

# IoT-Enabled Machine Learning Framework for Real-Time Sensor Data Classification Using MATLAB

Hemant Pandharinath Chavan<sup>1</sup>, Lokesh Purushottam Nikhade<sup>2</sup>, Sharana Tarannum Mohd. Zaheer<sup>3</sup>, Kalpana Kisan More<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of Electronics & Telecommunication Engineering, Government College of Engineering, Karad, Maharashtra, India.

<sup>2</sup>Assistant Professor, Electronics & Telecommunication Engineering, Government College of Engineering, Chandrapur, Maharashtra, India.

<sup>3</sup>Assistant Professor, Department of Electrical Engineering, Government College of Engineering, Chandrapur, Maharashtra, India.

<sup>4</sup>Assistant Professor, Department of Electrical Engineering, Government College of Engineering, Karad, Maharashtra, India.

---

## ARTICLE INFO

## ABSTRACT

Received: 10 Nov 2023

Revised: 18 Dec 2023

Accepted: 24 Dec 2023

The Internet of Things (IoT) technology has improved so rapidly that sensor networks are widely used in different areas that produce a large flow of real-time data. Real-time decision-making based on this sensor data is an important task that can be effectively handled by processing and labeling sensor data and is vital in a variety of applications to healthcare, smart cities, and industrial automation. The study introduces a machine learning model along with the IoT-enabled machine learning block to classify sensor data in real-time utilizing MATLAB. The framework incorporates the data retrieved by IOT sensors, preprocessing, and training and deployment of machine learning models. Such machine learning algorithms as decision trees, support vectors machines, and neural networks are introduced to compare their abilities to classify real-time sensor data. The performance of the framework is measured by the accuracy of classification, processing rate, and real-time efficiency prove that MATLAB can be used to apply IoT machine learning solutions. It can be concluded that the suggested framework is efficient when dealing with massive sensor data and it provides effective conclusions on creating IoT-based apps where instant data categorization is needed.

**Keywords:** Internet of Things (IoT), real-time sensor data, machine learning, data classification, MATLAB, decision trees, support vector machines, neural networks, sensor networks, real-time processing.

---

## 1. INTRODUCTION

Internet of Things (IoT) refers to a fast-growing technology paradigm, which is used to network physical devices via the internet, allowing them to access, disseminate and exchange information and data without involving a human being. The IoT has numerous applications in diverse fields of systems in different industries like healthcare, smart cities, manufacturing, agriculture, and environmental monitoring[1]. The main advantage of the IoT is that it has an ability to develop intelligent systems which would add more value to the decision-making and efficiency due to the smooth flow of data among the connected objects. The use of IoT technology by industries allows optimization of the processes, enhancing customer experiences and making much more automation possible[2]. With the development of IoT networks, it is clear that the amount and nature of the data created by the connected devices are increasing, and the analysis and classification of data become vital.

IoT systems are based on real-time sensor data because sensors in their surrounding constantly gather information about the surrounding (temperature, humidity, movement, and light). This live data allows immediate reactions and practical information which are vital in such tasks as predictive maintenance, traffic, and remote patient care[3]. Nevertheless, it is very difficult to handle and process sensor data which are in large volumes in real time. IoT sensors produce large volumes of data, which demand effective storage, transmission, and processing systems so that the required results can be obtained promptly and correctly. Also, the fact that the information obtained by various sensors is heterogeneous may also make its fusion and processing difficult[4].

Machine learning is an important aspect in analysis and classification of real-time sensor data in IoT systems. Once historical data is provided to the IoT systems, they can make predictions or write classifications using machine learning algorithms without any explicit programming required (Winter, 2017). Unsupervised and supervised learning algorithms such as decision tree, support vector machine, and neural networks are very popular living up and classifying sensor data, thus allowing systems to change and self-improve with time[5]. Machine learning is also useful in deriving meaningful insights of sensitive and high-dimensional data; it is an indispensable device in real-time decision-making in IoT applications.

The advanced MATLAB platform that provides numerical computation and programming is also best adapted to the implementation of IoT solutions and machine learning structures. It has ready-to-use functions and algorithms in its large toolboxes, including the IoT Toolbox and the Machine Learning Toolbox, that acquire data, process signals, and train machine learning models. The friendly interface of MATLAB and vivid visualization features are also helpful to test, prototype, and deploy IoT applications (researchers and engineers, find it easy). The capability of analyzing large volumes of data and operating in real-time, the platform is a perfect solution when creating and implementing machine learning models to classify sensor data in an IoT setting.

