also use input widgets to allow dashboard users to change the values of the validation on the dashboards.[12]

Rule engine

The Rule Engine is responsible for processing all sorts of incoming data and event. You may find most popular scenarios of using validation within rule engine below:
Generate alarms based on the logical expressions against attribute values
Use alarm rules to configure most common alarm conditions via UI or use filter nodes to configure more specific use cases via custom JS functions.[13]
Modify incoming client-side validation before they are stored in the database
Use message type switch rule node to filter messages that contain “Post validation” request. Then, use transformation rule nodes to modify a particular message.
React on the change of server-side attribute[14]
Use message type switch rule node to filter messages that contain “Validation Updated” notification. Then, use action or external to react on the incoming event.
Fetch attribute values to analyze incoming telemetry from device
Use enrichment rule nodes to enrich incoming telemetry message with validation of the device, related asset, customer or tenant. This is extremely powerful technique that allows to modify processing logic and parameters based on settings stored in the validation.[15]

Performance enhancement

You can achieve higher performance with Validation Cache enabled (see cache.validation.enabled property of the Configuration properties)

Having validation cache enabled ThingsBoard will load the specific attribute from the database only once, all subsequent requests to the attribute will be loaded from the faster cache connection.
If you are using Redis cache, make sure that you change maxmemory-policy to allkeys-random to prevent Redis from filling up all available memory[17]
Most current scientific and industrial efforts in IoT are geared towards building integrated platforms to finally realize its potential in commercial scale applications. The IoT and Big Data contemporary context brings a number of challenges, such as providing quality assurance (defined by availability and veracity) for sensor data. Traditional signal processing approaches are no longer sufficient, requiring combined approaches in both architectural and analytical layers. This paper proposes a discussion on the adequate foundations of a new general approach aimed at increasing robustness and antifragility of IoT-based smart applications. In addition, it shows results of preliminary experiments with real data in the context of precision irrigation using multivariate methods to

identify relevant situations, such as sensor failures and the mismatch of contextual sensor information due to different spatial granularities capture. Our results provide initial indications of the adequacy of the proposed framework.[18]

Emergence of IoT as one of the key data contributors in a big data application has presented new data quality challenges and has necessitated for an IoT inclusive data validation ecosystem. Standardized data quality approaches and frameworks are available for data obtained for a variety of sources like data warehouses, webblogs, social media, etc. in a big data application. Since IoT data differs significantly from other data, challenges in ensuring the quality of this data are also different and thus a specially designed IoT data testing layer paves its way in. In this paper, we present a detailed review of existing data quality assurance practices used in big data applications. We highlight the requirement for IoT data quality assurance in the existing framework and propose an additional data testing layer for IoT. The data quality aspects and possible implementation models for quality assurance contained in the proposed layer can be used to construct a concrete set of guidelines for IoT data quality assurance.[19]

IV. RESULTS

The Internet of Things has been touted as the latest potential trend, but the fact is that the IoT is already here and is impacting businesses and consumers everywhere. Any object that is Internet-enabled and has machine-to-machine communication capabilities is part of the IoT—including smartphones, cars, and fridges. In addition to the typical Internet-related security concerns, applications are being made specifically for these devices, all of which bring about additional security concerns that quality assurance teams need to consider.[20]

Lack of Standardization
Since the IoT is still relatively new, organizations have been slow to set standards. However, as more consumers look to become connected, businesses must ensure that their IoT products and services are secure. According to a recent Kelley Blue Book survey, not only are consumers unaware of and unconcerned about the security threat to connected cars, but a majority believe that manufacturers need to offer software to protect their vehicles. Unfortunately, the report also showed that companies are failing to implement common security measures into their IoT devices and services, making them viable targets for hackers. QA teams must ensure that the software lives up to industry expectations and that ample protections are in place to prevent unique IoT vulnerabilities.[12]

Data Access
Information is king for any business, and some data is critical for survival. With devices and sensors communicating across the board, more data than ever is being generated, and QA management must ensure that this information is protected. Many IoT devices are missing data encryption capabilities and have lax password requirements. This means that practically anyone can hack into your hardware and see your information. If the attacker has direct access to the device, they may not even have to break in to steal your data. These gaping holes in security should concern every QA team and must be accounted for when creating and testing for application functionality.[14]

Continuous Testing and Review The IoT is not going anywhere. In fact, it's only predicted to become a stronger force in the near future. For this reason, it will be critical for QA teams to use agile testing methodologies to help constantly reinforce app security. Information Age contributor Chloe Green noted that you'll need to prioritize code review and repeat analysis in order to reduce overall risk. It will be critical to review updates and new deployments to ensure they are thoroughly tested instead of allowing these patches to fly under the radar.

