



Real Time Sensor Data Validation Using FPGA Accelerated Artificial Neural Networks

Manal T. Ali^{1*}, Mohammed Sami Mohammed², Ali M. Al-Jumaili¹, Adham Hadi Saleh¹

¹ College of Engineering-Electronic Department, University of Diyala, Diyala 32001, Iraq

² College of Education for Pure Science-Computer Department, University of Diyala, Diyala 32001, Iraq

Corresponding Author Email: adham.hadi@uodiyala.edu.iq

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.580610>

ABSTRACT

Received: 23 April 2025

Revised: 28 May 2025

Accepted: 20 June 2025

Available online: 30 June 2025

Keywords:

ANN, FPGA, sensor data validation, hardware acceleration, parallel processing, embedded systems

This study suggests an Artificial Neural Network (ANN)-based model implemented on a Field Programmable Gate Array for real-time sensor data validation. The proposed ANN model achieved 96.8% accuracy with a fast training time compared to other Machine Learnings (MLs) of about 0:00:01 sec. The Field-Programmable Gate Arrays (FPGA) design efficiently processed sensor inputs through parallel computation, ensuring high speed, low power, and stable performance. Regression analysis and confusion matrix evaluations were measured to prove the performances which confirmed the strong predictive accuracy with a regression value near 0.93297. In addition, minimum errors were achieved among all other MLs with only 3.2%. The integration of ANN + FPGA enabled reliable real-time validation for embedded sensor networks that are utilized for time-sensitive applications. Future work would be focused on optimizing ANN parameters and exploring hybrid FPGA architectures for improved performance.

1. INTRODUCTION

Sensor validation is considered a critical process for many industrial applications for healthcare monitoring systems. Some conditions such as environmental conditions and aging components affected the sensor's accuracy in spite of their valuable collected data. Systems need to deal with real-time data types with high performances to provide accurate decisions to ensure the reliability of sensor data. It refers to verifying sensors' accurate output and stability against physical parameters regarding any conditions. Enormous readings may lead to false conclusions or even system failures. For example, in the medical field, inaccurate data that has been collected from sensors directly affects the diagnosis process which leads in turn to the wrong treatment. In addition, industrial systems may lead to unsafe decisions with higher costs of the manufacturing process. Sensor validation needs mathematical models that assist readers in specifying the inconsistencies of reading data which may cause these faults in several field types as mentioned by Liu et al. [1]. Sensors have several error types that have a direct impact on reading such as the calibration, which means that sensors should be aligned well with reference values and need to be checked over time. Otherwise, these readings will be affected by the wrong alignment and lead to degrade resulting in drifting values as demonstrated Gano by Gano et al. [2]. Electromagnetic interferences also impact its performances, like inaccurate reading for some sensors affected by extreme temperature conditions as explained by Peng et al. [3]. In the study [4], the authors also explained other error types such as electrical noise and mechanical vibrations which add noising signals to the sensor signals that produce accurate readings as well. Regular

validation is also necessary due to the aging effect that leads to drifting in sensor readings as in the study [5]. The actual values could drift due to these error types and impact decision-making that needs techniques to detect the sensor discrepancies. Traditional techniques are available such as monitoring its readings over time and specifying outlier readings or any other reading deviations from the expected behavior. For example, any readings that fall outside the mean or standard deviation values of the actual readings should be flagged as explained by Seshan et al. [6]. In the study [7], the statistical process was applied for monitoring data over time, which provides simple detection of any outlier's values based on a predefined threshold value. Some techniques are more interested in rule-based approaches to predefine values that may deviate from the originally expected readings, which are mainly based on the knowledge of a specific domain as defined by Jaber [8].

Using similar sensors for the same process as reference points to compare both readings was considered another common method for this validation. This process is applied especially when the calibration values are not available or even difficult to observe as explained in the studies [9, 10]. In spite of some techniques are not suited for real-time reading like the manual calibration process, but it was applied based on known standards. This approach could be time-consuming or even considered a resource-intensive process and cannot be applied for real-time readings. Dealing with large-scale systems in general makes these traditional techniques suffer from limitations. In addition, dealing with real-time reading processes or difficult environments introduces the same issues. As more utilization of sensors for different applications, automated approaches are also needed to provide dynamic and

accurate solutions for sensor validation. Machine learning (ML) has been used and applied as a powerful model for enhancing the accuracy of sensor validation processes but is limited in its ability to deal with large volumes of sensor data in real time. In addition, ML has another limitation with this process by adapting to variations in conditions. ML provides a dynamic alternative that may improve sensor validation by detecting anomalies and sensor failures before they occur. In the field of sensor validation, ML algorithms are utilized to analyze data collected from sensors to detect patterns with identify any deviations that may indicate sensor failure. By learning from these collected historical sensor data and its related feedback, it models predict future behavior related to these types of sensors making ML an invaluable tool in dynamic environments where sensor performance is subject to change. Any outliers that may deviate from the original behavior could be identified by using ML for anomaly detection applications. Based on the study [11], it indicates different sensor drifts with faults by applying various types of ML. In this article, the authors utilized AutoEncoders to compress data first and then reconstruct it. The wrong reconstruction process means that sensors indicate an anomaly. Some papers also deal with ML but using isolation data points that differ significantly from the rest of taken dataset as mentioned by Idowu [12]. Some types of ML are designed for this purpose based on their ability to classify these data depending on a hyperplane that separates the related classes as mentioned by Hinojosa-Palafox et al. [13] for the SVM technique. ML was applied not only for faulty sensor detection but also to reduce the need for manual oversight which led to improving efficiency. An impact field for ML is in predictive maintenance that could be trained to predict if sensors are degraded based on historical data with current sensor performance trends. This allows us to address any potential issues before leading to costly downtime or even data inaccuracies. In the study [14], regression models are utilized to analyze these related sensor data over time and to predict the behavior of sensors in the future. This process and prediction were useful to identify if the sensors need replacement or not based on ML results. Indicating upcoming failures needs models that deal with time series datasets such as LSTM as explained in the study [15], where LSTM is used to analyze the sensor sequential data. The identifications of sensor failures reduce the cost and provide higher operational uptime. In complex sensor networks, multivariate analysis helps to improve sensor validation by considering correlations between different sensors. ML models, particularly ensemble methods like RFT, are used to analyze multivariate sensor data with pattern detection across different sensor types. Alignment reading of connected sensors in a single network should be obtained otherwise; ML is used to compare the related sensor data to flag any drift. ML has also the ability to fuse data collected from multiple sensors across the network to generate a more accurate representation for the desired system that is mentioned by Chen et al. [16] for validation process enhancements. In the same field of sensor validation ML is utilized to validate sensor data by identifying inconsistencies, while for COVID-19 risk factor specification, ML is employed to analyze the redundancy between various features which rely on the ability of ML to process complex datasets that improve decision-making processes as in the study [17]. In the study [18], RFT has been used to develop high-performance prediction systems plus feature minimization to reduce unnecessary data dimensions. Both approaches for sensor

validation and feature minimizations rely on the power of ML to manage large while filtering out noise. The common points were in focusing on enhancing prediction accuracy by optimizing the data input and processing techniques. These innovations demonstrated how ML was applied to diverse fields to ensure better decision-making with reliable outcomes. ML also can provide real-time sensor validation. Traditional validation techniques suffer from the volume of data generated by modern sensors which are widely used now. ML models continuously learn from new data and adjust their predictions dynamically which makes ML good matched for environments where sensor data changes in a rapid manner which is mentioned and explained by Tripathy et al. [19]. Online/reinforcement learning enables ML to adapt new patterns to improve the validation accuracy through real-time datasets. Manufacturing and healthcare are the most common industries that have used ML based on sensor validation field as explained by Ali et al. [20]. ML has revolutionized the field of sensor validation by offering more scalable solutions compared to traditional methods. ML is paving the way for more reliable sensor networks across various industries as explained in the above sections. As sensor technologies continue to evolve, the integration of ML will be essential in maintaining sensor accuracy, particularly in complex and rapidly changing environments.