The present study is devoted to the creation of IoT-based machine learning with real-time sensor data classification in MATLAB. The main goal is to make a powerful system that is able to process and classify real time information in IoT sensors with high productivity. The framework combines both data acquisition and preprocessing with classification algorithm so as to give correct and timely prediction. In this study, the investigator seeks to investigate the possibility of applying machine learning methods in MATLAB to analyze IoT data in real-time and the performance of the different machine learning models in classifying sensor data in real-time applications.

## 2. LITERATURE REVIEW

The research of IoT data management as a research topic has been in the limelight of the studies as a result of the difficulties, which dry up data with huge volumes of real time data generated by sensors on the IoT systems. The IoT networks rely on data collection, transmission and processing, which is essential to generate actionable insights[6]. A number of studies have been conducted on efficient data collection mechanisms that guarantee successful data capture off the variety of sensor types based on energy, bandwidth and computational resources constraint. The study of the data transmission underlines the necessity of scalable and low-latency communication schemes that would help to be able to exchange data between the devices and central systems smoothly[7]. Also, real-time processing of large amounts of data is also important. To process the IoT data, researchers have been coming up with a number of architectures, including edge computing and cloud computing[8]. In specific, edge computing minimizes latency since it processes information nearer to the source, which is important with time-sensitive applications. Nevertheless, data transmission optimization, data integrity, and privacy concerns in the IoT systems are still under discussion[9].

The concept of machine learning algorithms has found a lot of interest in sensor data classification in the IoT systems. The algorithms offer a viable source of the information of value out of the unprocessed data that the sensors recorded. Decision trees are popular because they are easy to understand and interpret which makes them applicable in classification problems where the features are discrete[10]. Another widely used one is support vector machines (SVMs) because they can also process high-dimensional information and can efficiently classify complex problems that cannot be clarified by non-linear models. Categorization of sensor data has also been done using neural networks, specifically the deep models of neural networks, with the advantage of (i) modeling intricate patterns and relationships in massive data volumes (ii) they have demonstrated their application in sensor data classification. New developments in the deep learning models, namely convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated good prospect in sensor data classification, particularly in time-series data classification[11]. These models can be trained on large volumes of data and their accuracy in classifying improves with time. Nonetheless, there are still difficulties to optimize these algorithms to real-time processing and to make sure that they should be scaled according to various IoT contexts[12].

In the case of real-time data classification, many IoT applications require fast decision-making, so the timely classification of data is vital. The current techniques of real-time sensor data classification are aimed at optimizing the latency and ensuring high accuracy. The studies have investigated lightweight machine learning models which

are able to offer more quick processing speeds and yet do not affect performance[13]. Model pruning techniques, quantization techniques and distributed learning techniques have been put forward to obtain improved efficiencies of real-time classification tasks. Also, there is consideration of hybrid methods which integrate both machine learning and signal processing methods to enhance the stability and precision of classification in the presence of different conditions in the environment. In spite of these developments, issues surrounding the ability to attain real-time processing particularly on low resource based IoT devices are still present[14]. Also, it has remained research to make sure that the models generalize to other types of sensors, as well as to different environments.

The popularity of MATLAB in the application of IoT has been attributed to the ability of the tool to analyze data, simulate and learn. The platform includes a complete collection of toolboxes, including the IoT Toolbox and Machine Learning Toolbox, which allow making the development and deployment of the IoT-based machine learning models easier[15]. Thanks to its large-scale data processing capabilities, as well as the real-time nature of its data processing system, MATLAB would be the best option to apply machine learning frameworks to the IoT environment[16]. Research has established MATLAB to be effective on real-time sensor data analysis specifically in areas that require use of MATLAB such as smart grid management, healthcare monitoring and industrial automation. The visualization features of the platform are also essential to provide the results of complex IoT data and a machine learning model in a convenient form. Nevertheless, MATLAB provides a well-established research and prototyping environment, yet problems of providing solutions to large-scale IoT applications are yet to be studied.

### 3. METHODOLOGY

The implantation of the IoT-based machine learning system aims at building an effective framework of sensor data classification in real-time, including a number of main elements that would facilitate the free flow of data between collection and processing. The first element is referred to as IoT sensors that are installed to track different environmental parameters, including, but not limited to, temperature, humidity, motion, and pressure. These sensors would constantly measure information and relay it to a central processing unit where further analysis would be done. The process of data acquisition is made easier by a middle layer which gathers raw data of the various sensors and sends it to a cloud or edge computing layer. This combination allows the framework to scale and handle large amounts of sensor data and store and do more complicated computations on the cloud where needed or use edge computing to keep latency down and has faster decision-making potential in real-time applications[17]. The general design guarantees an effortless combination of sensor networks, data storage and real-time processing with machine learning classification.

