Proceedings of the Pakistan Academy of Sciences: A Physical and Computational Sciences 59(4): 13-23 (2022)

ISSN (Print): 2518-4245; ISSN (Online): 2518-4253
https://doi.org/10.53560/PPASA(59-4)784

Pakistan Academy of Sciences

Research Article

Design and Implementation of Low-Cost Data Acquisition System for Small and Medium Enterprises (SMEs) of Pakistan

Muhammad Imran Majid*, Ejaz Malik, Tahniyat Aslam, Osama Mahfooz, and Fatima Maqbool

Electrical Engineering Department, Institute of Business Management (IoBM), Karachi, Pakistan

Abstract: This paper presents the development of low-cost and robust industrial IoT based data acquisition system primarily focused on domestic manufacturing industries striving to achieve goals and benefits of "Industrial 4.0". This proposes aims to promote DAQ System integration in traditional manufacturing process of the small and mid-sized industries of Pakistan with limited capacity of investment. Proposed method comprises of Arduino and it's IoT features for Data Collection, along with a self-developed PC based Centralized Software for Collection of Data, Graphical User Display and Storing collected Data in Local SQL Database. PC based Software replaces requirement of multiple software in case of traditional low-cost DAQ systems, like OPC Software for collecting data from industrial hardware, Java or PHP based any GUI and SQL Data storage. The analysis of work is done with the help of the Message Queue Telemetry Transport (MQTT) protocol. This project will be in further stages evaluated to add features of Supervisory Control, along with Data Acquisition hardware with minimum increase in cost and further upgrading PC Software to add more features of Industry 4.0, as compared to costly commercial solutions available in the market. A machine learning algorithm, k-nearest neighbors algorithm has been used to classify sensitive and non-sensitive data for improvising cloud security. K-Nearest Neighbors is also called KNN algorithm which is supervised machine learning classifier.

Keywords: Industry 4.0, Arduino, IoT, UDP/TCP, PC Software, Database, MQTT

1. INTRODUCTION

A recent World Economic Forum status report detailing 21 tipping points that will change the the world uptill 2025 which will leave a deepend impact on the way we perceive technology and society. The concept referred to as Industry 4.0 has revolutionized Manufacturing Process with the help of Information and Communication Technologies (ICT) manufacturing Operations systems – forming Cyber Physical Systems (CPS). Due to this industrial revolution, the market is transformed into considering the full value chain and changed the way an organization operates [1]. Many countries have contributed in creating domestic programs to improve the development and acceptance of Industry 4.0 technologies. In deutechland, the concept of "High-Tech Strategy 2020", in US "Advanced Manufacturing Partnership", whereas in Sinoland the "Made in China 2025" alognwith the french "La Nouvelle France Industrielle" has been gaining ground [2].

To communicate between machines and devices requires customizable configurations products, especially for small quantity numbers [3]. Moreover, Cyber phsycial systems are used to process data and information, to enable entitties to have a quicker response to enable resource utilization[4]. Hence, autonomous systems and smart grids using smart grids and power savings helps improve efficiency [5]. There has been a lot of emphasis on practical implications and utilization of technology related to Industry 4.0 [6] [7] [8] [9]. Lutfi et al. [10] have

Received: July 2022; Accepted: December 2022
*Corresponding author: Muhammad Imran Majid <imran.majid@iobm.edu.pk>

implemented Accounting Information System to facilitate SMEs. A computerised accounting system (AIS) enables us to achieve accurate data collection, handles large-scale transactions, and creates useful reporting for evaluation. Given the lack of AIS implementation in SMEs, despite its significant benefits, the purpose of this study is to investigate the antecedents of AIS implementation and their implications for long-term business performance in Jordanian SMEs.

As seen in Table 1, there are a number of emerging technologies that are linked to the futuristic fourth industrial revolution [5].

Our research area mainly focuses on Data Acquisition systems, which is an integral part of Supervisory Control and Data Acquisition Systems (SCADA), one of the unique and competetive technologies of the industry 4.0 . In this research paper, first known attempt of design and implementation of low-cost Data Acquisition system is made. Instead of using PLCs and Data loggers in conventional SCADA systems. Figure 1 is showing the system design. Overall system consists of two main parts, i.e., Arduino (Ethernet) based Nodes, installed on Ethernet Shield W5100, Self-designed IO Shield and C# based software for acquiring Data, storing Data in Local SQL Database and Displaying Data in Real Time using UDP Protocol, for minimal cost and adequate accuracy and communication speed.

2. METHODOLOGY

The multi channel data acquisition system is revealed in terms of hardware and software description. Work flow demonstration of system design is clearly shown in Figure 1.