2. DATA DRIVEN ARCHITECTURE FOR SENSOR VALIDATION

The shift towards data-driven approaches in sensor validation has been fueled by advancements in ML in addition to big data analytics. Traditional sensor validation systems often rely on rule-based algorithms and predefined thresholds. Which fail to handle the growing complexity of sensor networks that currently occur increasingly. On the other hand, this architecture uses data as the primary source for decision-making to continuously validate sensor data in real time. It consists of several key components: data collection/preprocessing if needed followed by ML training then the real-time validation takes the list step and finally feedback mechanisms. These components work together to ensure that sensor data is continuously validated without the need for manual supervision. The first step is to collect data from sensors which is usually in real-time form. These data would be typically collected based on platforms or edge devices to gather from various sensor nodes in a specific network as mentioned by Patil et al. [21]. Preprocessing for the collected data will begin to remove noise and complete the missing values to provide consistency through filtering and different calculations such as normalization. This step was crucial to demonstrate that only high-quality data will be driven into the ML for the validation process as explained by Villegas-Ch et al. [22]. The normal behavior would be identified among the anomalous across various sensor types using the historical data and applying ML as defined and used in the study [23]. When ML has been trained, it will be deployed for real-time sensor validation by checking the incoming sensor data for any anomalous readings. This real-time validation is important for ensuring the reliability of sensor networks, especially in critical applications such as industrial monitoring [24]. This system usually incorporates feedback loops where sensor performance is continually provided. If any anomalous is detected in the sensor readings, the system updates itself by learning from these instances to

improve validation accuracy as shown in Figure 1 as presented in the study [25].

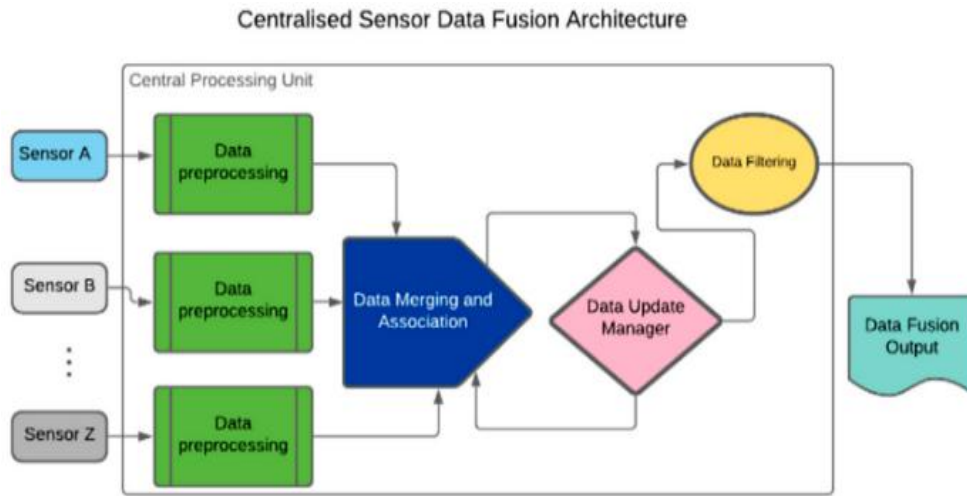


Figure 1. Sensor data fusion architecture which referring to the process management

Benefits of Data-Driven Architecture in Sensor Validation

The transition to a data-driven architecture offers several significant advantages compared to traditional sensor validation methods which can be described as follows:

- Traditional systems suffer when the number of sensors increases.
- It is capable of adapting to sensor evolution or addition in a specific network.
- Traditional methods are often based on static thresholds, which can be ineffective in dynamic systems. While ML on the other hand, continuously improve by learning from the data they process. This allows for more accurate detection of sensor malfunctions making it better than the traditional techniques.
- The ability to monitor sensor performance in real-time is considered one of the important benefits of this system. If sensor performance deviates from expected behavior, it will take immediate action to achieve cost reduction and downtime minimization.

There are several challenges in implementing data-driven sensor validation systems and this could be considered as a gap that should be filled or dealt with. One of the main challenges is ensuring the quality of sensor data which leads to poor performance if the system deals with inaccurate data. Also, increasing calculations due to processing large amounts of sensor data which occurs in real-time, requires higher resources. Edge computing solutions in general are utilized to manage these requirements, but they come with additional complexities and even more costs. A data-driven architecture for sensor validation provides a robust solution to managing sensor networks. By applying and embedding advanced ML techniques with modern computing infrastructure, users improve the accuracy of sensor validation while reducing the required supervision. In spite of challenges related to data quality, the benefit of this validation system is to make this approach essential for modern sensor networks. Authors in different articles such in the studies [26, 27] try to solve the data quality issue by using robust data preprocessing techniques like outlier detection. Techniques like Kalman filters and wavelet transforms are commonly used to de-noise sensor data before it is fed into ML. Also, as sensor networks get more complex, managing and validating data becomes increasingly challenging. Large sensor networks generate a large amount of data that traditional data processing systems

couldn't deal with. Edge/cloud computing allows data to be processed closer to the sending sensor to reduce the need for large-scale data transfer as mentioned in the studies [28, 29]. While Presciuttini et al. [30] utilized Explainable AI (XAI) techniques to provide transparency in ML models by explaining their decision-making process step by step to be understood by organizations for future aspects. In the study [30], Shapley Additive Explanations and Local Interpretable Model-agnostic Explanations are applied to offer insights into the model behavior and to specify the importance of different features in sensor validation decisions. In the study [31], authors employed streaming data processing for managing data when it arrives which allows for immediate validation. The authors used Apache Kafka to implement real-time data pipelines for sensor validation, which is done to ensure low-latency data processing. In addition, optimization techniques, such as model pruning are applied to reduce the computational cost of real-time validation tasks without sacrificing too much accuracy as introduced by Bagwari et al. [32]. Networks include different sensor types that operate under various conditions which may contain different levels of accuracy and different sample rates. These issues make sensor validation more complex which needs to deal with more complex data. A combination of transfer and multi-tasking could be applied to improve the working of ML across different types of sensors. These are used to adapt for application with other sensor types regardless of data limitation. While multi-task learning allows to learn from different sensors simultaneously to improve their ability to handle these issues as explained by Wang et al. [33]. Some issues are also becoming more important such as privacy and security especially when sensors are integrated into critical parts. Authors in the studies [34, 35] presented an explanation of secure ML to address these issues without sharing private data and securing data during processing. In spite of the benefits related to data-driven architectures for sensor validation, several challenges need to be addressed to ensure effective implementation. This article provides real-time processing systems and security measures to overcome these mentioned challenges. As sensor networks grow in scale as well as the complexity of modern sensors added over time to the same networks, the development of robust sensor validation systems will have presented to be crucial for the successful operation of industrial applications.

3. GEOPHONE SG-10

It is a specialized sensor used primarily for detecting ground motion and also to detect vibrations, which is used in oil exploration. It is a type of ground motion sensor that converts these vibes into electrical signals which are measured and written as voltage. The output voltage of these sensors is related to the ground motion and provides valuable data for analyzing subsurface structures. It is designed with advanced earth magnet technology, which provides several key benefits such as giving clean and accurate readings regardless of interference. In addition, it is used in maintaining accuracy in diverse conditions where the geophone suffers from even from its positions and tilt angle. It is designed and selected among other types due to its high versatility and help to tolerate these tilts. It is also built to meet the strictest criteria and to ensure that the sensor provides highly reliable data for any further studies. It was used in different environmental conditions like a variety of temperature conditions in the Arctic or Desert as well. The advanced magnetic design gives it the ability to detect very subtle ground motions, which is important for high-quality data collection.