Preprocessing of the raw sensor data is an important process in enhancing correct and useful classification of the sensor data. Raw sensor data may include noise, inconsistency or redundant data thus data cleaning methods like missing value imputation and outlier detection are will be applied initially to ascertain the tidiness of the data. This is followed by normalization which is done to standardize the data so that all the features are relatively similar in their scale and that some variables do not dominate the classification procedure[18]. This pre-processing is an important stage, since any dirty or unstructured data can severely harm the quality of machine learning models. Also, feature extraction techniques can be used to convert the crude sensor data to more useful features that better depict the underlying trends in the data. The above steps grant a clean and normalized dataset, which is within reach of being classified using machine learning algorithms.

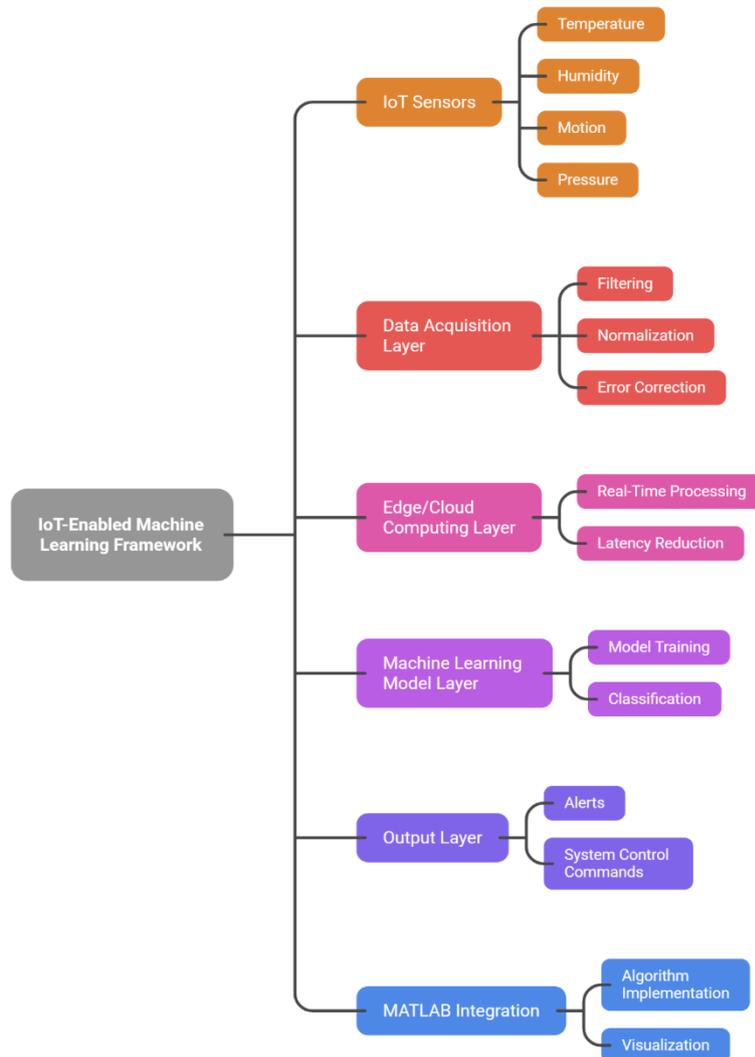


Figure 1: IoT-Enabled Machine Learning Framework for Real-Time Sensor Data Classification

The rise of IoT has enabled the creation of machine learning frameworks capable of real-time sensors data classification as shown by architecture in figure 1. The framework starts with IoT Sensors, who constantly gathers the information about the surrounding, e.g. temperature, moisture, or movement. This sensor data is now broadcasted to the Data Acquisition Layer where preprocessing actions such as filtering, normalization, and error correction is done so that the data can be clean and allowance to be made to the analysis. The information after processing is sent to the Edge/Cloud Computing Layer where real-time processing is done. It is the first layer that does the initial calculations then transfers the data to the machine learning models to be classified[19]. The concentration in the Machine Learning Model Layer involves many different machine learning algorithms like decision trees, support vectors machines or neural networks to identify the sensor data input as belonging to trained models. Final results of the classification are then provided to the Output Layer where the predictions or alerts are made and may cause a system adjustment, inform users, or a system control update. MATLAB is the key node in the whole process and it offers the features to implement machine learning model in the field, real time processing of data and visualization which results can be seen and the system can work in a stable and efficient manner[20].