"Implementing a software quality assurance benchmark on the software that interacts with IoT devices will become a standard operating practice," Green wrote. "Thanks to the complexity of IoT, if the software and its patches aren't continuously monitored and the code evaluated, this almost certainly guarantees failure."[16]

QA teams have a lot of challenges ahead of them with the rise of the IoT. By understanding what security risks the trend poses, QA professionals will be able to implement better safeguards across projects and better protect consumers from cyber attacks.

V. CONCLUSIONS

Internet of Things (IoT) has been generating a lot of interest in digitalisation. Particularly for industrial applications, the Industrial Internet of Things (IIoT) has seen adoption in an increasing speed. Vast amount and variety of sensors have been and are being deployed to generate insights and realise automated control to improve process and product/service quality, to improve efficiency and eventually to improve profitability, sustainability and customer satisfaction. To achieve the desired outcomes, the quality, i.e. accuracy and reliability of sensing data in the IoT or IIoT is crucial.

At National Metrology Centre (NMC), we have developed a data-driven approach for sensing data quality assurance in sensor networks for IoT and IIoT, namely Self-Diagnosis and Self-Healing (SDSH). Self-Diagnosis refers to autonomous and in-line monitoring and diagnostic of sensor health using a metrological ruler, i.e. the measurement uncertainty of the sensors. Self-Healing refers to the subsequent auto-compensation of error readings identified based on metrological principles.[18]

SDSH minimises lab-based calibration, which is laborious, interruptive to operations, costly and not feasible for IoT/IIoT applications due to the amount of sensors involved. It also enhances sustainability of IoT/IIoT as it minimises the resources needed for the maintenance of the sensing data quality and ensuring long-term reliability of the sensing data.

Here we would like to highlight three applications/projects which NMC is working on.

Application 1: Intelligent Buildings

Green buildings require smart design and intelligence in building condition and control for better sustainability. The intelligence comes from the many sensors installed in the building such as for indoor air quality (IAQ), temperature & humidity, energy meters, and so on. For the building controls to be effective, reliable and minimise energy consumption, the sensing data must be reliable and accurate to the acceptable level.

To help buildings achieve the sustainability goal, NMC's team is deploying IAQ sensor networks with SDSH function to drive building's fresh air ventilation control to achieve effective demand-controlled ventilation. Through continuous monitoring and correcting the sensing data by SDSH automatically, energy efficiency in fresh air ventilation is achieved without sacrificing indoor air quality and the long-term energy saving performance is sustained as a result of minimised sensing error.[20]

Green buildings require smart design and intelligence in building condition and control for better sustainability. The intelligence comes from the many sensors installed in the building such as for indoor air quality (IAQ), temperature & humidity, energy meters, and so on. For the building controls to be effective, reliable and minimise energy consumption, the sensing data must be reliable and accurate to the acceptable level.

To help buildings achieve the sustainability goal, NMC's team is deploying IAQ sensor networks with SDSH function to drive building's fresh air ventilation control to achieve effective demand-controlled ventilation. Through continuous monitoring and correcting the sensing data by SDSH automatically, energy efficiency in fresh air ventilation is achieved without sacrificing indoor air quality and the long-term energy saving performance is sustained as a result of minimised sensing error.[19]

Application 2: Smart Factories

Industry 4.0 opens up opportunities to manufacturing companies to implement digitalisation, maximise automation, improve productivity and quality, reduce human error, increase competitiveness and eventually save cost and enable long-term growth. One of the core technologies in Industry 4.0 is IIoT and the sensors in IIoT. To achieve the full potentials of IIoT, its sensing data accuracy and long-term reliability must be ensured with minimal resources for sustainability.

