EDGE COMPUTING ENABLED WIRELESS SENSOR NETWORKS: A CASE STUDY ON HYPERSPECTRAL IMAGE CLASSIFICATION

MOHAMMED, H. R.[1*] – ABDUL JABBAR, J. M.[2]

[1] *Computer Engineering Department, University of Mosul, Mosul, Iraq.*

[2] *Department of Control and Computer Engineering, Almaaqal University, Basra, Iraq.*

**Corresponding author*
e-mail: huthaifa.mohammed[at]uomosul.edu.iq

(Received 10th June 2023; accepted 15th August 2023)

Abstract. This paper evaluates the use of edge computing in wireless sensor networks (WSNs) for hyperspectral image classification in space. Hyperspectral images are computationally demanding and require significant resources. Edge computing can reduce the amount of data transmitted from sensor nodes, thus reducing the overall bandwidth requirements and improving system efficiency. Processing data locally can also reduce latency, which is crucial in extreme environments where communication can be challenging. The SVM, Logistic Regression, and Random Forest algorithms are applied to Indian Pines and Salinas datasets, resulting in six classification scenarios. Software utilizing artificial intelligence algorithms is designed and tested using Google Collaboratory cloud-based platform. Additional dimensionality reduction technique is incorporated and evaluated to enhance classification accuracy. Results indicate that edge computing can improve the efficiency of hyperspectral image classification in space. This research provides valuable insights into the use of edge computing for hyperspectral image classification and has important implications for remote sensing applications in space.
Keywords: *hyperspectral image classification, edge computing, wireless sensor network, dimensionality reduction, machine learning*

Introduction

The development of applications for space exploration has always been an exciting topic, and with recent advancements in technology, it is no longer just science fiction. As humans continue to study harsh environments, one of the most challenging yet fascinating areas is space. Communication between earth station and spacecraft and devices on celestial bodies faces many obstacles, including insufficient power resources, long distances, and limited bandwidth to transfer massive amounts of data. To explore planets deeply, most space rovers are equipped with hyperspectral cameras that provide detailed observations. In this paper, we propose the use of edge computing to process and classify hyperspectral images and evaluate its performance compared to cloud computing. The goal of this research is to address the challenges faced in space exploration and provide insights into the effectiveness of edge computing in processing and analyzing hyperspectral data (Rashvand et al., 2014).

Literature review

Edge computing enabled WSNs

In order to successfully design an edge computing enabled WSN, several criteria must be met, including system security, real-time application support, efficient resource management, energy consumption, system cost, handling of heterogeneous hardware, support for mobility, scalability, support for artificial intelligence, availability, and prevention of malfunctions. Different approaches and techniques to fulfill these criteria has been proposed, including Lockedge for system security, deep learning algorithms for cyber-attack detection, an edge gateway for data security, and a Bayesian optimizer for hardware resource allocation. Additionally, importance of efficient resource management due to the processor and memory limitations of high-end devices in edge computing must be highlighted, and the need to minimize energy consumption in WSNs through a three-layer network architecture or a system for energy consumption minimization. As well as the issues of cost associated with using the data transmission network and the challenge of handling heterogeneous hardware in WSNs (Mohammed and Abduljabbar, 2022).

Hyper spectral image classification

Hyperspectral imaging is a technology that allows cameras to capture numerous spectral data points for each pixel in an image continuously. This has led to a surge in the use of hyperspectral images (HSI) in various applications, such as remote sensing, medical imaging, and material analysis. However, due to the numerous redundant spectral bands, the small number of training samples, and the non-linear connection between spatial location and spectral bands, HSI classification is a difficult task (Govender et al., 2007). In recent years, numerous studies have been conducted to develop and improve classification methods for hyperspectral images. These efforts have led to the development of various machine learning (ML) algorithms, such as support vector machines (SVM), artificial neural networks (ANN), and random forests, among others. Classification is an important research area in hyperspectral imaging, and a significant proportion of research efforts in the field are likely to be devoted to this topic.The effectiveness of ML and DL-based classification approaches has been examined on widely used land cover datasets, including Indian Pines and Salinas Valley. The comparison shows that DL-based classification algorithms perform better than ML-based ones. Spectral-spatial HSI classification is noted to surpass pixel-by-pixel classification since it takes into account both spatial domain information and spectral characteristics (Haq et al., 2022).