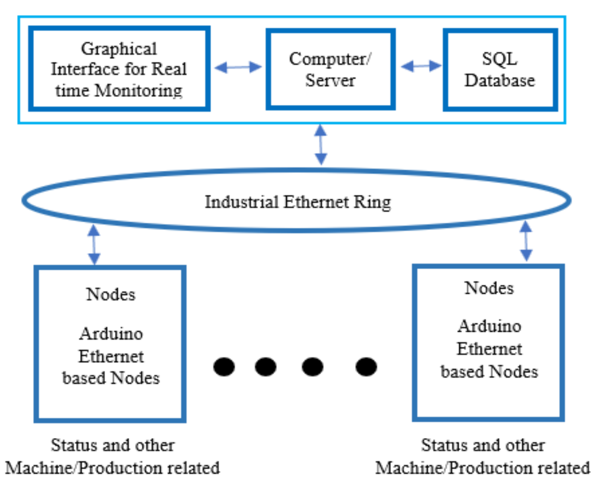

Fig. 1. Block level representation of System Design

Machines to collect Machine's status and other relevant data, their communication to Data Acquisition Server using User Datagram Protocol (UDP) and Data Acquisition Application Software for Data Collection, Displaying Real time Machine Status and other Relevant Data as well as Storing Data in SQL Server as per requirement of User. There are two parts of the system hardware and software. For the software part algorithm is presented. For PCB hardware, pictorial view of front and side is provided. Installed view is also given in the Figure 2, 3 and 4 respectively.

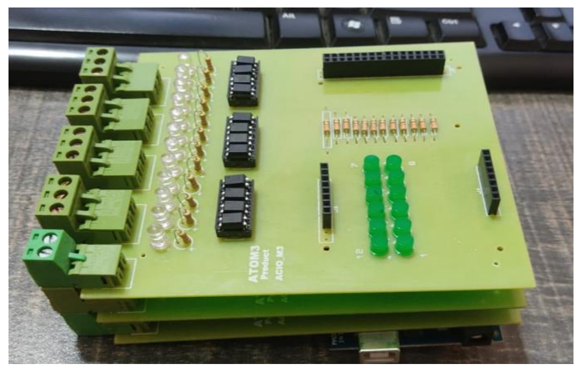

Fig. 2. PCB Hardware design (Front View)

2.1 Hardware Design

Hardware of each Node comprises of Arduino Mega, with its Ethernet Shield W5100 along with Fabricated Digital IOs Shields. Digital IOs Shield is used to convert 24VDC signals from Industrial Sensors and Relays to 5VDC signal which is compatible to Arduino Mega Voltages. Each Digital IOs Shield has 12 channels of 24VDC Input (Sink), total of 3 IOs Shield (36 Channels) are utilized for testing. Ethernet Shield (W5100 Chipset) is used for communicating to Data Acquisition Software. Arduino Mega is selected due to highest number of available Digital IOs over other models of Arduino Family.

Figures 2, 3 and 4 show the PCB hardware design and installation. The following Flow Chart describes the overall working of Arduino Ethernet based Nodes.

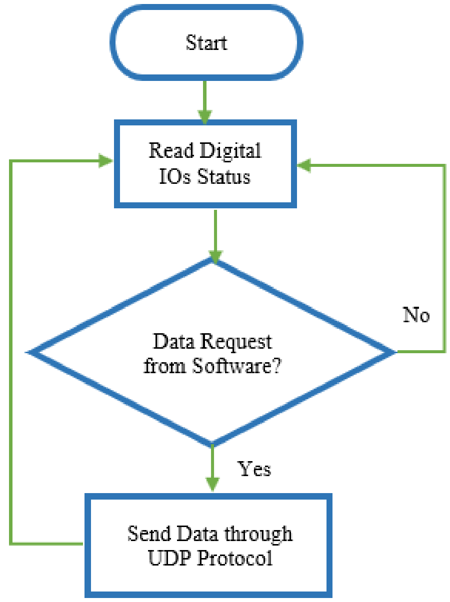

Table 1. Technologies of 4.0

Technologies	Definition
Computer-Aided Design and Manufacturing [CAD/CAM]	Development of projects and work plans for product and manufacturing based on computerized systems (Scheer, 1994).
Integrated engineering systems [ENG_SYS]	Integration of IT support systems for information exchange in product development and manufacturing (Kagermann et al., 2013; Bruun et al., 2015; Abramovici, 2007).
Digital automation with sensors [SENSORING]	Automation systems with embedded sensor technology for monitoring through data gathering (Saldivar et al., 2015).
Flexible manufacturing lines [FLEXIBLE]	Digital automation with sensor technology in manufacturing processes (e.g. radio frequency identification – RFID – in product components and raw material), to promote Reconfigurable Manufacturing Systems (RMS) and to enable the integration and rearrangement of the product with the industrial environment in a cost-efficient way (Brettel et al., 2014; Abele et al., 2007).
Manufacturing Execution Systems (MES) and Supervisory control and data acquisition (SCADA) [MES/SCADA]	Monitoring of shop floor with real time data collection using SCADA and remote control of production, transforming long-term scheduling in short term orders considering restrictions, with MES (Jeschke et al., 2017).
Simulations/analysis of virtual models [VIRTUAL]	Finite Elements, Computational Fluid Dynamics, etc. for engineering projects and commissioning model-based design of systems, where synthesized models simulates properties of the implemented model (Saldivar et al., 2015; Babiceanu and Seker, 2016).
Big data collection and analysis [BIG_DATA]	Correlation of great quantities of data for applications in predictive analytics, data mining, statistical analysis and others (Gilchrist, 2016).
Digital Product-Service Systems [DIGITAL_SERV]	Incorporation of digital services in products based on IoT platforms, embedded sensors, processors, and software enabling new capabilities (Porter and Heppelmann, 2014).
Additive manufacturing, fast prototyping or 3D impression [ADDITIVE]	Versatile manufacturing machines for flexible manufacturing systems (FMS), transforming digital 3D models into physical products (Weller et al., 2015; Garrett, 2014).
Cloud services for products [CLOUD]	Application of cloud computing in products, extending their capabilities and related services (Porter and Heppelmann, 2014).