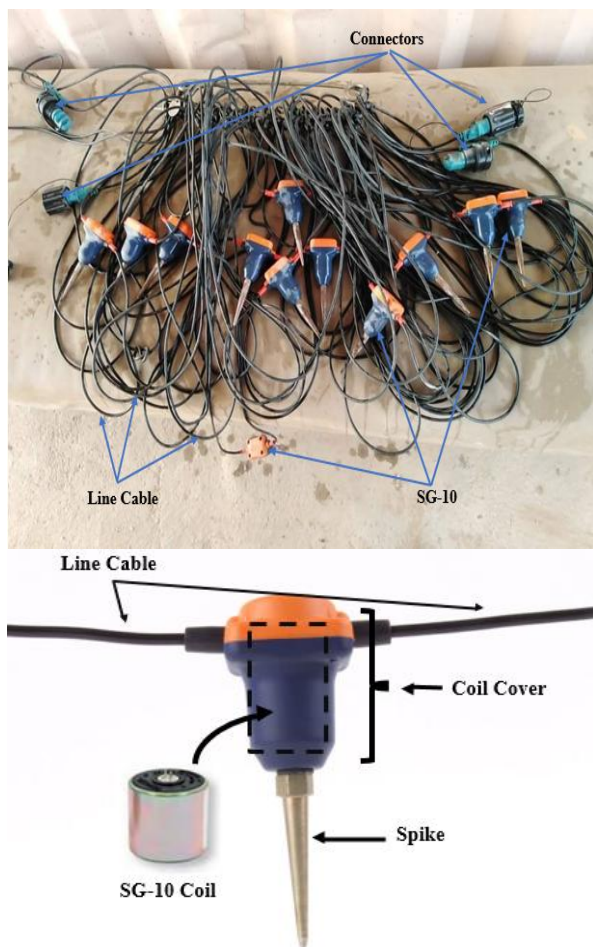


Figure 2. Geophone SG-10 main parts and architecture

It consists of twelve individual geophones which are string-connected in both ways of configurations (parallel and series). This connection setup allows for enhanced performance as the data from multiple geophones is combined to provide an accurate signal. It was one of the reasons to select this type in addition to consistent data collection across the entire string of geophones due to parallel configuration connections. For

series connections, it helps in distributing the load which leads in turn to improve the sensitivity of the sensor system. The different faults of sensors that can be detected by this type include changes in resistance as well as the tilt of the Geophone. In addition to all of these points, it detects unwanted loss of electrical signal that leads to reducing the accuracy of collected data if not well classified. Moreover, the ability to detect high levels of noise which interferes with the detection of ground motion. All of these issues are detected using these Sensor characteristics and by monitoring these factors, it determines whether a geophone is functioning properly or not. Figure 2 shows the main parts and architecture of these sensor types as mentioned in the studies [36, 37]. Regarding the characteristics of the (SG-10) sensors, they were built in a vertical array at fixed intervals to ensure accurate spatial resolution of seismic events. Each geophone was firmly coupled to the ground surface using spike mounts to ensure optimal transmission. During the experiment, data were recorded at a sampling rate of 2 kHz, which is better for capturing high-frequency microseismic signals associated with hydraulic fracturing events. These details have been added to the (SG-10) section to improve experimental transparency.

4. DATASET

The related dataset belonged to fault detection which is used in the field of oil exploration. The primary goal of using this dataset is to detect faults in geophone sensors that have been used to measure vibrations during oil exploration. It includes data from 1232 sensors that contain 587 faults and 645 faults free. The four features utilized in this article based on a dataset for classification include resistance, noise, leakage, and tilt, which provide indicators of the sensor functionality. Table 1 shows the number of samples recorded from sensor readings with related attributes.

Table 1 describes the characteristics of the sample dataset used in this study. The data contains 1,232 samples, including 587 errors and 645 errors-free that described by 4 different features. These data were obtained from microseismic monitoring of hydraulic fracturing operations, and collected as described in the studies [36, 37] for hydrocarbon fields in Ukraine. The seismic sensor data measurement protocols are structured in CSV format, facilitating straightforward processing and analysis.

The related features of the selected dataset contain 8 attributes; the first one is related to a unique identifier that belongs to each point of the network which means a sensor. The second attribute related to the sensor model that was used to collect such required data. The sequence of sensors in a specific network is also identified with a point value. In addition, the condition variations due to many reasons for sensors were affected by the resistance value which was also recorded in ohm. Noise level was also recorded as one of these features due to its main effectiveness on sensor performances. The losing energy was also provided by this data as a leakage measurement feature. Finally, the tilt of the sensor is also considered as one of the impact features on sensor reading. Total sample numbers are 1232 with 587 as a faulty sample (when sensors were not working correctly due to several feature reasons) and 645 free when sensors were working properly as utilized in the studies [36, 37].

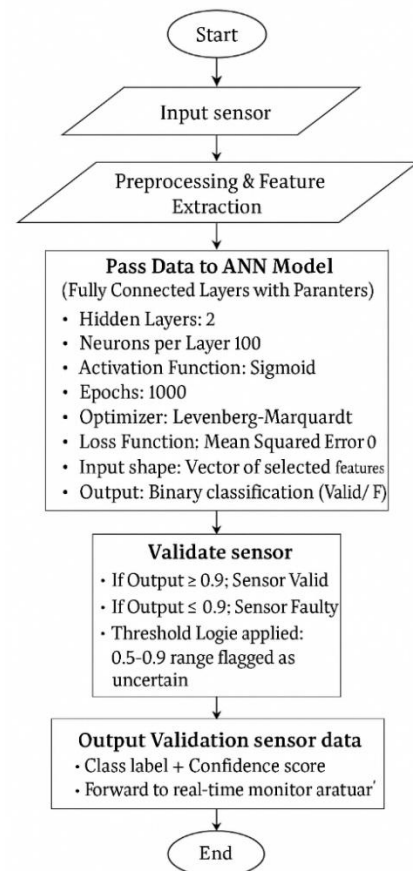
Table 1. The sample dataset characteristics used in this study

Parameter	Description
Number of samples	1,232
False Reading (Faulty Readings)	587
Positive Reading (Fault-Free Readings)	645
Number of features	4
Data Source	Seismic microseismic monitoring data downloaded from: [36, 37].
Data Acquisition Method	Data were collected using microseismic sensors during hydraulic fracturing operations.
Data Format	Structured dataset stored in CSV format, containing readings from 4 features representing seismic sensor outputs and associated parameters.

5. PROPOSED METHODOLOGY

This dataset was analyzed using several techniques in ML to determine the best performance approaches that identify fault readings fully corrected. The non-linear data need to be handled with a nonlinear model such as Naïve Bayes (NB) which applies Bayes rules. Also using a simple technique such as a decision tree to focus on the fast detection process. In addition, a coarse tree (CT) was to applied the dataset. a boosted version of the decision tree combined with Random Under Sampling (RUS) to apply it for unbalancing datasets when data has fewer fault samples. Another state-of-the-art technique was used from Support Vector Machine (SVM) which is fine fine-tuned version that is applied to nonlinear datasets. In addition, Neighbors (KNN) that applied to the dataset. Finally, the proposed method uses ANN that deals with complex patterns like sensor readings. Different type of metrics was measured such as Accuracy, the prediction cost, the speed of the total process, and the training time. This dataset is a valuable resource for exploring fault detection in geophone sensors that are commonly used in oil exploration. By utilizing different ML models, this article aimed to find the most effective method for classifying sensor readings. These techniques were combined with Field-Programmable Gate Arrays (FPGAs) which are considered as a powerful approach to accelerate the computation of the utilized model. This was especially applied for real time in addition to low latency predictions which could be found in embedded systems as mentioned for edge computing. FPGAs provide a high degree of parallelism which allows ML to be implemented efficiently with custom hardware accelerators. The proposed ANN architecture consists of one input layer with 4 neurons corresponding to the input features: resistance, noise, leakage, and tilt. This is followed by two fully connected hidden layers, that containing 100 neurons. A sigmoid activation function is applied in all layers, including the hidden layers and the output layer, to handle non-linear relationships and support binary classification. The output layer contains a single neuron that outputs the classification result, indicating whether the sensor data is faulty or fault-free. The main focus was on the utilization of FPGA and ANN among all other techniques. FPGAs perform operations in parallel across multiple processing units which is useful for the matrix operations in ANN. FPGAs are also more energy efficient than GPUs, especially for applications with a high-power value constraint that is connected to edge points such as mobile devices. In addition, FPGAs allow custom hardware designs, which assist researchers and designers in the same manner to optimize the hardware for the specific ANN architecture. Moreover, FPGAs are often applied for systems that require low latency inference which helps to reduce the time taken for inference significantly compared to traditional ML. Implementing an

ANN + FPGA involves translating the computational graph of ANN into hardware description language (HDL) that will be correctly analyzed and then loaded into the FPGA. The main goal for using ANN + FPGA due to that FPGAs work better with fixed point arithmetic and input data and weights in the ANN are often taken to reduce the complexity and memory usage. For example, instead of using 32-bit floating point numbers for FPGA, it is reduced to 8-bit which leads to reduced power consumption and increases speed in turn. Also, the computation for each layer can be parallelized on an FPGA by using dedicated hardware blocks for convolutions. In addition, one of the critical challenges when combining ANNs + FPGAs is efficiently managing memory. Since FPGAs have limited memory usage due to continuous data streaming and reuse of intermediate results. FPGAs have different stages of computation which have to be processed simultaneously that enable higher throughput. For ANN, pipelining is applied at the layer level to keep the hardware working continuously and maximize throughput. The validation system follows a sequential process, as shown in the flowchart in Figure 3.