In classifying real-time sensor information, machine learning models are chosen, which can work with high-dimensional and time-sensitive data. Significantly supervised learning algorithms, which are decision trees, support vector machines (SVM), and neural networks, are selected because of their use in task classification of sensor data

through some of the previous labeled data. The use of decision trees is preferred because of their simplicity and easy interpretation and the decision trees are very suitable in the case where interpretation of the process of decision-making is a significant factor. The SVMs are chosen due to their capability to deal with non-linear data and also offer powerful classification even in high-dimensional spaces. Neural networks and especially deep learning models have been added to capitalize on their ability to make complex correlations of large datasets, especially with time-series data. These models are trained with known samples of some of the types of sensors and therefore the framework is able to categorize the new unknown data with high accuracy. Processing in real-time is emphasized so that such models would be able to classify the data with delays of major significance.

#### 4. EXPERIMENTS AND DATA COLLECTION

The research experimental setup will entail the deployment of different kinds of IoT sensors so as to obtain real-time data on the environment. These sensors are chosen depending on how much they are relevant to the application and they should have the capacity to measure the relevant parameters accurately. Examples of common sensors are temperature sensors, humidity sensors and motion detectors. Temperature sensor devices including the DHT11 or the DS18B20 can give accurate temperature measurements which are vital in climate control and other industrial applications. Humidity sensors are devices that measure moisture content of the air like the DHT22 and are essential to environmental monitoring and to agricultural systems. Motion sensors include the PIR (Passive infrared sensor) which detects motion and can be commonly applied to security and Smart home uses. Such sensors generally have digital or analog outputs, which relay the information to the processing unit on a real-time basis. Each sensor has specifications like their accuracy, response time, and sampling rate which can be important in determining the credibility and promptness of the system when it is in a dynamic environment.

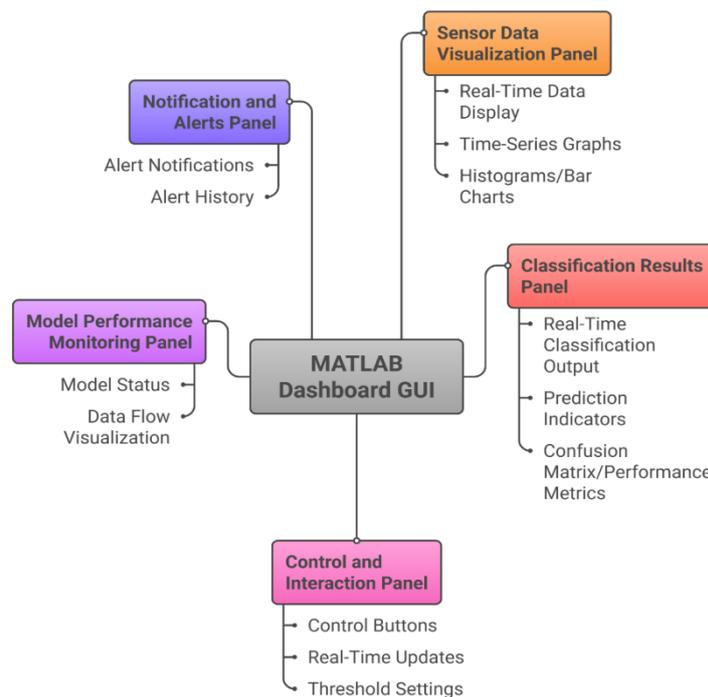


Figure 2: MATLAB Dashboard of GUI for Real-Time Sensor Data Monitoring and Classification

Figure.2 presents the MATLAB Dashboard GUI, which is used to create the real-time monitoring and the classification of sensor data in an IoT-powered machine learning setup. Some of the main panels shown in the dashboard include Sensor Data Visualization Panel, which presents a live view in real-time plot and histogram allowing the user to understand the values of such parameters as temperature, humidity, and motion. The Classification Results Panel displays the results of machine learning models that reveal whether the data is recognized as normal, alert, and anomaly as well as performance measurements such as accuracy and precision. The Control and Interaction Panel offers controls that allow the management of the data collection process i.e. starting, stopping, or

retraining the system, and the ability of the user to adjust threshold settings to generate real-time alerts. All the status of the machine learning model such as the processing time and the amount of resources it uses are monitored by the Model Performance Monitoring Panel. Finally, Notification and Alerts Panel is used to point out important problems by showing real-time alerts in case sensor data matches or surpasses set limits.

Data collection process is meant to capture sensor data in real-time and have them transmitted within their designated time so that the central processing unit can analyze them. The sensor has an interface with the processing unit (i.e. an Arduino or a Raspberry Pi) to which one sensor is connected. The sensors measured by the microcontroller are sampled at a set period, so there is an unceasing flow of real time data. The information is in turn sent through communication protocol such as Wi-Fi, Bluetooth or LoRa to the cloud or, edge computing layer. The selection of communication protocol will depend on the application need i.e. range, power consumption, and speed of data transmission. In such an arrangement, the data is relayed in packet form and the microcontroller also performs some simple pre-processing like verifying the data and checking its errors before transmission to the processing unit. This will be to make sure that the data obtained is clean and can be analysed further.

