SMART SENSORS FOR ANOMALY DETECTION DRIVEN BY TINYML IN IOT ARCHITECTURE

Eduardo S. Pereira[a], Leonardo S. Marcondes[a], Josemar M. Silva[a]

[a]SENAI, São José dos Campos, SP, 12211180, Brazil.

Abstract: In general, the architecture of Internet of Things (IoT) systems is organized in layers, starting with the perception layer, or physical computing, which involves the sensing process, followed by communication, processing, or middleware layer, application, and business layers. In this paper, we propose a new middleware layer positioned just above the perception layer, by integrating more powerful microcontrollers capable of running artificial intelligence algorithms, known as TinyML. We have developed an algorithm based on extreme value theory for real-time anomaly detection. The results indicate the feasibility of implementing the proposed architecture for real-time monitoring and automated decision-making in industrial systems.

Keywords: Smart Sensors, Artificial Intelligence, Internet of Things, Embedded System.

SENSORES INTELIGENTES PARA DETECÇÃO DE ANOMALIAS IMPULSIONADA POR TINYML EM ARQUITETURA DE IOT

Resumo: A arquitetura de sistemas de Internet das Coisas (IoT) é organizada em camadas, começando pela camada de percepção, envolvendo o sensoriamento, seguida pelas camadas de comunicação, processamento, aplicação e negócios. Neste artigo, propomos uma nova camada de processamento acima da camada de percepção, utilizando microcontroladores mais potentes capazes de executar algoritmos de inteligência artificial para dispositivos de baixo consumo, chamada TinyML. Desenvolvemos um algoritmo baseado na teoria de valores extremos para a detecção de anomalias em tempo real. Os resultados mostram a viabilidade da implementação dessa arquitetura para monitoramento e tomada de decisão automática em tempo real em sistemas industriais.

Palavras-chave: Sensores Inteligentes, Inteligência Artificial, Internet das Coisas, Sistemas Embarcados.

1. INTRODUCTION

When a system operates smoothly without any issues, the monitoring data collected follows a “normal” pattern. However, any deviations from this pattern are considered anomalies. These anomalies indicate a departure from the expected behavior of the system and can be indicative of underlying problems or irregularities. By identifying these anomalies, we can detect potential errors or malfunctions, and take necessary corrective actions.

Anomalies are characterized by significant deviations from the median of a probability distribution based on a set of observations. These events can be analyzed by considering limiting statistical distributions that arise when examining the maximum value from a large collection of randomly observed data points drawn from a specific distribution. [6]. This concept is the central focus of Extreme Values Theory (EVT).

Typically, IoT architecture is divided into layers [1,2,5,7]. A simple architecture with four layers has the following elements: I) Perception layer - consists of data sensors; II) Network layer - receive the useful information in the form of digital signals from perception layer and transmit it; III) Middleware layer - processes the information received from the sensor devices; IV) Application layer - realizes the applications of IoT for all kinds of industry, based on the processed data.

In fog computing, a novel tiny middleware layer is introduced above the perception layer. Leveraging the capabilities of TinyML, local devices possess the ability to make decisions autonomously. Furthermore, beyond the transmission of raw data from the network layer, a significant level of contextual knowledge can be directly conveyed to the application layer. In this study, a novel middleware layer is introduced, positioned directly above the perception layer, integrating advanced microcontrollers capable of executing a family of artificial intelligence algorithms known as Tiny machine learning (TinyML). This middleware layer harnesses the power of TinyML to enable efficient and real-time anomaly detection. Specifically, a TinyML algorithm based on the principles of extreme value theory is presented, allowing for the detection of anomalies in real-time.

By integrating powerful microcontrollers and leveraging TinyML capabilities, the proposed approach enhances the overall functionality and intelligence of the IoT system. The utilization of artificial intelligence algorithms facilitates the identification of anomalies directly at the edge, reducing reliance on centralized processing and enhancing the system's responsiveness and efficiency.