**Figure 3.** Flowchart of the validation system based on the main sequential process

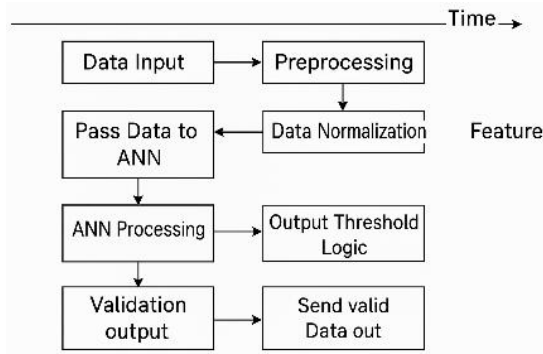


Figure 4. The timing diagram based on the sequence of events during FPGA-based sensor data validation

It begins with the input of raw sensor data, that preprocessed through normalization and feature extraction steps. After preprocessing, the data is passed to an ANN for validation. The ANN determines whether data is valid or invalid based on a threshold value. Then logic step follows, where the output of the ANN is checked to decide if the data meets the required criteria. Finally, validated sensor data is either passed on to complete the additional preprocessing steps or flagged as occurrence errors, completing the cycle of real-time sensor data validation on the FPGA as shown in Figure 3. The timing diagram describes the sequence of events during FPGA-based sensor data validation, explaining the parallel processing of the system. As sensor data arrives, it is directly input into the FPGA. The data passes through preprocessing steps, such as normalization in addition to feature extraction in parallel with the ANN processing. Once the data is processed by the ANN, it is passed through threshold logic to determine whether the

sensor data is valid or invalid. The final validated data is then providing an output value or flagged for errors. Based on these steps in parallel and within a short time frame to ensure minimal time delay, which is impact for real-time applications as also shown in Figure 4.

The design of ANN was in a high-level framework using TensorFlow to be implemented more efficiently on the FPGA. A software library for deploying these models on Xilinx FPGAs that provides tools to compile models for FPGA execution. Using a toolkit (named OpenVINO) allows for deploying ANN on Intel FPGAs to optimize performance models. In addition, High-Level Synthesis (HLS) is used to convert high-level code into HDL for FPGA implementation. This allows for easier implementation of complex algorithms regarding ANN on FPGAs. Another tool used for analyzing and also for optimizing HDL designs for ANN was the Vivado Design Suite. When the design is optimized, it will be converted to VHDL to generate the hardware design from the code. After this step, the design is deployed onto the FPGA. The FPGA will run the inference tasks for the neural network in parallel. The challenges in this application and proposed methodology were that designing ANNs + FPGAs requires knowledge of hardware design with HDL and how to optimize ANN for hardware. the optimization of FPGA architecture needs better code and better time-consuming using ANN as well. Using FPGAs + ANN allows for high performance with low power in addition to the ability of real-time applications. The combination of hardware parallelism and ANN makes FPGAs an attractive option for deploying neural networks in resources which has some limitations and constraints. However, it requires knowledge to fully leverage the potential of FPGAs for ML tasks. The main steps of combining ANN + FPGA are shown in Figure 5.

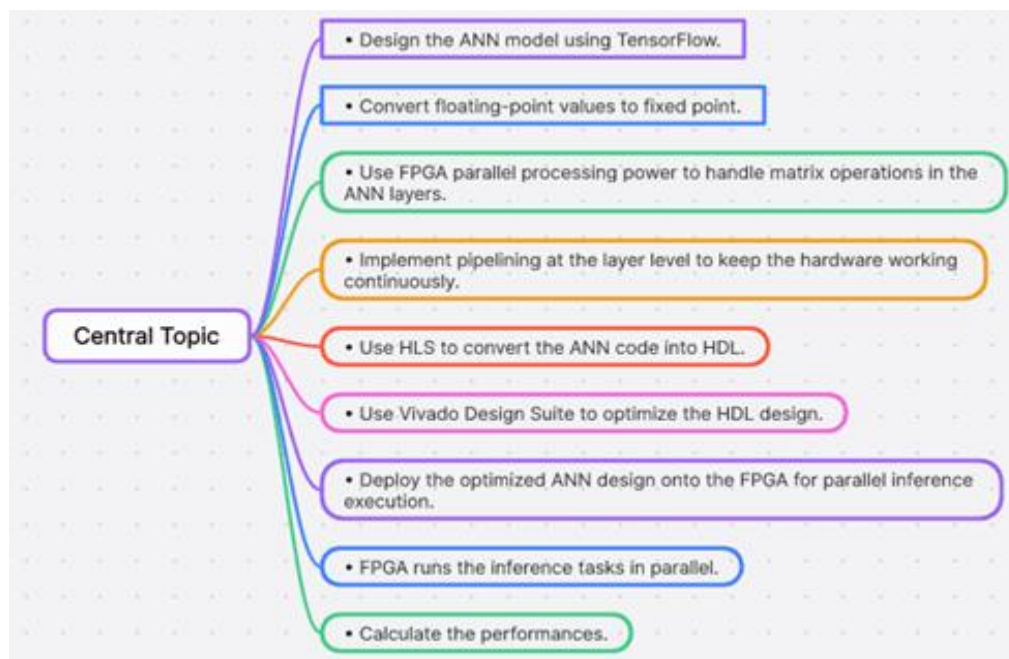


Figure 5. ANN + FPGA. Combination process for validating sensor data

6. RESULTS

Several techniques have been applied to select the best parameters that are related to cost with prediction speed and training time. The observation as shown in Table 2 showed

that ANN-related model 6 had the highest accuracy with about 96.8% and the cost 0.93297 with the good prediction speed under 0.0317 at 1000 epoch for observation numbers in a single second. In addition, it had the fastest training time 0:00:01sec. Model 5 (related to KNN) was also highly

accurate with about 96.1% and had the highest prediction speed of 32,000 for observation numbers in a single second but took slightly longer to train with about 3.0172 sec. However, model 4 which was related to SVM was also highly accurate with about 93.6% and had the highest prediction speed 53,000 for observation numbers in a single second but took slightly longer to train with about 4.597 sec. Model 3 which was related to the RUS boost version of the tree technique performed well but had a slower prediction speed of about 14,000 and had a higher misclassification cost of 129 compared to previous models 4 and 5. however, model 2 related to CT had the highest prediction speed of 19,000 but had a higher cost of about 142. Moreover, the related NB model 1 had the lowest accuracy among all methods with

about 87.5% and the slowest speed equal to 7,600.

Table 3 compares the performance of 6 different ML used for sensor data validation. The ANN model (Model 6) achieved the highest accuracy of 96.8%, with a low misclassification cost and a fast training time of just 1 second. The KNN model (Model 5) also performed well with 96.1% accuracy and the second-fastest prediction speed. The SVM model (Model 4) had a slightly lower accuracy (93.6%) but offered the highest prediction speed of 53,000 observations per second. Although the NB (Model 1) had the lowest accuracy (87.5%) and slowest prediction speed, it required minimal training time. Overall, the ANN model demonstrated the best balance between accuracy, speed, and cost, making it the most suitable for accurate and efficient sensor data validation.