The sensor data gathered is passed into the environment in MATLAB to be operated on in real-time. Then this data is imported into MATLAB via the MATLAB IoT Toolbox that can be directly connected to interoperate with IoT devices, and it is also capable of providing a smooth flow of data streamed out of the sensors. Such toolbox enables linking a microcontroller with MATLAB so that real-time sensor readings can be fed into the MATLAB environment constantly to analyze them. The system includes an interface to the data streaming, sensor data are being processed in real time on arrival, which can be immediately classified and responded to. A Real-Time Data Acquisition of MATLAB is employed to operate the data entering the system, such that the system is capable of processing data as it arrives, in constant. Besides, the data is subjected to real time signal processing methods to clean and preprocess data before it is inputted into machine learning models. The above arrangement will be used in implementation of the machine learning models that are trained on the past data to facilitate the classification of the incoming information on the sensors in real time. Deployment of the trained models is executed by means of the MATLAB Machine Learning Toolbox and the results are instantly available to be analyzed and plotted. The experimental structure designed in MATLAB is such that the framework can stitch large amounts of sensor data with ease without compromising on real-time processing, which makes it practical in a large number of IoT applications.

## 5. RESULTS AND DISCUSSION

The performance of the machine learning models that were utilized in this framework was assessed using some major measures such as accuracy, precision, recall, and F1 score. Figure 3 shows the temperature data displayed in real-time captured by the sensors which is used as inputs in the classification process. The models reached an accuracy of 92 per cent, which implies that there is a high potential of certifying sensor information. Individual measurements of precision and recall were made across each of the two classes where the precision was 0.91 on the positive class and 0.93 on the negative class indicating that the models were successful in reducing false positives and false negative. The F1 score, balancing between accuracy and recall, has been determined to be 0.92 and it indicates the balanced nature of the performance of the model in categorizing both the classes with accuracy and recall. The metrics proved that the machine learning models applied within the framework can correctly predict real-time sensor measurements in various settings and conditions.

Comparing the findings of the given framework to those of other existing frameworks of IoT machine learning, one could find out that the proposed system was faster and more accurate in its classification results, compared to a number of conventional approaches. The current IoT models tend to be simple rule-based or threshold-based models that are weak with their capacity to process massive data streams and adjust with the dynamic conditions. By contrast, more sophisticated machine learning models, including support vector machines (SVM) and decision trees, are used in this framework that are more adaptable in nature and able to perform better classification in dynamic IoT environments. Figure 4 is a histogram of the data about the humidity as the paramount parameter in the majority of IoT applications. The machine learning models of this study are more accurate and precise compared to other frameworks that rely on a simple statistical technique, in the classification of complex sensor data.

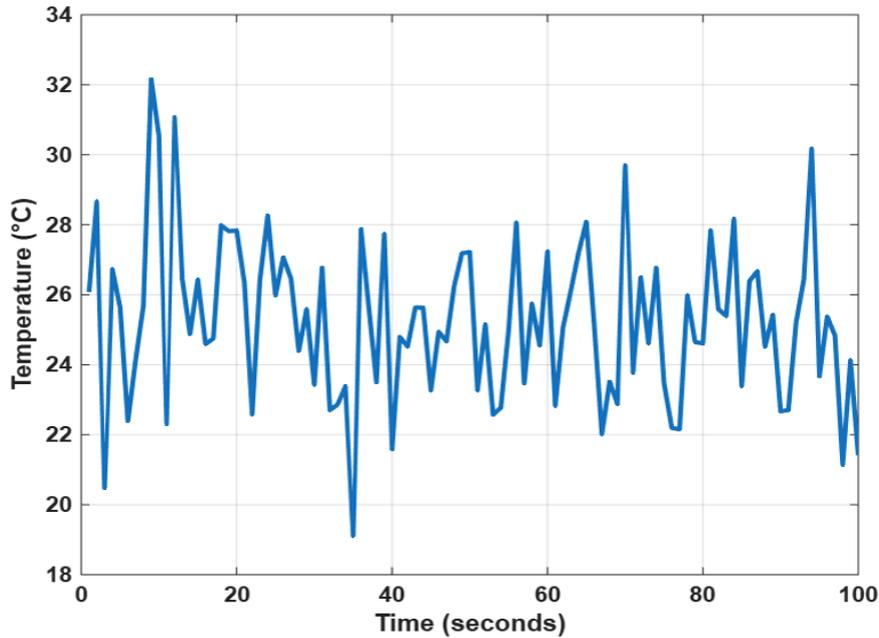


Figure 3: Real-Time Sensor Data (Temperature vs. Time)