In addition, logistic regression has been explored in recent studies as a simple yet powerful algorithm that can handle high-dimensional data and produce interpretable results. Several studies have combined logistic regression with other ML algorithms, such as random forests and ANNs, to improve the accuracy of hyperspectral image classification. The integration of spatial information into the classification process is also an important aspect of recent studies on the use of logistic regression in hyperspectral image classification. Classification is a crucial area of research in hyperspectral imaging, and numerous studies have been conducted to improve the accuracy of classification methods for hyperspectral images. ML algorithms, including support vector machines (SVMs), artificial neural networks, and logistic regression, have been explored to classify hyperspectral images. Logistic regression is a promising algorithm for hyperspectral image classification due to its simplicity, ability to handle high-dimensional data, and ability to produce interpretable results (Gewali et al., 2018).

Materials and Methods

This study evaluates the performance of edge computing in WSNs in space using hyperspectral image classification as a testing algorithm. The use of edge computing can reduce data transmission and improve the efficiency of the system. Indian Pines and Salinas datasets were used to test the performance of these three algorithms. The software was designed using artificial intelligence algorithms and tested using Google Collaboratory. This can be valuable in extreme environments where communication can be challenging due to distance, interference, and limited power resources. We developed multi-algorithmic software that classifies hyperspectral images, allowing users to select the appropriate algorithm and database to obtain desired results. The software generates a classification report, visualizes the dataset bands, and provides a confusion matrix and a resultant classified image. Our software accurately classifies different types of crops, buildings, trees, and steel towers, demonstrating excellent performance in terms of accuracy and speed. The procedures for classification are indicated in *Figure 1*.

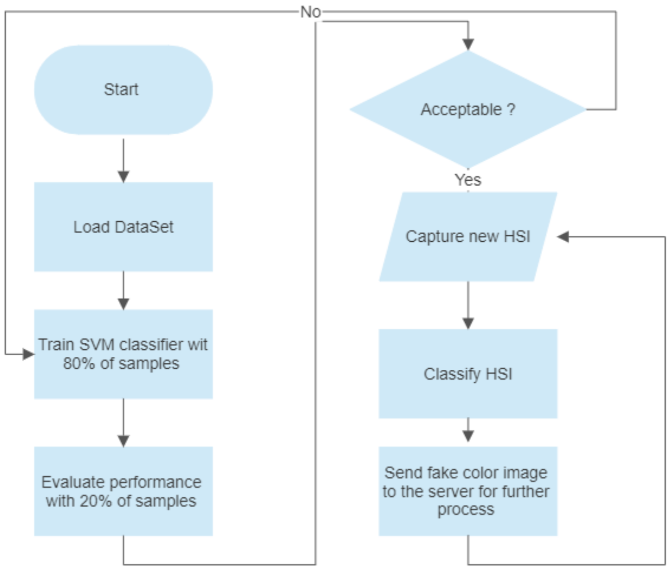

Figure 1. *ML algorithm classification procedures using cross validation.*

Classification algorithms

Support Vector Machine (SVM) is a popular supervised learning algorithm used for classification and regression tasks. It works by finding a hyperplane that separates the data into different classes while maximizing the margin between them. SVMs are known for their ability to handle high-dimensional data and are effective in solving non-linear problems by using kernel functions. SVM is a powerful algorithm for binary classification tasks, but it can also be extended to multi-class classification problems (Müller and Guido, 2016). Logistic Regression is a statistical algorithm used for binary classification tasks. It estimates the probability of an event occurring based on input variables. Logistic regression uses a sigmoid function to map any input value to a value between 0 and 1, representing the probability of the event occurring. The goal of logistic regression is to minimize the error between the predicted and actual outcomes. Logistic regression is a simple and efficient algorithm that can handle noisy data and works well with a large number of features (Müller and Guido, 2016). Random Forest Classifier is an ensemble learning algorithm that combines multiple decision trees to improve the accuracy and reduce overfitting. Random forest creates a set of decision trees, each trained on a subset of the features and data samples, and then aggregates the predictions of all the trees to produce the final result. Random forest is a versatile algorithm that can handle both classification and regression problems and works well with high-dimensional and noisy data. It also provides feature importance scores, which can help identify the most important features in the dataset (Breiman, 2001).