Table 2. ML performances according to geophone sg-10 collected dataset

Model	Accuracy	Cost	Prediction Speed (obs/sec)	Training Time (sec)	Proposed Model	Parameters
3.2	87.5%	154	7,600	4.5316	NB	Gaussian kernel with 4 splits
5.3	88.5%	142	19,000	4.2039	DT	Gini's diversity index
5.25	89.5%	129	14,000	9.6724	RUS	20 splits and 30 learners with 0.12 learning rate
5.12	93.6%	79	53,000	4.597	KNN	Gaussian kernel with 0.5 scale
6.2	96.1%	48	32,000	3.0172	ANN	10 neighbors with Euclidean distance

Table 3. Performance comparison of ML models for sensor data validation

Model No.	Model Type	Accuracy (%)	Misclassification Cost	Prediction Speed (obs/sec)	Training Time (sec)
1	Naïve Bayes (NB)	87.5	2.0	7,600	0.5
2	Classification Tree (CT)	90	1.42	19,000	1.2
3	RUSBoosted Tree	91.5	1.29	14,000	2.1
4	Support Vector Machine (SVM)	93.6	1.1	53,000	4.597
5	K-Nearest Neighbors (KNN)	96.1	1.0	32,000	3.0172
6	Artificial Neural Network (ANN)	96.8	0.93297	31.7	1.0

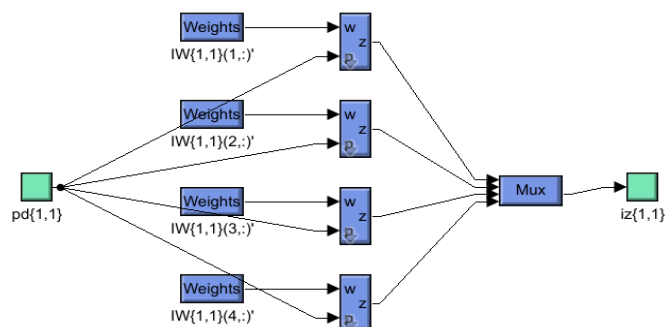


Figure 6. The main ANN architecture of data driven data by sensor validation

Figure 6 shows the ANN architecture which was designed using MATLAB based on ML models and according to the data-driven architecture for sensor validation. Figure 7 indicated the best training performance achieved by ANN and had an MSE value of about 0.031708 at iteration number 1000. This is considering that the difference between the predicted and actual values was so low. Figure 8 provided details about the training state of ANN at epoch 1000 with a gradient value of about 0.023228 which meant that the rate at which the ANN loss function has been minimized. This value meant that the lower value of the gradient indicated that ANN had approached a minimum value for these data. The Validation Checks section indicated points at which ANN performance

on a validation set was checked.

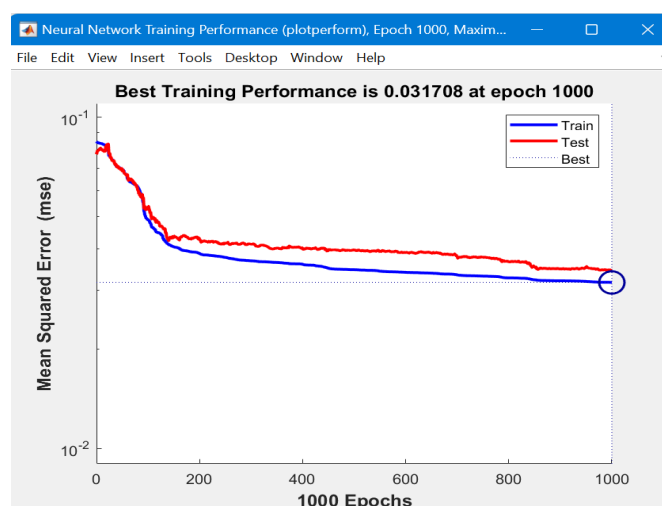


Figure 7. The best training ANN performances based on data driver dataset

Figure 9 shows an error histogram with 20 bins, which determines the distribution of errors across the training and test. This figure showed that the error was near zero error which referred to that ANN was an accurate predictor for the related data. Figure 6 through Figure 9 shows the ANN performances which were measured using MSE and its

training progress has been monitored through gradient values in addition to validation checks. Figure 9 also provided insights into the distribution of prediction errors, which is used to understand ANN accuracy in validating sensor data.

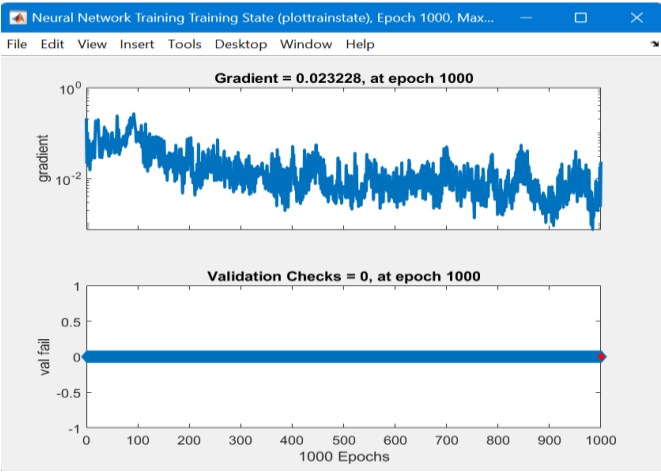


Figure 8. The gradient and validation check for proposed ANN

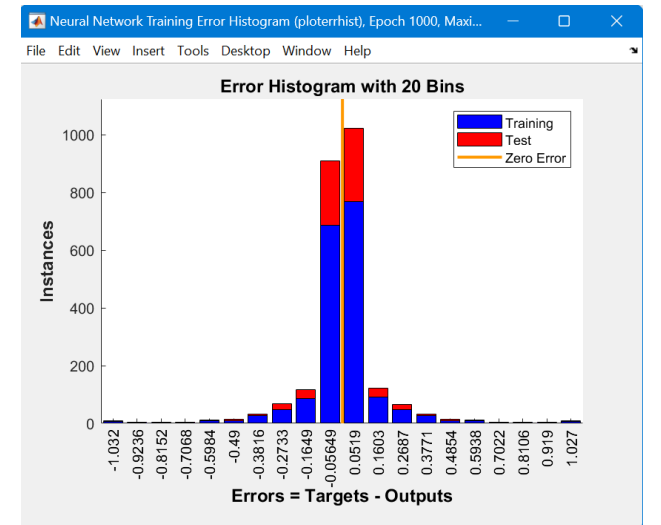


Figure 9. The error histogram for the training/testing based on data driven by sensors

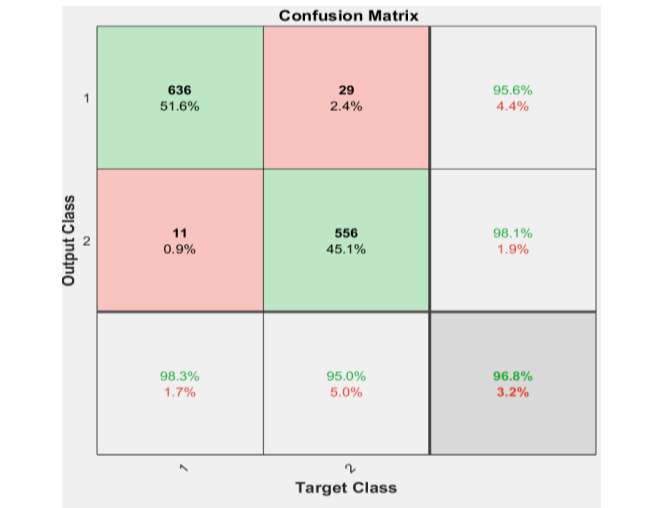


Figure 10. The confusion matrix of proposed ANN model for validating sensor data

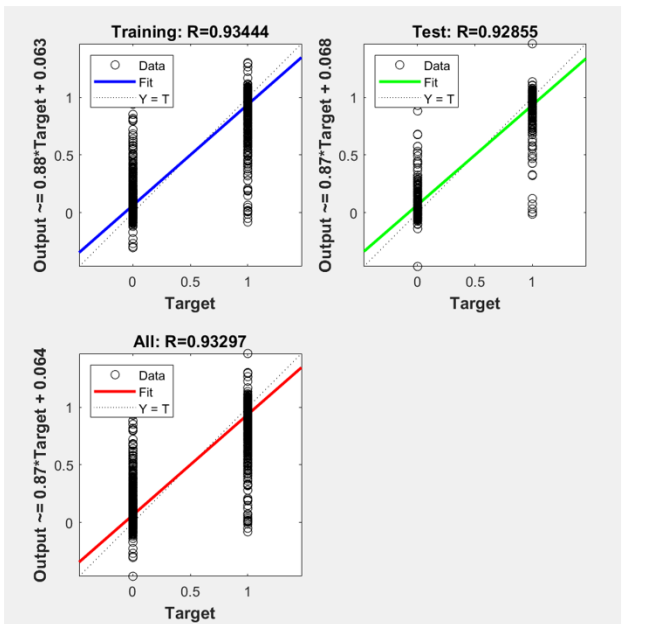


Figure 11. The regression performance plots of ANN model for validating sensor data