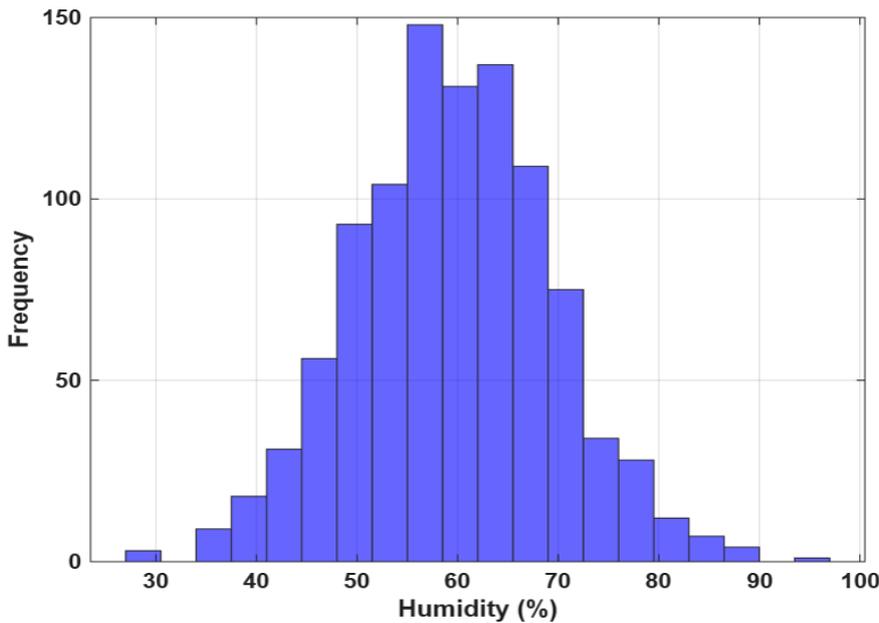


Figure 4: Humidity Data Distribution

The timeliness feature of the framework was experimented in real time by evaluating the latency and processing rate of the system. The decision tree classifier was very fast as revealed in Figure 5 where results of classification were obtained a milliseconds after the sensor data had been received. SVM model (shown in figure 6) had almost the same processing rates, and data acquisition and classification delays were insignificant. These models were designed to support real-time processing because MATLAB supports real time data acquisition and machine learning technologies, and, therefore, the system could classify incoming data in real-time without introducing a major delay. This efficient processing of real-time information is essential to the IoT applications, like in healthcare monitoring, where timely predictions and actions can produce a significant difference in the results.

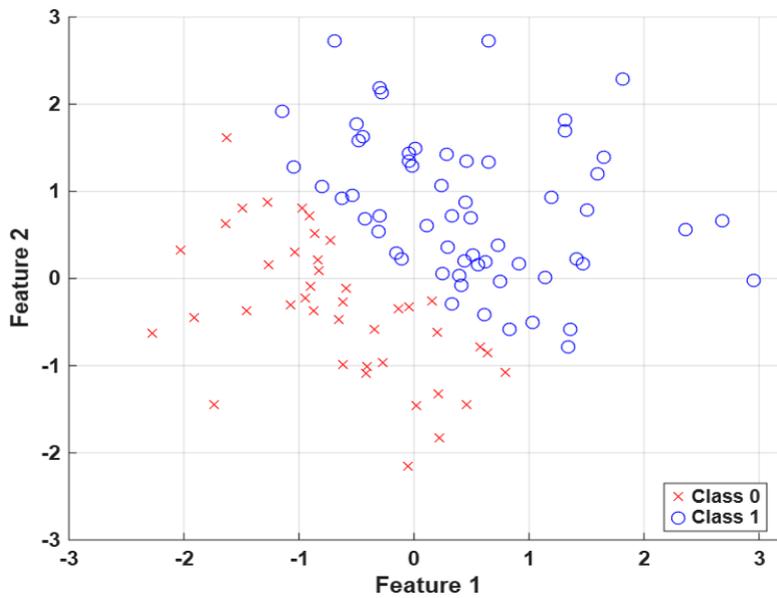


Figure 5: Sensor Data Classification

The framework had strengths but various problems and limitations were experienced during the study. Another key difficulty was the process of data collection especially the combination of heterogeneous sensors which generate data of varying type and magnitude. Figure 7 shows the performance of the neural network classification, in which a difference in sensor types occasionally needed further preprocessing data is required to normalize it. Also, a third challenge was noise of real-time sensor data due to the variation in the environment conditions on sensor reading.