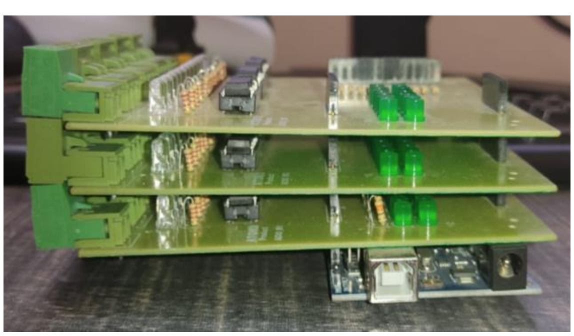

Fig. 3. PCB Hardware design (Side View)

Fig. 4. Hardware installed at Site

2.2 Software Design

In this section, software of the Data acquisition system is explained. Software is developed using C# language-based Windows Forms. C# is used for 1- Collecting Data from Arduino vis UDP Protocol, using Socket Programming as well as 2- Display collected data in real time on Windows Forms and 3- Store and retrieve that collected Data from SQL Database on timer based or on the request of the User/Operator. This self-developed software does not have any limitations of usage as like using any free OPC Software for data collection, also this software is a low-cost variant of any commercial run time Monitoring software or GUI.

2.3 Data Collection Layer

Collection of Data consist of Polling Mechanism. C# based software makes regular data requests to Controller (Arduino) to obtain field system data and store it in system database.

Arduino is equipped with Ethernet Interface via W5100 Ethernet Shield and uses Ethernet/IP Protocol for data communication. Communication Process is as follows: C# Software sends the data request to Arduino via Ethernet/IP Protocols. Ethernet/IP has UDP, TCP two communication protocols to choose from, because C# software (Server) is only responsible for sending "request packets", Arduino is only responsible for sending "reply packets", so the communication process does not need to use the connection-based TCP Protocol, but directly uses the UDP Protocol, which does not require high reliability and ensure the rate of data transmission.

2.4 Data Storage and Display

Collected data is analyzed using Main C# Software, according to user requirements. In our test case, Data must be stored if it's changed from its previous value. i.e. If "Status" of any Arduino Digital Inputs (36 Inputs, Channels in our Test) is changed from it's previous value, then this "new value" of that "Input" must be stored in the Database with current Timestamp, with its "Input Number". Values are being stored in Local SQL Database in test case, using Microsoft SQL Server. MS SQL Server is already being used in the Test company/Industry, makes it our first choice to work with. "Status" of Arduino Input Pins is also displayed on the GUI, which is the part of Main C# Software. "OFF" Pins are showed in Red, and "ON" Pins are displayed as Green.

When Status of "Input" e.g. INPUT0 is ON it means Machine's Line1 is ON and is in "Production". When Status of "Input" e.g. INPUT0 is OFF it means Machine's Line1 is OFF and is "Idle" i.e. not producing. Change in the "Status" of INPUT0 (ON to OFF) means that machine is stopped after completing a product and is now repeating it's cycle to produce new item.

So if we store only "Change of Input Status" in our database, we can count how many times Machine Line1 was "ON" in 24 Hours, thus calculating amount of Production made in that day.

3. DAQ AND COMMUNICATION PROTOCOLS

A data acquisition system (DAS or DAQ) is used for sampling signals that can evaluate physical conditions and convert measured values into digital which could be used by a computer. It converts analog waveforms into digital values for processing such as temperature, pressure, current, and other measurements into digital values for controllers or computers to process. Data Acquisition hardware communicates over several communication protocols. Some of them are discussed below [11].

3.1 Transmission Control Protocol/ Internet Protocol (TCP/IP)

TCP/IP protocol suite delivers the resources to the devices, to interact with one another via an Ethernet Local-Area Network (LAN), or by Wide-Area Network (WAN). TCP/IP allows the application layer messages to transfer between the devices. But it does not assure that the devices will interoperate with each other. Ethernet/IP-based data acquisition modules communicate via Industrial Ethernet protocol via TCP/IP protocol suites and standard IEEE 802.3. These are presented in both digital and analog I/O configurations and could be integrate with the standard cabling of Ethernet [12].

3.2 Modbus

It communicates by using a client-server (master-slave) method. It includes one device either a master or client that can initiate communications known as queries. Other devices such as either servers or slaves react by delivering the requested information to the master, or by initiating a request. A slave is a device such as a network drive, I/O transducer that can process data and transfer the information to the master by using Modbus [13].