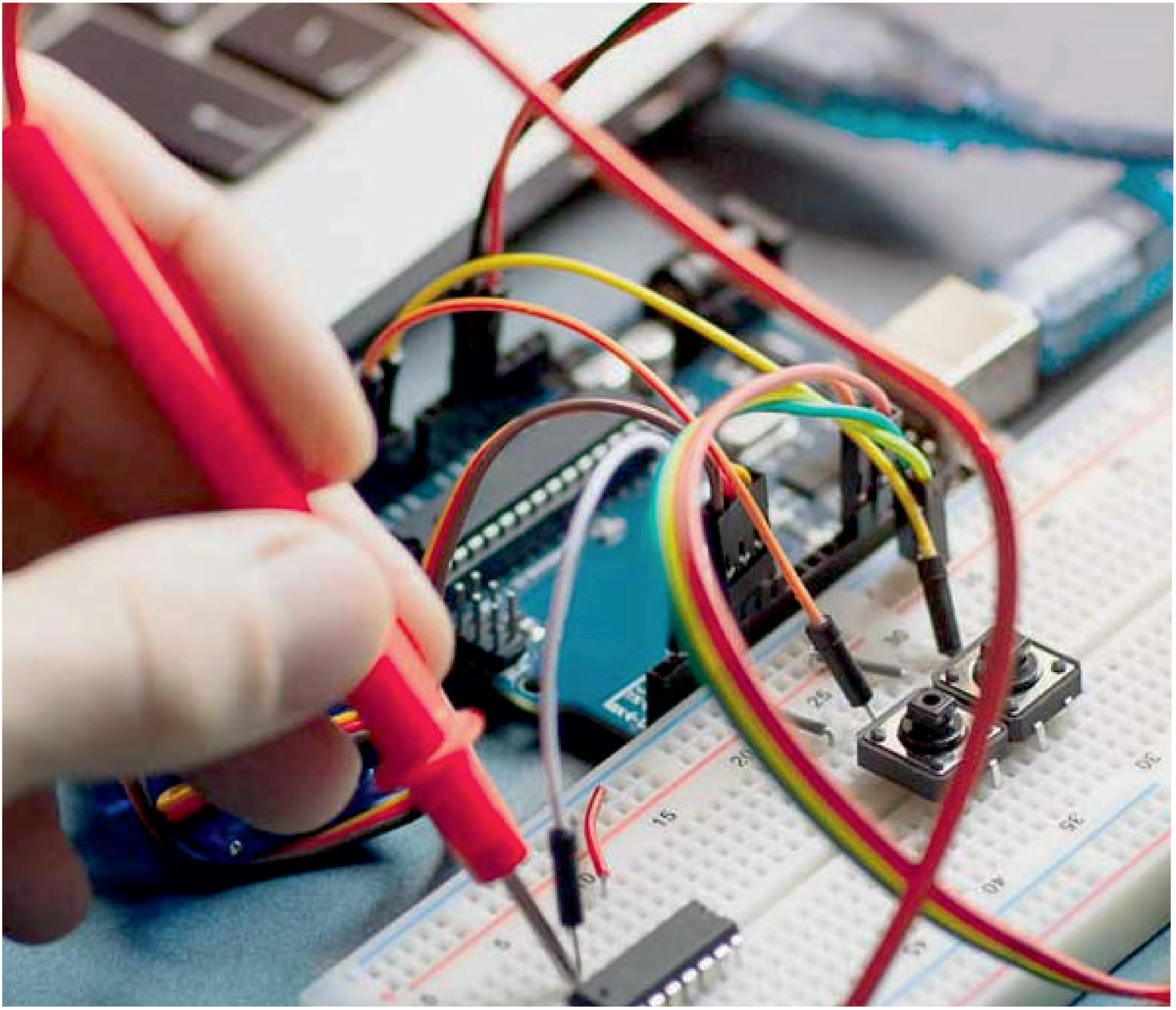
Temperature and vibration sensors are among the most commonly used for product quality monitoring/prediction, machine condition monitoring/predictive maintenance and more. NMC's team is further developing SDSH for temperature and vibration sensors in a manufacturing shopfloor environment. Machine learning method is being applied to learn sensor behaviour and to separate it from unwanted disturbances and noises. The target is to provide reliable sensing data with minimal disturbance to manufacturing process and minimal resources including manpower, time and service cost.[17]