Figure 10 shows the confusion matrix that was used for the performance evaluation of ANN classification. It also shows some points that should be noted for the next process related to FPGA design. Figure 10 showed the actual target classes as rows, while the output predicted values were represented in columns. The correct prediction in elements (1,1) and (2,2) in the main diagonal represented the correct predictions that matched the actual classes. The secondary diagonal as (1,2) and (2,1) represented the incorrect predictions where this prediction did not match the actual values. ANN predicted 636 samples correctly as class 1 with 61.6% of the total predictions for this class. Only 29 samples were incorrectly predicted 29 instances as class 2 with only 2.4% of the total predictions. The total accuracy for this class 1 was 95.6% with an error rate of about 4.4%. On the other hand, ANN predicted 556 samples correctly as class 2 with 45.1% of the total predictions, and only 11 samples were incorrectly predicted that had 0.9% of the total class 2 predictions. The total accuracy for this class 2 was 98.1% with an error rate of about 1.9%. The overall accuracy of ANN is about 96.8%, with an overall error rate of about 3.2%. From this matrix evaluation, ANN performed slightly better in class 2 with 98.1% compared to class 1 with 95.6%. ANN's proposed model was performing well overall with high accuracy and low error rates for both classes. In the context of sensor validation, this confusion matrix could have been utilized to evaluate the model performances in classifying sensor data among different states.

Figure 11 contained three subplots that showed regression performance for the ANN during the learning process to evaluate the predicted outputs matching percentage to the target values. This figure represented the best-fit line according to the target value in the x-axis and according to the value of the predicted value. The subplot in the top left was related to training data ANN performances with a regression coefficient equal to 0.93444 R which meant that the model had a strong correlation between predictions and actual values. The left top subplot with green line indications represented the testing data for ANN model performances with R=0.92855, which was lower than training, indicating good generalization. The last subplot figure with red line indicators represented the

overall system performances with a value of 0.93297 which combined training and test data. These values of training, testing, and overall system near 1 indicate strong predictive performances. In Figure 12, the ROC curve was presented to determine the classification performances for the ANN model. In this figure, the false positive coordinator was on the x-axis while the correctly classified sample was on the y-axis. Two lines represented two class classifications; the ANN model represented high accuracy due to the closeness of the top left corner for ROC. It also means that the ANN model effectively distinguishes between classes with minimal false error for both classes. This model based on these results demonstrated a strong regression accuracy with excellent classification ability due to the high value of ROC. It also meant that ANN is well trained and also had a good optimization for the given dataset.

Figure 13 represents a Simulink model for implementing ANN on an FPGA for sensor data validation. This design processed multiple sensor inputs using an ANN implemented in FPGA hardware. The input sensor data was processed through an FPGA pipeline consisting of many steps such as preprocessing and ANN computation. The ANN was structured to analyze the incoming data and determine if it needed any processing to fit the whole process.

The design consists of:

1. Input Blocks that contain:

I. Gateway in blocks were indicated in yellow color which is used to receive data and connect sensors with FPGA logic circuits.

II. Preprocessing blocks which were indicated in blue color that contained gain controllers and feature extraction layers which were important for scaling data before making predictions.

2. ANN Layers that contain:

I. Fully connected layers that were indicated as gray and blue blocks to process sensor data through weighted connections that were trained previously.

II. Sub-system blocks which represent the hidden layer of ANN and consist of neurons to finalize the calculations based on weights and biases. The connection between figure blocks represented the forward propagation in an ANN.

3. Output Processing that contains:

I. Threshold blocks which are indicated in gray color to determine whether the processed data from the ANN meets a validity threshold or not.

II. Gateway out blocks which were indicated in yellow to send the processed and validated sensor data back to the system for another action if required.