3.3 Modbus TCP/IP

It is also known as the Modbus Remote Terminal Unit (RTU) protocol with a TCP interface which operates on Ethernet. The structure of Modbus messaging is the application protocol which specifies the instructions for managing and understanding the information independent of the transmission medium. TCP/IP indicates the transmission medium for Modbus TCP/IP messaging. It allows the transmission of binary data between the processors. It is a global standard that assists as the basis for the WWW. The main feature of TCP is to confirm that all data packets are transferred properly, whereas IP makes sure that messages are routed and addressed correctly. This protocol is used in the industrial field to understand the linkage among the industrial equipments. Initially, the communication protocol was intended to support interaction between Programmable Logic Controllers (PLCs). The protocol also contains built-in checksum protection that can ensure reliable data transmission in IoT networks [13].

3.4 MQTT Protocol

Message Queue Telemetry Transport (MQTT) protocol is depend on Transmission Control Protocol (TCP). It is a protoccol for IoT devices to interact with one another by using MQTT messages. The use of network resources can be decreased because the messages are lightweight in MQTT due to their optimized headers. The protocol comprises of MQTT brokers and clients. The MQTT broker can be placed in an IoT gateway, whereas the MQTT client could be an IoT actuators that can control data transmisssion between control commands and sensors to IoT sensors. This protocol can deliver reliable data transmission. Moreover, IoT actuators might need to establish a link with the IoT gateway during a period. By employing MQTT brokers, IoT actuators can create a outgoing TCP protocol to allow data communication [14].

3.5 IoT CAN BUS

This protocol is created on the CAN standard. It is a lightweight, centralized, flexible, and open system which can promptly integrate with controllers, sensors, and communication devices on the IoT-CAN bus board. This protocol is applied in many applications such as industrial manufacturing agriculture observing and environmental monitoring, etc. IoT-CAN does not require a master-slave protocol to sustain the system, therefore there is no longer need for registration mechanism of the master-slave protocol, employs static ID, distributes the CAN-ID interval depending on the module type and allow standard frames [14].

Figure 5, shows that the Sensors may be connected to an IoT gateway to form an edge that collects data from sensor nodes. The raw data may be fed into a simulator box for further processing and then transmitted to an on premises server or to a remote location using an IoT platform as shown in figure 5 below. The information may then be analyzed using AI or ML techniques for predicting machine health.

The figure 6 given shows a typical architecture of an MQTT based sensor module. It consists of power supply to power up the microcontroller and the required digital or analog sensors conntected to it. The amplifiers converts the intensity of current to to meet the reuired power for the sensors. An lcd would display the results received from the sensors after being processed by the microcontroller.

4. MQTT FUNDAMENTALS

Many IoT applications, such as smart cities, buildings, vehicles and industries, etc., are vulnerable to attacks. Therefore, it is necessary to improve the security such as access control, secure booting and device authentication [5]. Several IoT communications are rely on the hypertext transfer protocol (HTTP) protocol, but they require larger resource requirements and latency. On the other hand, the Message Queue Telemetry Transport

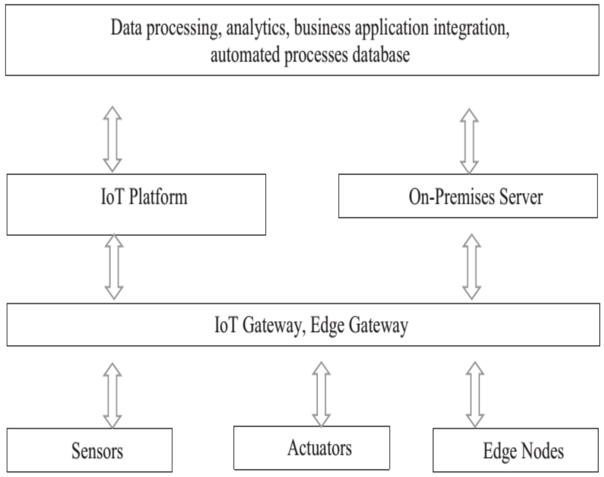

Fig. 5. IoT EDGE Architecture

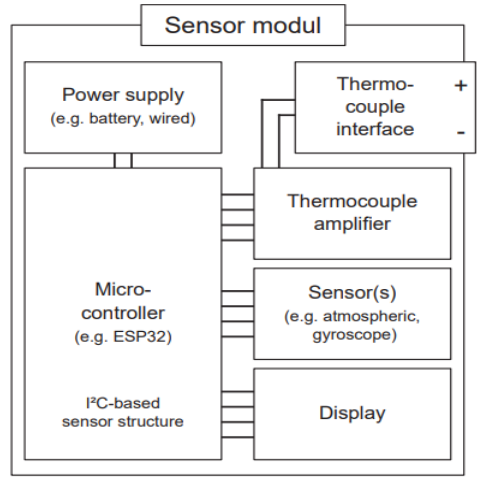

Fig. 6. MQTT Enabled Sensor Module [16]

(MQTT) protocol requires fewer resource requirements and employs less payload overhead packets with various quality of service (QoS), to overcome the problems of HTTP protocol. IBM released an open-source protocol called MQTT for publishing/subscribing messaging over client-server transport. The MQTT protocol are used in communication environments for the IoT and machine to machine (M2M) applications, where a low code footprint and small network bandwidth is required [6].