Application 3: Structural Health Monitoring

Offshore structures exposed in extreme ocean waves and winds from time to time are subject to structural ageing and degradation. An efficient way for ensuring the offshore structural integrity and minimising the risk of incident is utilising the structural health monitoring systems, in which a large amount of strain, vibration and acoustic sensors are deployed. Since these sensors operate in harsh offshore and marine environments, a critical issue is the data accuracy and fidelity of the sensor measurement in continuous operation over long term. As these sensors are generally embedded in the offshore structure, the traditional approach by shutting down the offshore structure operation or dismantling the structure for calibration of the sensors isn't favourable. The team at NMC are developing data-driven technologies and solutions to assure the accuracy and fidelity of the sensor measurement data in-situ, without unnecessary interruption to the operation of the offshore structure. The offshore structural operational process is typically non-stationary, random and time varying, which makes the diagnosis of sensor health and identification of the sensor fault from structural heath degradation more challenging. Hence state-of-the-art data analysis algorithms for data correlation and signatures extraction corresponding to the sensor health status is being developed. With training data traceable to physical measurement standards, senor health diagnosis is being achieved by machine learning and artificial neural network. For possible deterioration of the sensor health leading to sensor drift, self-healing, i.e. autonomous compensation of the drift will be implemented based on metrological principle.[20]

REFERENCES

1. ISO 9000:2005, Clause 3.2.11
2. ^ Smith, Larry (2001). "Shift-Left Testing".
3. ^ "Quality Assurance vs Quality Control – Learning Resources – ASQ".
4. ^ "ASQ – Practical Quality Assurance for Embedded Software".
5. ^ "Define, Measure, Analyze, Improve, Control (DMAIC Approach) – ASQ".
6. ^ The Marketing Accountability Standards Board (MASB) endorses this definition as part of its ongoing Common Language in Marketing Project.
7. ^ "Quality Assurance vs Quality Control: Definitions & Differences | ASQ". asq.org. Retrieved 2020-11-21.
8. ^ Stebbing, L. (1993). Quality Assurance: The Route to Efficiency and Competitiveness (3rd ed.). Prentice Hall. p. 300. ISBN 978-0-13-334559-9.
9. ^ Prause, Christian; Bibus, Markus; Dietrich, Carsten; Jobi, Wolfgang (2016). "Software Product Assurance at the German Space Agency". Journal of Software: Evolution and Process. 28 (9): 744–761. doi:10.1002/smr.1779. S2CID 13230066.
10. ^ Garvin, D.A. (15 October 1984). "What Does "Product Quality" Really Mean?". MIT Sloan Management Review. Massachusetts Institute of Technology. Retrieved 29 November 2017.
11. ^ ASQ – History of Quality. Retrieved 17 November 2014
12. ^ Brooks, F.W. (1925). "William de Wrotham and the Office of Keeper of the King's Ports and Galleys". The English Historical Review. 40 (160): 570–579. doi:10.1093/ehr/XL.CLX.570.
13. ^ "Samuel Pepys and the Navy". Royal Museums Greenwich. 2015-08-17. Archived from the original on 2017-11-29. Retrieved 29 November 2017.
14. ^ Papp, J. (2014). Quality Management in the Imaging Sciences. Elsevier Health Sciences. p. 372. ISBN 978-0-323-26199-9.
15. ^ Wood, J.C.; Wood, M.C., eds. (2003). Henry Ford: Critical Evaluations in Business and Management. Vol. 1. Taylor and Francis. p. 384. ISBN 978-0-415-24825-9.
16. Methodology for data validation 1.0
17. ^ Data Validation, Data Integrity, Designing Distributed Applications with Visual Studio .NET
18. ^ Frequently Asked Questions about the new ISBN standard Archived 2007-06-10 at the Wayback Machine ISO.
19. ^ Chapter10. Data Validation
20. ^ More Efficient Data Validation with Spotless



International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

9940 572 462 ✆ 6381 907 438 ✉ ijareeie@gmail.com

www.ijareeie.com

Scan to save the contact details