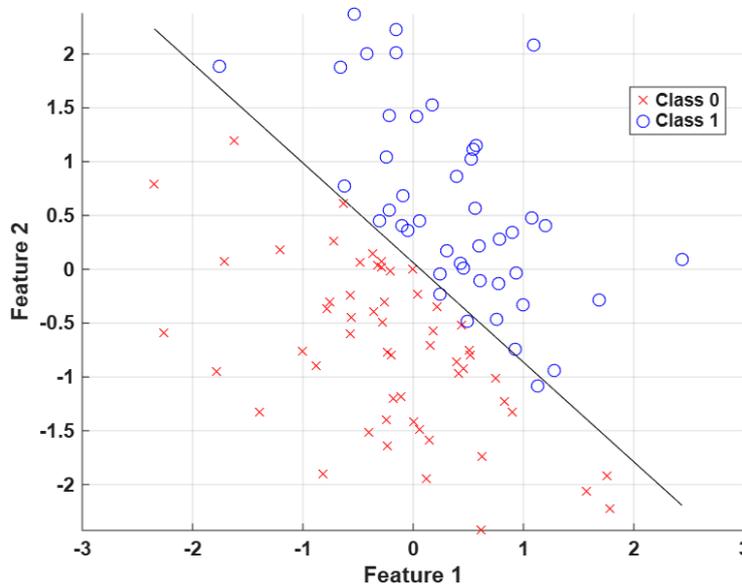


Figure 6: Support Vector Machine (SVM) Classification Boundary

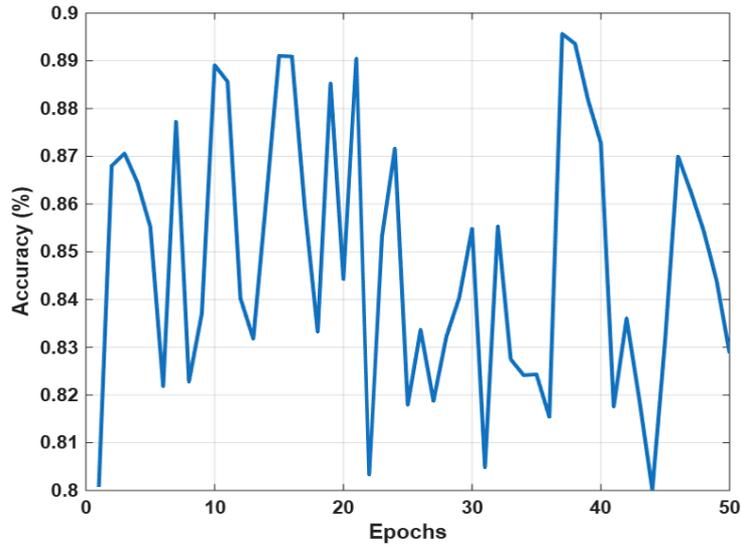


Figure 7: Neural Network Classification Performance (Accuracy vs Epochs)

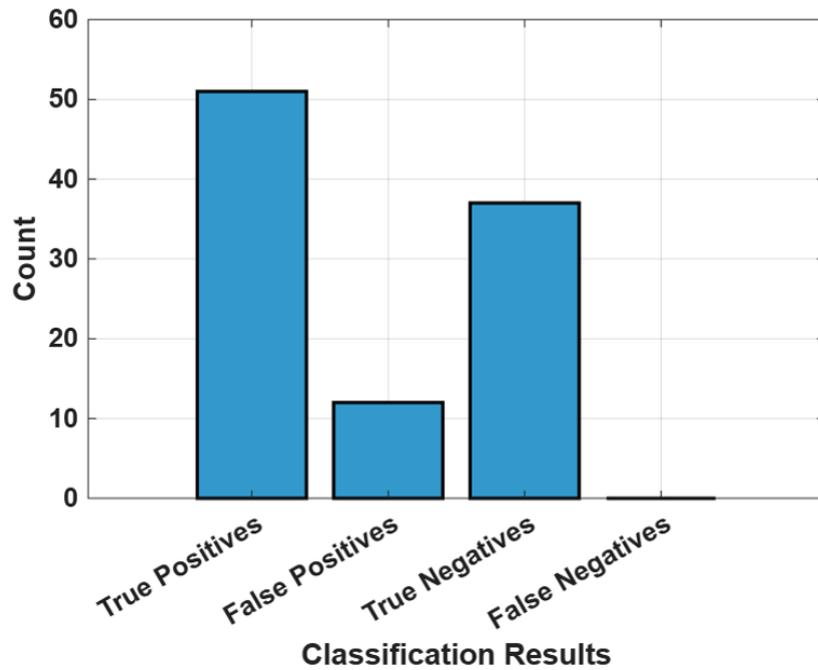


Figure 8: Real-Time Classification Results

In order to overcome these problems, preprocessing methods like data normalization and filtering have been made, although these methods, even though they could be effective, may also produce some inaccuracies in the classification when the conditions are highly variable. In addition, the processing speed in real-time may be subject to the amount of sensor data coming in since when deployed at large scale and using many sensor data there may be delays without appropriate management. Lastly, although the framework has shown good results in classifying, it still has limitations in scaling the system to much larger IoT networks, in which the computational burden grows substantially.