Figure 7, illustrates the basic setup of the MQTT communication protocol. A fundamental element is known as MQTT broker, that manages the transfer of information between the MQTT clients sending information (publisher) and the MQTT clients receiving the information (subscriber). Hence, no direct connections between the clients. But at a same time, a client can also be a subscriber and a publisher. The MQTT protocol can be deployed with the existing internal networks. Therefore, by employing proper sensor and server (broker) technology, MQTT allows a flexible and comprehensive process for monitoring. The information transmission can be done with three QoS levels by using the MQTT protocols [15].

- QoS 0 is used for only one delivery
- QoS 1 is used for at least one delivery
- QoS 2 is used for exactly-once delivery

5. TECHNOLOGICAL CONCEPT

A microcontroller (ESP32 or alternative ESP8266) used as the sensor module (Publisher). The publisher is integrated with the network and transfer the recorded data, collected from different sensors via MQTT protocol to a database. The sensor can be powered by a rechargeable battery (e.g. LiPo or Li-Ion) or operated by a USB connection through a power supply unit. In sensor module, the connection between an optional display and the transmission of the sensor data to the microcontroller can be done by using two different communication protocols: Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I2C). whereas, a Raspberry Pi (4 or Compute Module 4 with IO Board) equipped with the software can be serve as a MQTT broker. An existing corporate WiFi network is assumed. Figure 8, illustrate the proposed network of the project.

The proposed network lessens the burden of more processing in the cloud layer because we are getting reasonable amount of processing power at the IoT gateway which is being utilized to achieve more processing at the front of DAQ. Eventually this would affect the latency drastically and it would be reduced.

6. RESULTS & DISCUSSION

Data collected from DAQ System (as shown in Table 2) is used to calculate “Total Production” of a Machine in its 24 Hour of Operation. As previously manually count of “Total Production” was made by the Department In charge at the end of 24-Hour operation. Test results showed, that If Machine and Our system is working for 24 Hours, Data collected