The paper is organized as follows: Section 2 provides an overview of the methodology, including the general concept of the TinyML algorithm for anomaly detection and the proposed IoT architecture. Section 3 presents the results and corresponding discussion. Finally, the concluding remarks and summarizes the key findings of the study are in section 4.

2. METHODOLOGY

According to [3], if the data and a system (or its components) are continuous and bounded, and have multiple failure modes, the anomalies are best modeled by the

Weibull extreme distribution function. The Weibull cumulative distribution function can be written as [4]:

$$F(x) = 1 - e\left[-\left(\frac{x-\eta}{\lambda}\right)^{\kappa}\right], \quad (2)$$

where κ, λ and η are the shape, scale, and location parameters, respectively. As the data is continuously collected in real-time within a defined time window, our study assumes a value of $\eta = 0$. As a result, we obtain the two parameters Weibull Cumulative Distribution Function (CDF). Consequently, we can derive the two parameters of the Weibull CDF through real-time data processing. The estimation of these parameters is performed dynamically based on the incoming data.

The data collected from the sensor is stored in an array of size W and follows a first-in, first-out (FIFO) scheme. This means that when new data is added, the oldest data in the array is automatically removed to make space for the new data.

To determine whether a data point is suitable for model training, to obtain λ and κ parameter of Weibull CDF, it is computed the standard deviation value (σ) each time new data is added to the data window array. If the new data point value is is greater than or equal to a multiple of σ, it is considered appropriate for model training. It should be noted that the data bound is dynamically constructed in response to variations in the characteristics of the data.

To ascertain whether a given data point is classified as an anomaly, we compare the output of the CDF with a threshold value, denoted as δ. Empirically determined, the threshold δ typically assumes a value greater than 0.94. In this scenario, if the result of the CDF calculation is equal to or surpasses the threshold δ, the input data is deemed anomalous.

Figure 2.1 illustrates the TinyML algorithm developed in this study. Upon activating the monitoring system, it enters the learning mode. Once the number of detected extreme data points exceeds N the fitting mode is triggered. Subsequently, the shape and scale parameters of the Weibull CDF are computed. Following this, the system transitions into the evaluation mode, where new data is inputted into the CDF. If the value returned by the function exceeds the threshold δ, it is considered indicative of an anomaly within the data. In such instances, the associated value and the timestamp of occurrence are logged and stored in the anomaly table. If the average time interval between anomaly occurrences is shorter than a specified time parameter, T, the algorithm will trigger an alert, identifying these data points as part of a collective anomaly.

In Table 2.1 is summarized the meta-parameters of the algorithm, determined empirically.

Table 2.1 Meta-parameters of TinyML for anomaly detection.

γ	Parameter of bound values. Used to classify a point as possible to be used in the training to obtain the CDF Weibull parameters.
δ	Threshold of CDF to evaluate if a data point is an anomaly.
N	Total number of maximum data points used to find the shape and scale parameters of Weibull CDF.

T	Average time range among anomalies occurrence necessary to classify the data as collective anomaly.
R	The data time sample ratio.

The architecture of Fog computing, depicted in Figure 2.2, showcases the inclusion of a smart sensor that executes the TinyML anomaly detection algorithm. Within this framework, the sensor data undergoes preprocessing, and the extracted information serves as input for the algorithm. The resulting outcome is subsequently published utilizing the MQTT protocol.

Each individual data from the smart sensor is transmitted to an MQTT broker, facilitating efficient communication. Subsequently, the processed data can be directed towards either a cloud-based platform or a local application, enabling real-time monitoring and automated decision-making capabilities. Detected anomalies can be registered in a blockchain contract. This enables future audit and ensures the integrity of the recorded information.

Figure 2.1 The diagram of TinyML for anomaly detection.