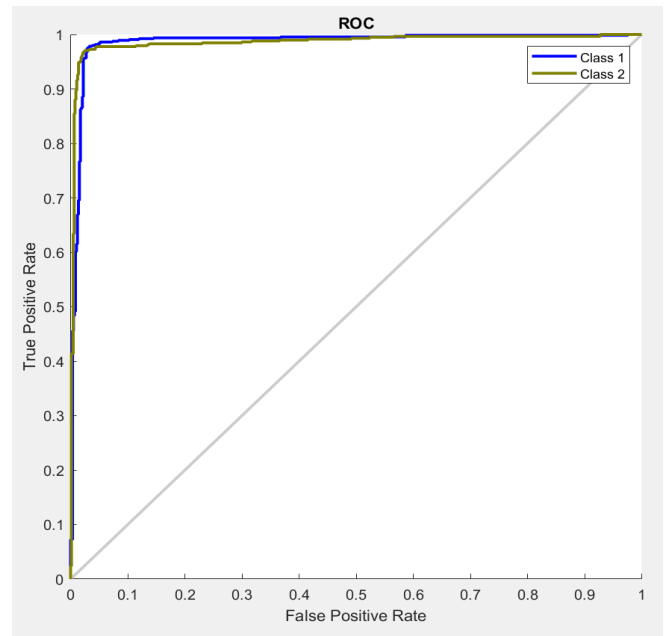


Figure 12. The ROC curve of proposed ANN model for validating sensor data

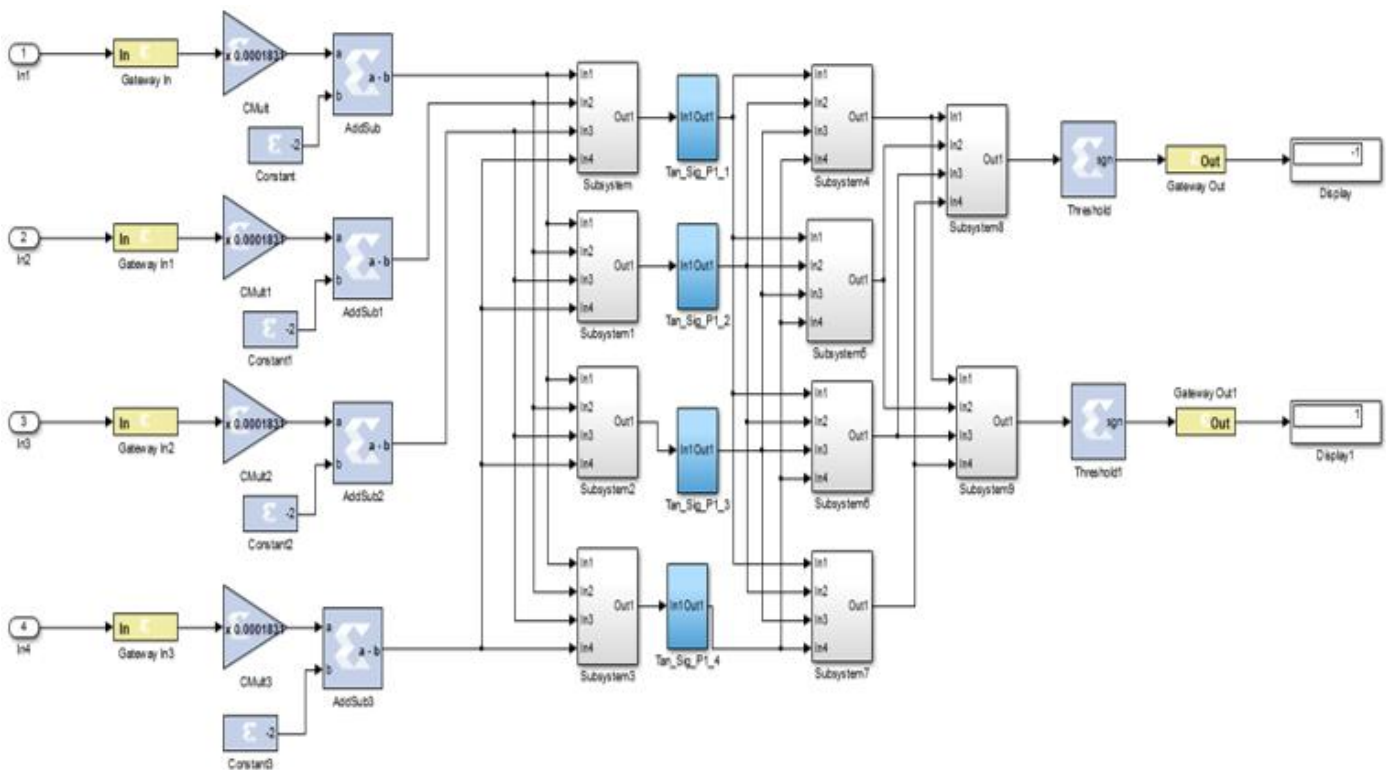


Figure 13. Overview of the FPGA based ANN design for validating sensor data

Table 4. Device utilization summary

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1,538	47,744	3%
Number of 4 input LUTs	15,707	47,744	32%
Number of occupied Slices	9,508	23,872	39%
Number of Slices containing only related logic	9,508 out of	9,508	100%
Number of Slices containing unrelated logic	0	9,508	0%
Total Number of 4 input LUTs	17,122 out of	47,744	35%
Number used as logic	15,695		
Number used as a route-thru	1,415		
Number used as Shift registers	12		
Number of bonded IOBs	50	469	10%
Number of DSP48As	104	126	82%
Average Fanout of Non-Clock Nets:	2.13		

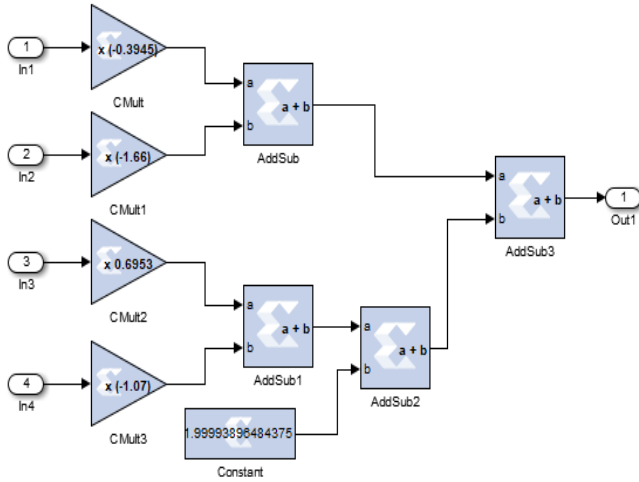


Figure 14. The confusion matrix of proposed ANN model for validating sensor data

Figure 13 details the computational flow of an FPGA-implemented model for sensor data validation. The normalized input data was labeled from In. 1 to In. 4, which represents any type of reading data such as temperature, and vibration signals. The values were determined by scaling mechanism to match the ANN training range and to maintain the high regression accuracy which was near 0.39 as noted in Figure 14. The processing blocks were noted as Computational Modules (CM) to make the multiplications and activation functions from the ANN previously trained layers. In addition, the AddSub. Blocks that combined intermediate results such as merging outputs from parallel ANN layers. The structured flow ensures deterministic processing, crucial for real-time systems where sensor data must be validated based on strict constraints related to time. Figure 13 also has constant blocks with a fixed value to convert normalized ANN output value back to physical value. Moreover, Figure 13 consists of Out.1 which represents the final validated sensor value, if this part deviates significantly from expected sensor behavior the system flags an error. The optimization process of FPGA allowed for efficient hardware mapping.

CM/AddSub. blocks when each block was implemented as a dedicated circuit. The high regression value depends on the computational flow to ensure ANN predictions align with ground truth sensor behavior. Out1. Was also used to distinguish between valid/invalid data reliably based on a threshold value. The parallel CM blocks act as sensor inputs at hardware speeds, which is important for time-sensitive applications. Also, this design eliminated software-related unpredictability in addition to ensuring consistent validation

under different environments like heavy sensor data loads. Also, this design was ideal for embedded systems that required minimum power due to the optimization process. Table 4 shows a summary of the use of the components of the FPGA.

The Xilinx Artix-7 FPGA was selected for this design due to its optimal balance of performance, cost, and power efficiency. It provides sufficient logic resources to implement complex neural network architectures while remaining affordable for research and prototyping. Additionally, the Artix-7 series is well supported by MATLAB and Simulink FPGA design tools, which facilitates low code generation, and hardware verification. This makes it an ideal choice for efficient development and deployment of FPGA-based Artificial Neural Network models.

7. CONCLUSION

This study presented a data-driven approach for sensor validation using machine learning (ML) techniques, with a particular focus on real-time performance, accuracy, and adaptability. A comprehensive analysis of six machine learning models—including ANN, KNN, SVM, and others—was conducted to determine their suitability for validating sensor data in critical applications. Among these, the Artificial Neural Network (ANN) model demonstrated the highest accuracy (96.8%), lowest misclassification cost, and efficient training time, making it the most effective solution for real-time validation of sensor readings. The proposed methodology was designed with scalability and adaptability in mind, addressing common challenges in sensor networks such as noise, drift, aging, and environmental interference. A robust literature review was integrated into the paper, contextualizing the study within existing research and highlighting the limitations of traditional methods. The Geophone SG-10 sensor, known for its sensitivity and resilience in extreme conditions, was also analyzed as a practical application of the validation system. The system demonstrated strong performance, with the ANN model achieving 96.8% accuracy, a low error rate (3.2%), and fast training time. The FPGA implementation ensured deterministic, high-speed processing critical for real-time sensor monitoring. This combination of ANN and FPGA marries the predictive accuracy of machine learning with efficient hardware acceleration, making it highly suitable for embedded and industrial sensor networks. Future work may focus on optimizing ANN parameters and investigating hybrid FPGA architectures to further enhance processing speed and scalability. The novelty of this work lies in combining ANN-based validation with a data-driven

architecture that supports continuous learning, error detection, and system feedback. This framework enables autonomous monitoring with minimal human intervention and is adaptable to complex sensor environments. Future work will focus on implementing this validation architecture in real industrial and healthcare systems, integrating edge computing for faster decision-making, and exploring explainable AI (XAI) techniques to improve transparency in model predictions. Additionally, further optimization of model performance under varying network sizes and real-time constraints will be considered to ensure deployment readiness in scalable sensor networks.