Dataset

Indian Pines Corrected Dataset is a popular hyperspectral dataset that was collected using the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. The dataset contains 145x145 pixels with 224 spectral bands, making it a high-dimensional dataset with a total of 32,560 features. The dataset was collected over an agricultural area in Indiana, USA, and consists of 16 different land cover classes, including crops, trees, and bare soil. The Indian Pines Corrected Dataset is widely used in the remote sensing community for testing classification algorithms and evaluating their performance. The dataset poses several challenges, such as the presence of noise, atmospheric effects, and the high dimensionality of the data (Huang et al., 2002). Salinas Dataset is another popular hyperspectral dataset that was collected using the AVIRIS sensor. The dataset contains 512x217 pixels with 224 spectral bands, making it a high-dimensional dataset with a total of 115,344 features. The dataset was collected over an agricultural area in Salinas Valley, California, and consists of 16 different land cover classes, including lettuce, broccoli, and bare soil. The Salinas Dataset is widely used for testing classification algorithms and evaluating their performance. The dataset poses several challenges, such as the presence of noise, atmospheric effects, and the high dimensionality of the data (Uno et al., 2005).

Both Indian Pines Corrected Dataset and Salinas Dataset are widely used in the remote sensing community for testing and evaluating classification algorithms. The datasets pose several challenges due to their high dimensionality and the presence of noise and atmospheric effects. Several classification algorithms, including Support Vector Machines, Random Forests, and Logistic regression, have been tested on these datasets, and their performance has been evaluated using various metrics such as accuracy, kappa coefficient, and F1-score. These datasets are valuable resources for researchers and practitioners who are working on developing new algorithms for hyperspectral image classification and related applications.

Personal component analysis

Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction and feature extraction in hyperspectral data analysis. The goal of PCA is to transform the original data, which typically has a high number of features or dimensions, into a new set of variables or principal components (PCs) that capture the most significant information in the data while retaining as much of the variability as possible.

Rassberry pi 3 and USB current meter

Raspberry Pi 3 is a credit-card-sized single-board computer developed by the Raspberry Pi Foundation. It is a low-cost, low-power device that can be used for a wide range of applications, including edge computing in Wireless Sensor Networks (WSNs). The Raspberry Pi 3 is equipped with a quad-core ARM processor, 1GB of RAM, and various interfaces such as Ethernet, Wi-Fi, and Bluetooth. Raspberry Pi 3 is an ideal platform for edge computing in WSNs because of its small size, low power consumption, and ability to run Linux-based operating systems. The Raspberry Pi is capable of running Linux-based operating systems. Raspberry Pi can work in space if the necessary precautions are taken to ensure its proper functioning in the harsh environment. To operate a Raspberry Pi in space, it would need to be equipped with radiation-hardened components and a ruggedized housing to protect it from the extreme temperature fluctuations and dust storms on the planets like Mars. It would also require a power source that can operate in the low-oxygen environment, such as a nuclear power source or solar panels , It has been used in various space missions, including the International Space Station (ISS) and the Hubble Space Telescope (Nguyen et al., 2022). USB current meter is a device that used to measure the current flowing through a USB port. Raspberry Pi 3 can be used as a gateway device that can process the data collected from the sensors and perform local analytics. Edge computing within Raspberry Pi 3 can help in reducing the latency and bandwidth requirements of the network by processing the data locally and only transmitting the relevant information to the cloud or central server.

Results and Discussion

Classification performance

The SVM, Logistic Regression, and Random Forest algorithms were applied to Indian Pines and Salinas datasets, resulting in six classification scenarios as indicated in *Table 1*. Our software has been demonstrated excellent performance in terms of accuracy and speed. The software is implemented using Google Collaboratory, which offers a user-friendly interface and enables seamless collaboration and sharing of results. For each scenario, we calculated the accuracy and classification performance using a confusion matrix and classification report, as well as the kappa coefficient, and compared the results. The comparison of the results is presented in *Figure 2a* and *Figure 2b*.