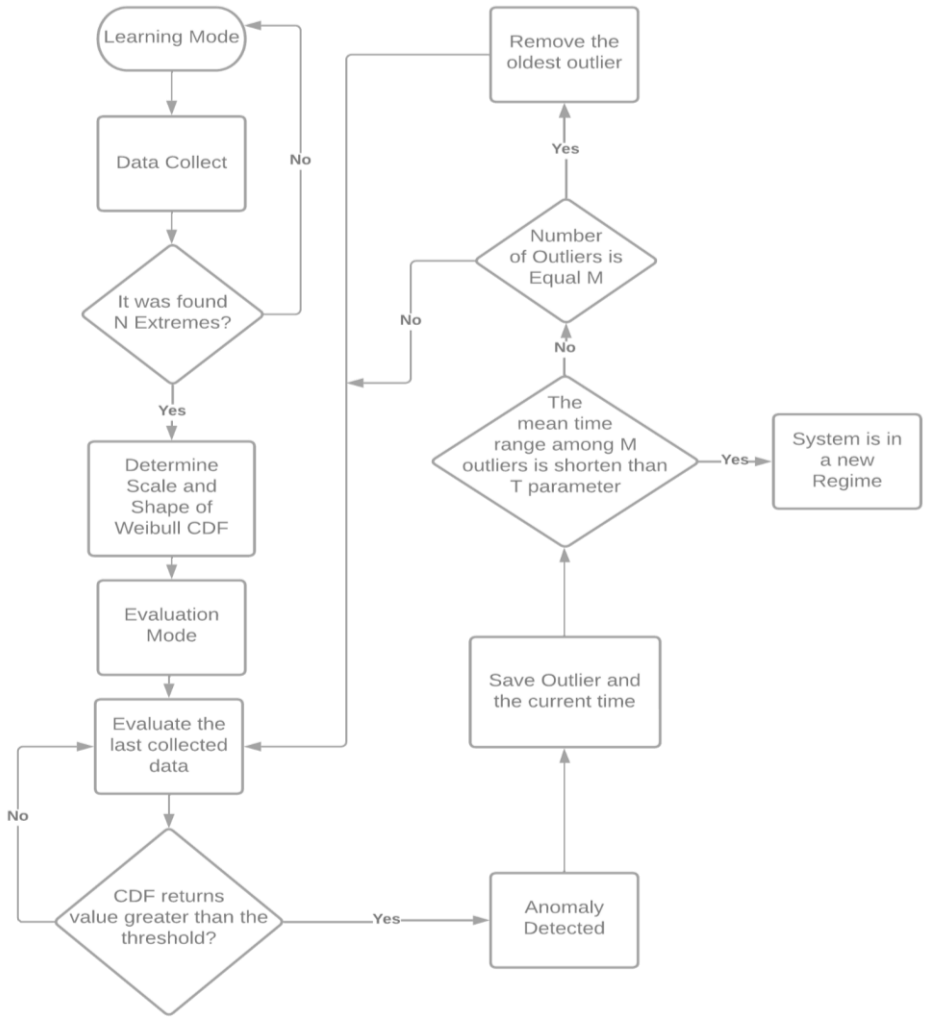


Figure 2.2. Fog Computing Architecture.