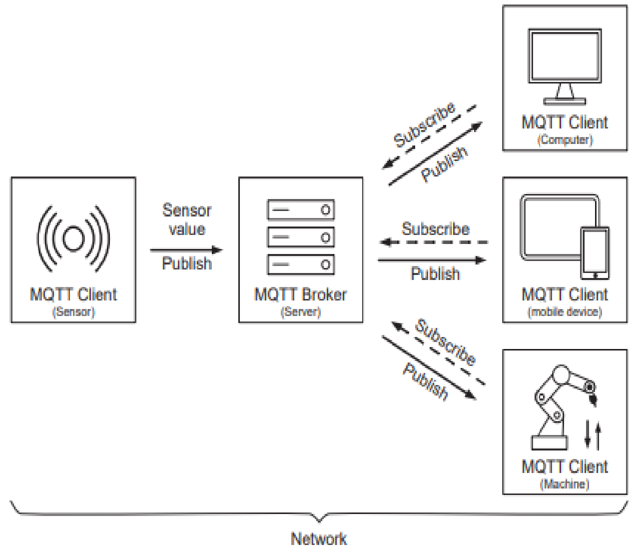

Fig. 7. MQTT communication protocol setup

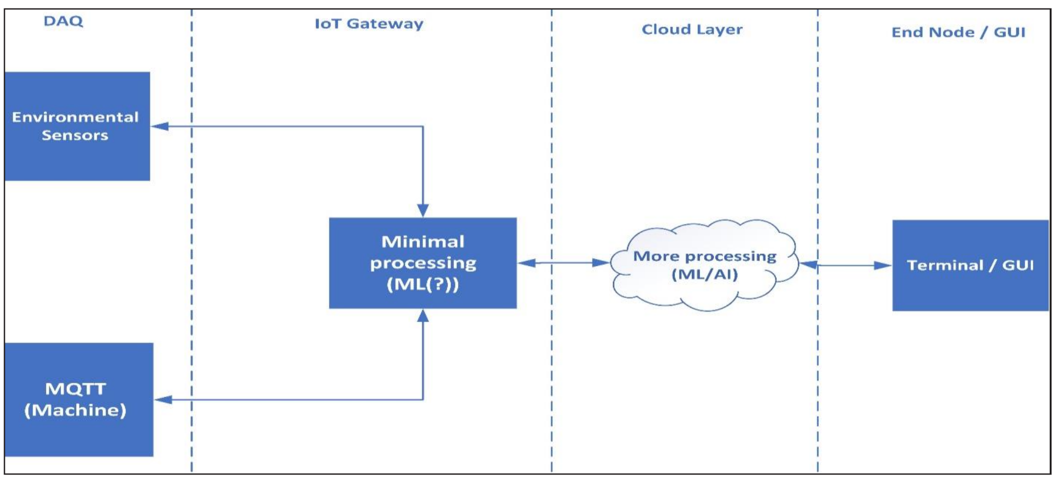

Fig. 8. Proposed Network of project

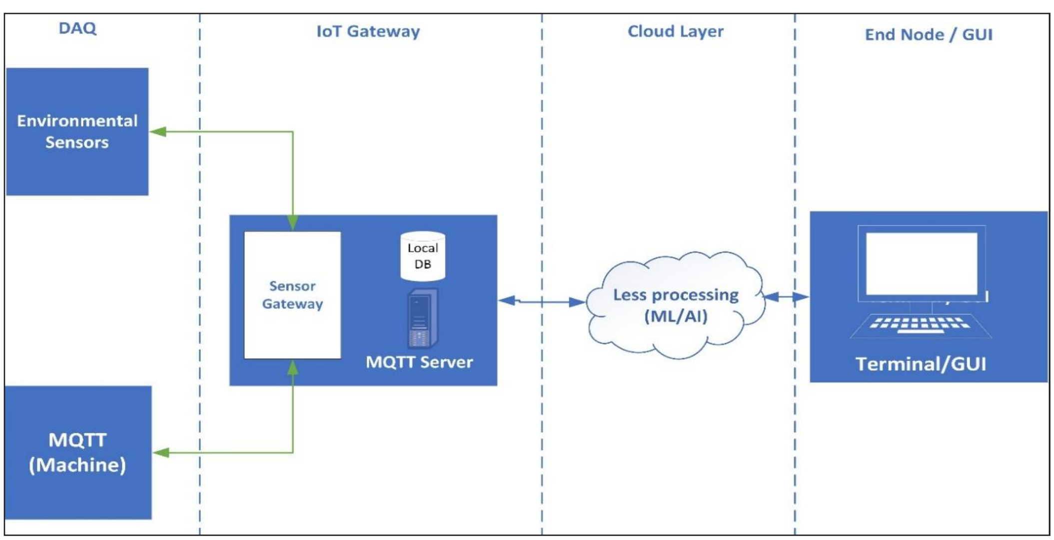

Fig. 9. Proposed Network

Table 2. Data collected from DAQ system

Day	Production count by DAQ System (Database)	Manual Production count from Production Department
Day 1	518	503
Day 2	541	575
Day 3	308	574
Day 4	404	537
Day 5	557	574
Day 6	415	539
Day 7	343	546
Day 8	432	501
Day 9	560	574
Day 10	14	180

is almost as same as Production Data collected from Production Department of our Test Site/Company.

Discussion related part includes description of Machine Learning at the edge, Edge AI, Artificial Intelligence at the cloud layer, Machine Learning at the cloud layer, K-nearest Neighbour KNN, supervised Machine Learning algorithm for cloud security and the algorithm related to the working of KNN.

As per Figure 9, the concept of DAQ based Machine Learning at the Edge and utilizing multiple NodeMCUs connected to IoT gateway can be incorporated. Here we would incorporate advanced features as in Machine Learning at the edge as well as AI/ML at the cloud layer.

6.1 Machine Learning at the Edge

Machine Learning at the edge (ML@Edge) concept refers to Machine Learning based models' capability to edge devices to collect data from the sources which are on the distance from cloud [17]. ML@Edge can be beneficial for security and safety purpose, predictive maintenance and detection of defects in production lines [18].

6.2 Edge AI

Edge AI is a buzz word in today's world. Edge computing is the combination of various techniques which are employed for data collection, data analysis and processing of data into the network [19]. Whereas, Artificial Intelligence refers to computing technique which made machine intelligent to take decision. Integration of Artificial Intelligence with Edge Computing is known as Edge AI. Edge AI can fasten the data processing, make network more secure and efficient [20]. Figure 10 is showing role of ML and AI at the Edge [21], [22].

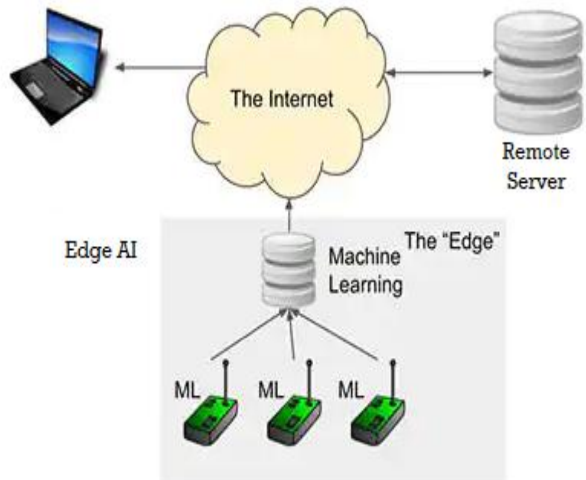

Fig. 10. AI and ML at the Edge [21], [22]

6.3 Artificial Intelligence at the Cloud Layer

The combination of Artificial Intelligence and Cloud Computing is emerging as business forward in the various ways beyond Information Technology (IT) [23]. AI based algorithms are applied in cloud computing to provide customer more effective services with the help experience. The two most common examples are Siri and Alexa [24].