REFERENCES

- [1] Liu, Y., Wang, S., Xie, Y., Xiong, T., Wu, M. (2024). A review of sensing technologies for indoor autonomous mobile robots. *Sensors*, 24(4): 1222. <https://doi.org/10.3390/s24041222>
- [2] Gano, B., Bhadra, S., Vilbig, J.M., Ahmed, N., Sagan, V., Shakoor, N. (2024). Drone-based imaging sensors, techniques, and applications in plant phenotyping for crop breeding: A comprehensive review. *The Plant Phenome Journal*, 7(1): e20100. <https://doi.org/10.1002/ppj2.20100>
- [3] Peng, H., Yang, X., Wu, X., Peng, W. (2024). A wireless temperature sensor applied to monitor and measure the high temperature of industrial devices. *IEEE Transactions on Antennas and Propagation*, 72(6): 5273-5282. <https://doi.org/10.1109/TAP.2024.3389213>
- [4] Nazeer, M., Salagrama, S., Kumar, P., Sharma, K., Parashar, D., Qayyum, M., Patil, G. (2024). Improved method for stress detection using bio-sensor technology and machine learning algorithms. *MethodsX*, 12: 102581. <https://doi.org/10.1016/j.mex.2024.102581>
- [5] Topolov, V.Y., Bowen, C.R., Bisegna, P. (2015). New aspect-ratio effect in three-component composites for piezoelectric sensor, hydrophone and energy-harvesting applications. *Sensors and Actuators A: Physical*, 229, 94-103. <https://doi.org/10.1016/j.sna.2015.03.025>
- [6] Seshan, S., Vries, D., Immink, J., van der Helm, A., Poinapen, J. (2024). LSTM-based autoencoder models for real-time quality control of wastewater treatment sensor data. *Journal of Hydroinformatics*, 26(2): 441-458. <https://doi.org/10.2166/hydro.2024.167>
- [7] Ali, M.T., Abd, B.H. (2022). An efficient area neural network implementation using tan-sigmoid look up table method based on FPGA. In 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-7. <https://doi.org/10.1109/INCET54531.2022.9825348>
- [8] Jaber, A.A. (2024). Diagnosis of bearing faults using temporal vibration signals: A comparative study of machine learning models with feature selection techniques. *Journal of Failure Analysis and Prevention*, 24(2): 752-768. <https://doi.org/10.1007/s11668-024-01883-0>
- [9] Ghemari, Z., Belkhir, S., Saad, S. (2024). A piezoelectric sensor with high accuracy and reduced measurement error. *Journal of Computational Electronics*, 23(2): 448-455. <https://doi.org/10.1007/s10825-024-02134-z>
- [10] Sun, Z., Yao, Q., Jin, H., Xu, Y., et al. (2024). A novel in-situ sensor calibration method for building thermal systems based on virtual samples and autoencoder. *Energy*, 297: 131314. <https://doi.org/10.1016/j.energy.2024.131314>
- [11] Ye, M., Zhang, Q., Xue, X., Wang, Y., Jiang, Q., Qiu, H. (2024). A novel self-supervised learning-based anomalous node detection method based on an autoencoder for wireless sensor networks. *IEEE Systems Journal*, 18(1): 256-267. <https://doi.org/10.1109/JSYST.2023.3347435>
- [12] Idowu, J. (2025). Deploying isolation forest at the edge: A synthetic data-driven approach for real-time IoT anomaly detection. *Artificial Intelligence (AI)*, 1: 6.
- [13] Hinojosa-Palafox, E.A., Rodríguez-Elías, O.M., Pacheco-Ramírez, J.H., Hoyo-Montaño, J.A., Perez-Patricio, M., Espejel-Blanco, D.F. (2024). A novel unsupervised anomaly detection framework for early fault detection in complex industrial settings. *IEEE Access*, 12: 181823-181845. <https://doi.org/10.1109/ACCESS.2024.3509818>
- [14] Gawde, S., Patil, S., Kumar, S., Kamat, P., Kotecha, K. (2024). An explainable predictive maintenance strategy for multi-fault diagnosis of rotating machines using multi-sensor data fusion. *Decision Analytics Journal*, 10: 100425. <https://doi.org/10.1016/j.dajour.2024.100425>
- [15] Selmy, H.A., Mohamed, H.K., Medhat, W. (2024). A predictive analytics framework for sensor data using time series and deep learning techniques. *Neural Computing and Applications*, 36(11): 6119-6132.
- [16] Chen, Y., Wang, X., Zhang, J., Shang, X., Hu, Y., Zhang, S., Wang, J. (2024). A new dual-branch embedded multivariate attention network for hyperspectral remote sensing classification. *Remote Sensing*, 16(11): 2029. <https://doi.org/10.3390/rs16112029>
- [17] Mohammed, S.J., Mohammed, M.S. (2022). COVID-19 risk factors specification using Decision Tree based on the degree of redundancy between features. In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, pp. 1-11. <https://doi.org/10.1109/GCAT55367.2022.9971950>
- [18] Mohammed, S.J., Ahmed, A.M.S., Mohammed, M.S. (2023). Feature minimization for diabetic disorders high performances prediction system-based on random forest tree. *JOIV: International Journal on Informatics Visualization*, 7(3-2): 2032-2039. <http://doi.org/10.30630/joiv.7.3-2.1868>
- [19] Tripathy, J., Balasubramani, M., Rajan, V.A., Aeron, A., Arora, M. (2024). Reinforcement learning for optimizing real-time interventions and personalized feedback using wearable sensors. *Measurement: Sensors*, 33: 101151. <https://doi.org/10.1016/j.measen.2024.101151>
- [20] Ali, M.T., Saleh, A.H., Aziz, H.S. (2024). Efficient detection of brain stroke using machine learning and artificial neural networks. *Mathematical Modelling of Engineering Problems*, 11(12): 3369-3378. <https://doi.org/10.18280/mmep.111215>
- [21] Patil, A., Soni, G., Prakash, A. (2024). Data-driven approaches for impending fault detection of industrial systems: A review. *International Journal of System Assurance Engineering and Management*, 15(4): 1326-1344. <https://doi.org/10.1007/s13198-022-01841-9>
- [22] Villegas-Ch, W., García-Ortiz, J., Sánchez-Viteri, S. (2024). Toward intelligent monitoring in IoT: AI applications for real-time analysis and prediction. *IEEE*

- Access, 12: 40368-40386. <https://doi.org/10.1109/ACCESS.2024.3376707>
- [23] Waqar, M., Bhatti, I., Khan, A.H. (2024). Leveraging machine learning algorithms for autonomous robotics in real-time operations. *International Journal of Advanced Engineering Technologies and Innovations*, 4(1): 1-24.
- [24] Sharma, M., Tomar, A., Hazra, A. (2024). Edge computing for industry 5.0: Fundamental, applications, and research challenges. *IEEE Internet of Things Journal*, 11(11): 19070-19093. <https://doi.org/10.1109/JIOT.2024.3359297>
- [25] Ekerete, I., Garcia-Constantino, M., Nugent, C., McCullagh, P., McLaughlin, J. (2023). Data mining and fusion framework for in-home monitoring applications. *Sensors*, 23(21): 8661. <https://doi.org/10.3390/s23218661>
- [26] Melo, A., Câmara, M.M., Pinto, J.C. (2024). Data-driven process monitoring and fault diagnosis: A comprehensive survey. *Processes*, 12(2): 251. <https://doi.org/10.3390/pr12020251>
- [27] Lin, K.Y., Jamrus, T. (2024). Industrial data-driven modeling for imbalanced fault diagnosis. *Industrial Management & Data Systems*, 124(11): 3108-3137.
- [28] Veeramachaneni, V. (2025). Edge computing: Architecture, applications, and future challenges in a decentralized era. *Recent Trends in Computer Graphics and Multimedia Technology*, 7(1): 8-23.
- [29] Thakur, A., Mishra, S.K. (2024). An in-depth evaluation of deep learning-enabled adaptive approaches for detecting obstacles using sensor-fused data in autonomous vehicles. *Engineering Applications of Artificial Intelligence*, 133: 108550. <https://doi.org/10.1016/j.engappai.2024.108550>
- [30] Presciuttini, A., Cantini, A., Costa, F., Portioli-Staudacher, A. (2024). Machine learning applications on IoT data in manufacturing operations and their interpretability implications: A systematic literature review. *Journal of Manufacturing Systems*, 74: 477-486. <https://doi.org/10.1016/j.jmsy.2024.04.012>
- [31] Dingorkar, S., Kalshetti, S., Shah, Y., Lahane, P. (2024). Real-time data processing architectures for IoT applications: A comprehensive review. In *2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP)*, Bali, Indonesia, pp. 507-513. <https://doi.org/10.1109/TIACOMP64125.2024.00090>
- [32] Bagwari, A., Logeshwaran, J., Usha, K., Raju, K., Alsharif, M.H., Uthansakul, P., Uthansakul, M. (2023). An enhanced energy optimization model for industrial wireless sensor networks using machine learning. *IEEE Access*, 11: 96343-96362. <https://doi.org/10.1109/ACCESS.2023.3311854>
- [33] Wang, D., Wang, Y., Xian, X. (2024). A latent variable-based multitask learning approach for degradation modeling of machines with dependency and heterogeneity. *IEEE Transactions on Instrumentation and Measurement*, 73: 1-15. <https://doi.org/10.1109/TIM.2024.3374288>
- [34] Alwabli, A. (2024). Federated learning for privacy-preserving air quality forecasting using IoT sensors. *Engineering, Technology & Applied Science Research*, 14(4): 16069-16076. <https://doi.org/10.48084/etasr.7820>
- [35] Abdalzaher, M.S., Fouda, M.M., Elsayed, H.A., Salim, M.M. (2023). Toward secured IoT-based smart systems using machine learning. *IEEE Access*, 11: 20827-20841. <https://doi.org/10.1109/ACCESS.2023.3250235>
- [36] Kobrunov, S., Verpakhovska, O. (2024). Методика мікросейсмічного моніторингу гідророзриву пласта для родовищ вуглеводнів України. *Geofizicheskii Zhurnal*, 46(6): 71-80. <https://doi.org/10.24028/gj.v46i6.311666>
- [37] Dean, T., Grant, M. (2024). A beginner's guide to seismic sensors. *Preview*, 2024(230): 38-44. <https://doi.org/10.1080/14432471.2024.2395647>