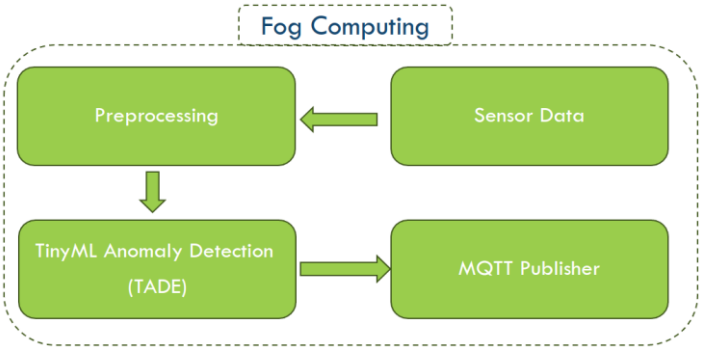
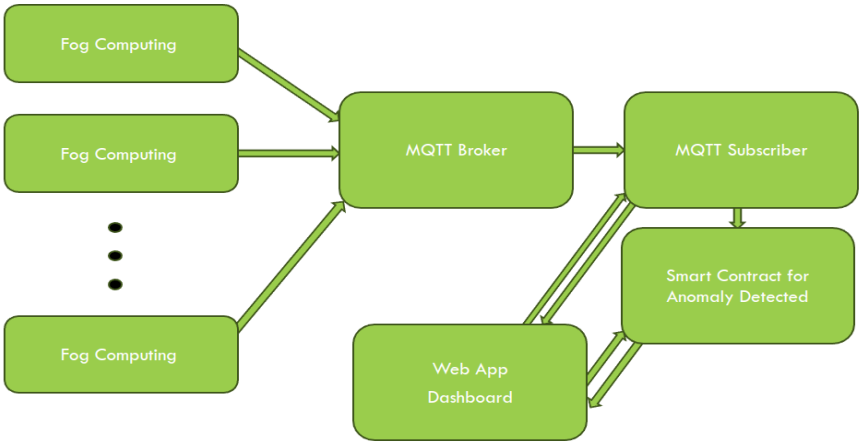


Figure 2.3 illustrates the comprehensive architecture of the anomaly monitoring system, highlighting the integration of these components. This holistic approach facilitates effective data management, analysis, and decision-making processes within the system.

Figure 2.3. System Architecture



In the next section it is presented the evaluation and results of the implemented architecture.

3. RESULTS AND DISCUSSION

In this study, a five-layer IoT architecture was implemented. To enable real-time anomaly detection and processing, a novel and compact middleware layer was introduced directly on the sensor device. This middleware layer was designed to efficiently process and identify anomalies at the edge, enhancing the overall performance and responsiveness of the IoT system.

The anomaly dection algorithm, running on perception device, was implemented using MicroPython and tested on an ESP32-PICO-D4-based board, which integrates an MPU9250 incorporating a gyroscope. The board, powered by the ESP32 system-on-a-chip (SoC), offers advanced features such as Wi-Fi and Bluetooth connectivity.

In the experimental setup depicted in Figure 3.1, the ESP32-PICO-D4 board was securely mounted on an electric motor, enabling the measurement of vibrations using the integrated MPU9250 gyroscope. The collected data was then transmitted to a Raspberry Pi, where a Mosquitto MQTT server was deployed as a broker to facilitate communication between devices.

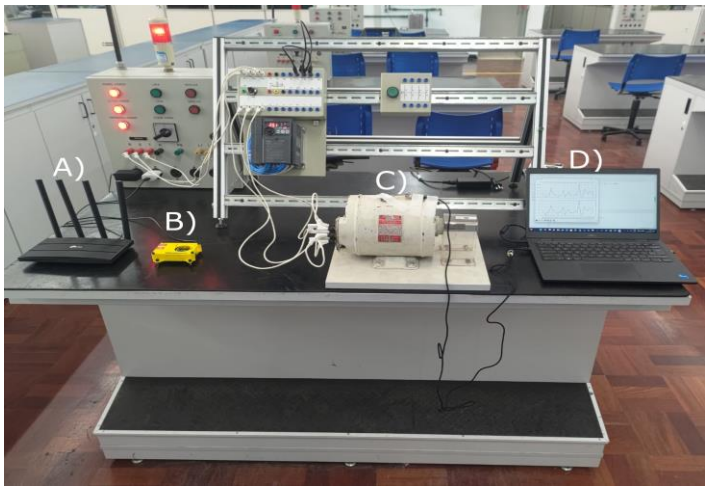
For real-time data visualization and analysis, an application running on a notebook was utilized. This application provided a user-friendly interface, presenting the collected data and enabling real-time monitoring and analysis.