6.4 Machine Learning at the Cloud Layer

Machine Learning has enhanced the capability of cloud computing with the help of various algorithms used for binary prediction, value prediction and category prediction. Machine Learning algorithms can help in predicting and forecasting business trends and help organizations in making decisions. Machine Learning as a Service (MLaaS) has become very powerful tool to train the data and make it cost effective [25]. Figure 11, is showing Machine Learning as a Service for cloud Computing [26].

6.5 Machine Learning Algorithm for Cloud Security

Various machine learning algorithms such as Artificial Neural Networks (ANN), K Nearest

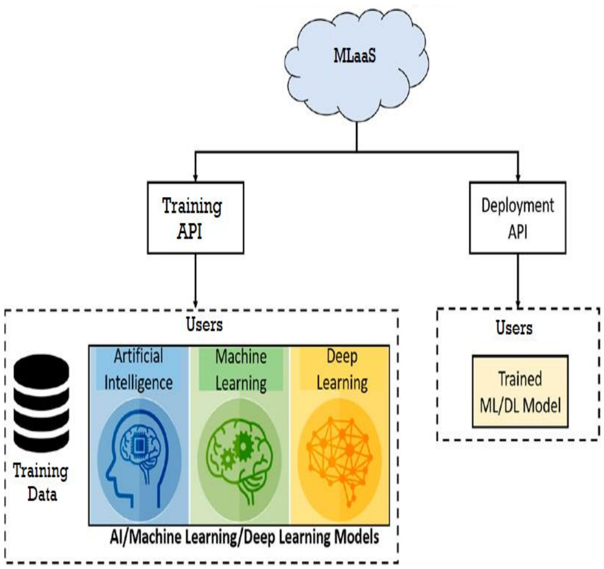

Fig. 11. Machine Learning as a Service (MLaaS) [26]

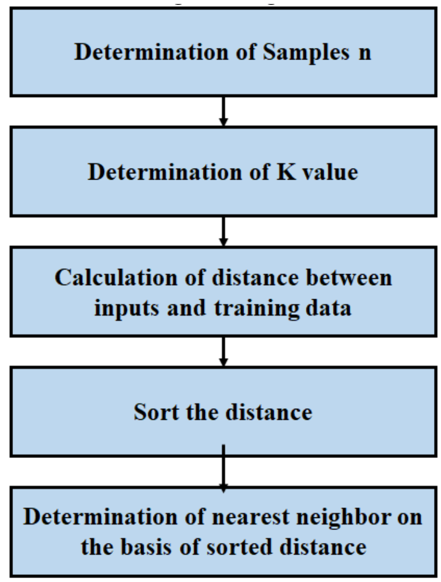

Fig. 12. KNN Classifier Working [27]

Table 3. Classification of Data using KNN [27]

Before Classification		After Classification				
Total Size of File (KB)	Total Record In File	Class-1		Class-2		Time Taken by KNN Classifier in (mS)
		Size	Record	Size	Record	
512	5094	352	3450	160	1644	1075

Neighbor (KNN), Naïve Bayes and Logistic regression can be used to improvise cloud security [27]. In this paper, we have reviewed use of KNN for cloud security management.

6.6 KNN for Improvising Cloud Security

The K Nearest Neighbor comes under supervised machine learning algorithm. This algorithm is used for pattern recognition, estimation, classification and prediction. In KNN, there is n number of labelled or defined samples [28].

$$D=\{d1,d2,d3\ldots\ldots dn\}$$

Where D is representing total number of samples and d1,d2…dn is representing different data or samples. Figure 12, is r showing working of KNN Classifier.

KNN algorithm was used to classify sensitive and non-sensitive data. KNN has divided data into two distinct classes Class-1 and Class-2. Class-1 is representing non sensitive data whereas class-2 is representing sensitive data. KNN Classifier has taken 1075ms to classify data, as shown in Table 3.

7. CONCLUSION

The paper is an attempt to propose a novel framework for data acquisition of environment sensors positioned at a machine stationed in industry based in either textile or automotive sectors. Proposed method comprises of Arduino and its IoT features for Data Collection, along with a self-developed PC based Centralized Software for Collection of Data, Graphical User Display and Storing collected Data in Local SQL Database. The production count spread over a 24-hour period is accounted. The experimental analysis of work is done by employing a lightweight MQTT protocol. The basic network is analyzed in terms of edge computing and advanced analytics including Machine learning and Artificial Intelligence. Proposed future extensions are proposed in these domains based on the ideas presented.

8. CONFLICT OF INTEREST

The authors declare no conflict of interest.