Table 1*. Six classification scenario*

Algorithm DS	SVM	Logistic	Random forest
Indian pines	Scenario 1	Scenario 2	Scenario 3
Salinas	Scenario 4	Scenario 5	Scenario 6

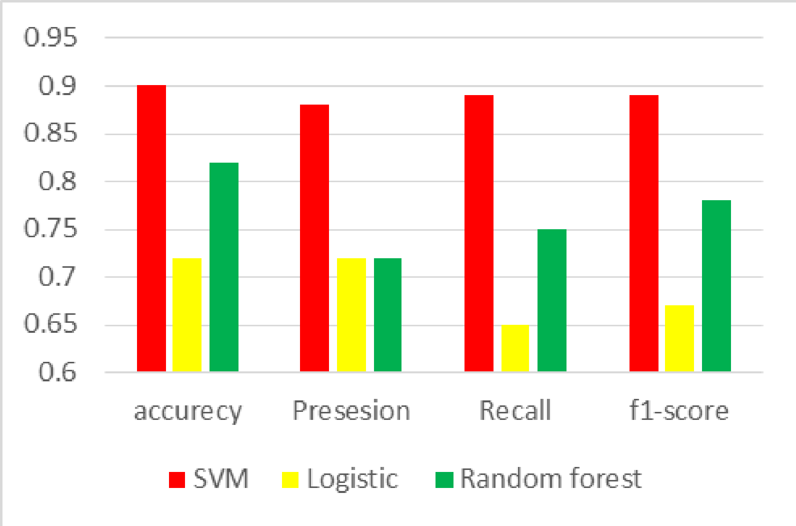

Figure 2a. *A comparison between applying different ML algorithms on Indian Pines dataset.*

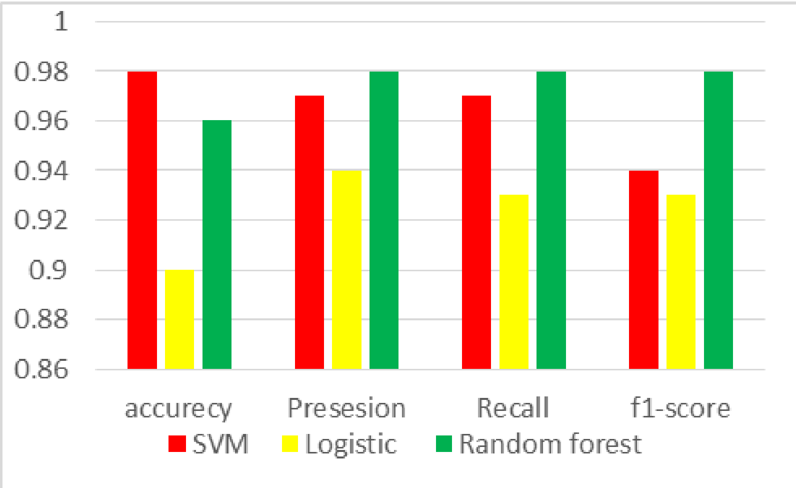

Figure 2b. *Comparison between applying different ML algorithms on Salinas dataset.*

In *Figure 3a*, *Figure 3b*, and *Figure 3c*, SVM, logistic regression, and random forest classification algorithms were applied to different datasets, resulting in varying levels of accuracy, precision, recall, and F1-score. SVM performed better than the other two algorithms in terms of accuracy and F1-score, while logistic regression had the lowest performance due to its assumptions about linearity and independence. Random forest could have performed better with increased trees or optimized hyperparameters. The choice of algorithm depends on the data, study goals, and performance/interpretability trade-off. For the Salinas dataset, SVM outperformed logistic regression and random forest with an accuracy of 0.98, precision of 0.97, recall of 0.97, and an F1-score of 0.94. SVM is well-suited for complex and high-dimensional datasets, while logistic regression and random forest had relatively good performance. The better performance of ML algorithms on Salinas compared to Indian Pines due to the former's higher number of training samples and less noisy data, as well as more distinctive and separable spectral signatures. Class separability, data quality, dataset size, and feature selection are critical factors in classification algorithm performance (Molnar, 2022). Well-separated classes, high data quality, and larger datasets can improve accuracy, while poor data quality and small datasets can lead to misclassification and reduced accuracy.