Figures 3.2 and 3.3 present the experimental results in this study. The top of figures depict the absolute difference between the current and previous x-axis values of the gyroscope data. On the bottom of figures is presented the CDF function. The dashed vertical line indicates the point at which the TinyML algorithm learned the parameters of the Weibull function. The diamond markers represent the detected anomalies within the system, and the crosses over the diamonds indicate that these anomalies can be classified as part of a collective anomaly.

The speed of the electric motor was regulated by a frequency inverter. The motor's nominal operating frequency is set at 60 Hz. Anomalies were deliberately induced by modifying the frequency of operation using the inverter. Notably, when the frequency was halved, an increase in motor temperature and alterations in vibration patterns were observed, indicating the presence of anomalies.

Across all experiments, consistent parameters were employed, including a sample ratio (R) of 50 ms, an average time (T) among anomalies, to be classified as collect, of 1000 ms, a threshold (δ) of 0.94 for classifying anomalies, and a fixed number of extreme values (N) set at 5.

Figure 3.1. Experimental Setup: A) Wi-Fi Router; B) Edge Computing with Raspberry Pi server; C) Fog Computing with ESP32 board with embedded MPU; D) Client-side application.



The results clearly demonstrate that variations in γ have a significant impact on the detection of anomalies. Specifically, reducing the value of γ in the bound function resulted in the algorithm identifying relatively minor peaks as anomalies. Conversely, increasing the value of γ led to the algorithm missing certain anomalies, thus reducing overall accuracy. Notably, the increase in γ also resulted in longer training times due to the requirement of a wider observation window to capture data with a higher standard deviation.

Figure 3.2. Experimental Result for $\gamma = 2\,\sigma$.

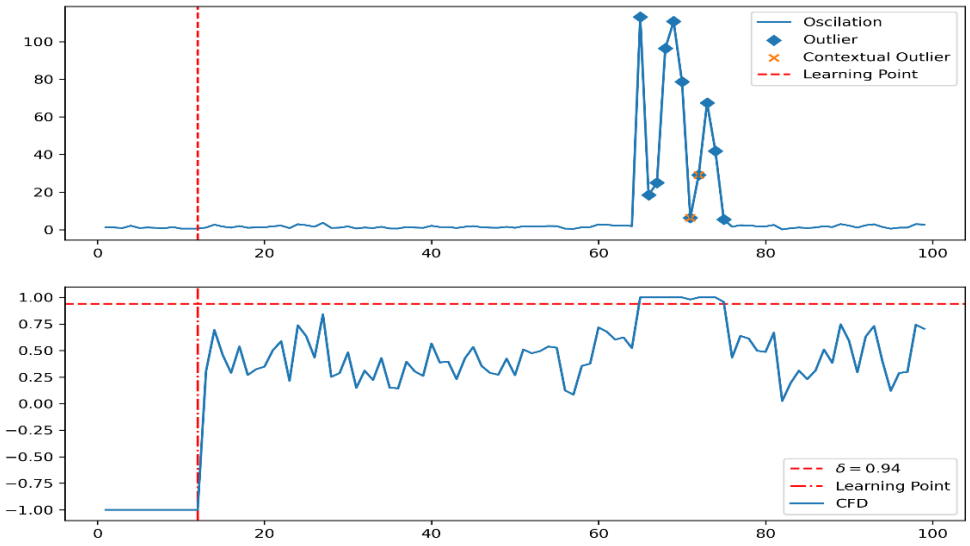
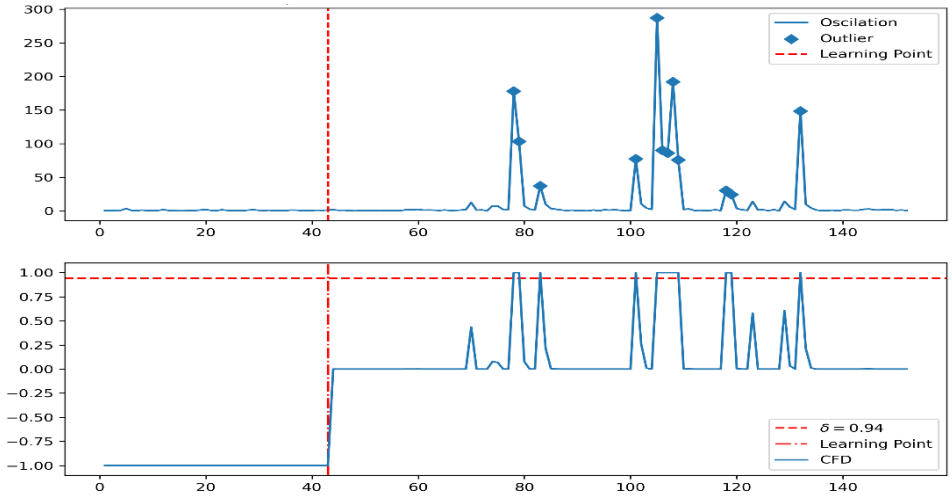