9. REFERENCES

1. G.L. Tortorella, and D. Fettermann. Implementation of industry 4.0 and lean production in brazilian manufacturing companies. International Journal of Production Research 56: 2975-2987 (2018).
2. L.S. Dalenogare, G.B. Benitez, N.F. Ayala, and A. G. Frank. The expected contribution of Industry 4.0 technologies for industrial performance. International Journal of Production Economics 204: 383–394 (2018).
3. S. Wang, J. Wan, D. Li, and C. Zhang. Implementing Smart Factory of Industrie 4.0: An Outlook. International Journal of Distributed Sensor Networks 2016: (2016).
4. G. Schuh, R. Anderl, J. Gausemeier, M.T. Hompel, and W. Wahlster. Industrie 4.0 Maturity Index Managing the Digital Transformation of Companies. acatech STUDY (2017).
5. S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert. Industrial Internet of Things and Cyber Manufacturing Systems. Industrial Internet of Things 3-19 (2017).
6. G.J. Cheng, L.T. Liu, X.J. Qiang, and Y. Liu. Industry 4.0 Development and Application of Intelligent Manufacturing. International Conference on Information System and Artificial Intelligence (ISAI) 407-410 (2016).
7. H. Kagermann. Change Through Digitization-Value Creation in the Age of Industry 4.0. Management of Permanent Change, H. and P. A. and R. R. Albach Horst and Meffert, Ed. Wiesbaden: Springer Fachmedien Wiesbaden (23–45) 2015.
8. M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M.J. Harnisch. Industry 4.0 The Future of Productivity and Growth in Manufacturing Industries. Economics, History (2016).
9. T. Stock, and G. Seliger. Opportunities of Sustainable Manufacturing in Industry 4.0. Procedia CIRP 40: 536-541 (2016).
10. A. Lutfi, A.L. Al-Khasawneh, M.A. Almaiah, A. Alsyouf, and M. Alrawad. Business Sustainability of Small and Medium Enterprises during the COVID-19 Pandemic: The Role of AIS Implementation. Sustainability MDPI 14: 5362 (2022).
11. R.A. Atmoko, R. Riantini, and M.K. Hasin. IoT real time data acquisition using MQTT protocol. Journal of Physics: Conference Series 853: 012003 (2017).
12. A. Sharma, S. Airan, and D. Shah. Designing C Library for MODBUS-RTU to CANBUS and MODBUS-TCP IOT Converters. Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India (2021).
13. K. Yang. B. Zhang, J. Zhang, and J. Zhu. Design of Remote-Control Inverter Based on MQTT Communication Protocol. Proceedings of the 2021 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan (1374-1378) 2021.
14. H. Gong, J. Li, R. Ni, P. Xiao, and H. Ouyang. The data acquisition and control system based on IoT-can bus. Intelligent Automation & Soft Computing 30: 1049–1062 (2021).
15. T. Yokotani, and Y. Sasaki. Comparison with HTTP and MQTT on required network resources for IoT. Proceedings of the International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC) (1–6) 2016.
16. E. Riedel. MQTT protocol for SME foundries: potential as an entry point into industry 4.0, process transparency and sustainability. Procedia CIRP 105: 601-606 (2022).
17. M. Verhelst, and B. Murmann. Machine Learning at the Edge. NANO-CHIPS 2030, The Frontiers Collection book series (FRONTCOLL) 293-322 (2020).
18. M. Polese, R. Jana, V. Kounev, K. Zhang and S. Deb. Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks. IEEE Transactions on Mobile Computing 20: 3367-3382 (2021).
19. M. Merenda, C. Porcaro, and D. Iero. Edge Machine Learning for AI-Enabled IoT Devices: A Review. sensors MDPI 20: 2533 (2020).
20. E. Bertino, and S. Banerjee. Artificial Intelligence at the Edge. A Computing Community Consortium (CCC) (2020).
21. S. Hymel. What is Edge AI? Machine Learning + IoT. https://www.digikey.com/en/maker/projects/what-is-edge-ai-machine-learning-iot/4f655838138941138aaad62c170827af. [Accessed 26 07 2022].
22. G. Anadiotis. Machine learning at the edge: TinyML is getting big (2021). https://www.zdnet.com/article/machine-learning-at-the-edge-tinyml-is-getting-big/. [Accessed 26 07 2022].
23. Z.A. Mohammad, R. Belgaum, S. Musa, M.M. Alam,

and M.S. Mazliham. Role of artificial intelligence in cloud computing, IoT and SDN: Reliability and scalability issues. International Journal of Electrical and Computer Engineering (IJECE)11: (2021).
24. S. Mohammed, W.C. Fang, and C. Ramos. Special issue on "artificial intelligence in cloud computing". Computing 105: 507-511 (2023).
25. A. Qayyum, A. Ijaz, M. Usama, W. Iqbal, J. Qadir, Y. Elkhatib, and A. Al-Fuqaha. Securing Machine Learning in the Cloud: A Systematic Review of Cloud Machine Learning Security. Big Data 3: (2020).
26. G.B. Dasgupta. AI and its Applications in the Cloud strategy. ISEC 2021: 14th Innovations in Software Engineering Conference (formerly known as India Software Engineering Conference) (2021).
27. U.A. Butt, M. Mehmood, S.B.H. Shah, R. Amin, M.W. Shaukat, D.Y. Suh, and M.J. Piran. A Review of Machine Learning Algorithms for Cloud Computing Security. Electronics, MDPI 9: 1379 (2020).
28. M.A. Zardari, L.T. Jung, and N. Zakaria. K-NN Classifier for Data Confidentiality in Cloud Computing. 2014 International Conference on Computer and Information Sciences (ICCOINS) (2014).