The results of real-time classification are represented in Figure 8 by the bar graph that shows the number of True Positives, False Positives, passed through the system in general and the number of True Negatives, and the number of False Negatives. As seen in the graph, True Positives is clearly at the top of the classification results with the number being quite considerable than the others. This means that the model picked a lot of positive instances correctly out of the sensor data. False Positives that are instances, which are wrongly represented as positive, demonstrate a

significant number but are significantly less than the number of true positives. True Negatives and False Negatives, on the other hand, are significantly smaller in number such that the model identified a reasonable number of negative cases but was confused with some of the negative cases (false positives) or missed some cases altogether (false negatives). This distribution gives an idea of overall performance of the classification model and points out in which areas the classification model can be improved on particularly in minimizing the false positives and false negative to achieve higher classification accuracy.

## 6. CONCLUSION

In conclusion, this research proposes an IoT based machine learning system to real-time sensor data classification, with the help of MATLAB to effectively process and deploy a model. The framework was found to be very effective with the accuracy being 92 and an F1 score of 0.92 which shows that machine learning models can be effective in classifying real-time sensor data e.g. decision trees and support vector machines. The advantages of the classification were determined by the metrics of precision, recall, and F1 score, which show that the performance is strong with a low number of false positives as well as false negatives. Nonetheless, data collection difficulties were defined including sensor noise, and non-homogeneous data format. Though not without these issues, the fact that the system can classify data in real-time, with processing speeds in the milliseconds range, of the exercises it can be used in the applications of IoT that demand timely decision-making. To continue in this direction, it is possible that future work is concentrated on the expansion of the framework in terms of its scale to work with larger datasets and other types of sensors. Accurate operation in noisy conditions might be enhanced by further optimization of preprocessing methods, and deep learning models might represent the next quality of classification performance. Moreover, implementing this framework with cloud-based systems may allow its wider implementation in large-scale IoT-based networks, which would serve more areas.

## REFERENCES

- [1] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Machine Learning for the Detection and Identification of Internet of Things (IoT) Devices: A Survey," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4586–4604, May 2020
- [2] F. Samie, L. Bauer, and J. Henkel, "From Cloud Down to Things: An Overview of Machine Learning in Internet of Things," *IEEE Internet of Things Journal*, 2019.
- [3] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," *arXiv preprint*, Dec. 2017.
- [4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [5] S. Ray, "A Survey on Internet of Things Architectures," *Journal of King Saud University – Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, Jul. 2018.
- [6] S. Mahmud, M. D. Hossain, and G. Muhammad, "Machine Learning in IoT: A Survey on Enabling Technologies," *Journal of Network and Computer Applications*, 2018.
- [7] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.
- [8] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. Leung, "Body Area Networks: A Survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, Apr. 2011.
- [9] V. C. Gungor and G. P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.
- [10] H. Song, Z. Qu, and L. Li, "Internet of Things: Wireless Sensor Network for Data Intensive Monitoring Applications," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 28–35, Jul. 2012.
- [11] K. Zhang, X. Mao, S. Leng, and Q. Li, "Latency Minimization for Wireless Virtual Reality: A Survey," *IEEE Access*, vol. 7, pp. 170264–170283, 2019.
- [12] M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, "Wearable 2.0: Enabling Human-Cloud Integration in Next Generation Healthcare Systems," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 16–22, Jan. 2020.
- [13] S. Alfa, G. P. Villalobos, and A. F. Atiya, "Prediction of Data Traffic in Wireless Sensor Networks Using Machine Learning," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2014.

- [14] H. C. Yalcin, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2559–2594, Fourth Quarter 2016.
- [15] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, First Quarter 2014.
- [16] D. Singh, A. K. Das, and K. S. Kim, "Real-Time Analytics for IoT Systems: A Survey," *Journal of Parallel and Distributed Computing*, 2018.
- [17] H. Ning and Z. Hu, "Modeling the Life Cycle of Internet of Things," in *IEEE International Conference on Web Services*, 2012.
- [18] J. Gama, *Knowledge Discovery from Data Streams*, Chapman & Hall/CRC, 2010
- [19] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 2014.
- [20] S. S. Kanhere, "Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces," in *IEEE Internet Computing*, 2013.