Figure 3.2. Experimental Result for $\gamma = 4\,\sigma$.



These findings underscore the importance of carefully selecting and tuning the value of γ in the bound function to achieve optimal anomaly detection performance. Balancing the sensitivity to detect true anomalies while minimizing false positives and maintaining reasonable training times is crucial in ensuring accurate and efficient anomaly detection in practical applications.

4. CONCLUSION

In conclusion, this work presented a comprehensive investigation into the development and implementation of an IoT anomaly detection system. Through the proposed five-layer IoT architecture, a novel middleware layer was introduced, enabling direct processing and identification of anomalies at the sensor level. This approach enhanced the efficiency and responsiveness of the system by enabling localized decision-making and reducing reliance on centralized processing.

The experimental results demonstrated the effectiveness of the developed system in accurately detecting anomalies in sensor data. By implementing the algorithm at the edge and utilizing the power of TinyML, the system showcased improved efficiency, reduced latency, and enhanced reliability.

Overall, this study highlights the potential of utilizing advanced microcontroller units, middleware layers, and edge computing techniques to enable real-time anomaly detection and decision-making within IoT systems. The findings contribute to the growing body of knowledge in the field of IoT anomaly detection and pave the way for further advancements in IoT applications.

Acknowledgments

The author(s) would like to thank Mr. Fernando Manoel Gonçalves CEO from the São Jose dos Campos SENAI for supporting this work. Also, the SENAI-SP's technology director team for encouraging research and publishing.

5. REFERENCES

1 - BURHAN, Muhammad et al. IoT elements, layered architectures, and security issues: A comprehensive survey. Sensors, v. 18, n. 9, p. 2796, 2018.

2 - FAROOQ, M. Umar et al. A review on internet of things (IoT). International journal 0f computer applications, v. 113, n. 1, p. 1-7, 2015.

3 - GUMBEL, Emil Julius. Statistical theory of extreme values and some practical applications: a series of lectures. US Government Printing Office, 1954.

4 - JIANG, R.; MURTHY, D. N. P. A study of Weibull shape parameter: Properties and significance. Reliability Engineering & System Safety, v. 96, n. 12, p. 1619-1626, 2011.

5 - MRABET, Hichem et al. A survey of IoT security based on a layered architecture of sensing and data analysis. Sensors, v. 20, n. 13, p. 3625, 2020.

6 - SCHEIRER, Walter J. et al. Meta-recognition: The theory and practice of recognition score analysis. IEEE transactions on pattern analysis and machine intelligence, v. 33, n. 8, p. 1689-1695, 2011.

7 - ZHONG, Chang-Le; ZHU, Zhen; HUANG, Ren-Gen. Study on the IOT architecture and gateway technology. In: 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES). IEEE, 2015. p. 196-199.