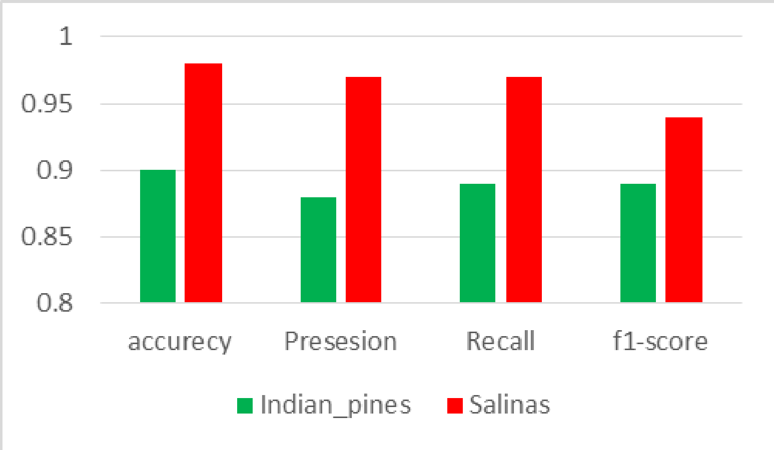

Figure 3a. *A SVM classification performance with different datasets.*

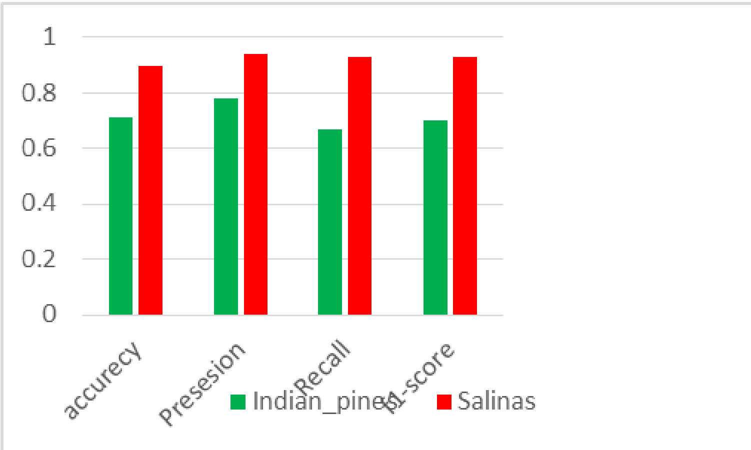

Figure 3b. *Logistic regression classification performance with different datasets.*

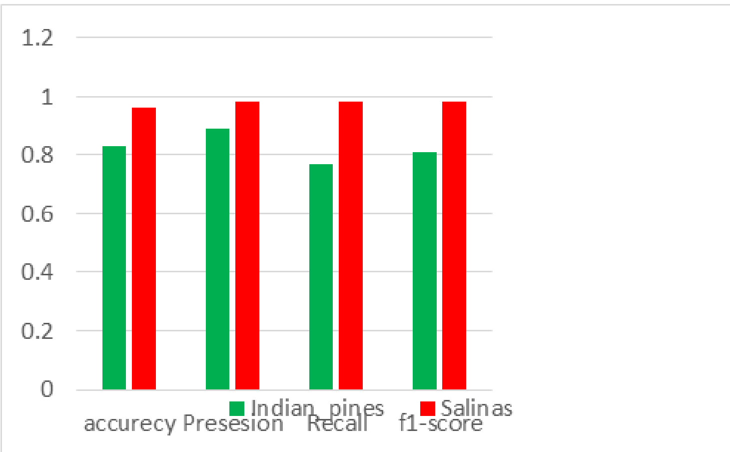

Figure 3c. *Random forest classification performance with different datasets.*

Dimensionality reduction

Principal Component Analysis (PCA) is a popular method for feature reduction in machine learning, which eliminates redundant or irrelevant features and improves model performance. It identifies the directions of maximum variation in the data, called principal components, and projects the data onto them. We determined the optimal number of components using a scree or cumulative variance plot. We assessed the impact of different numbers of components on accuracy and choose the optimal number that balances accuracy and efficiency (*Figure 4*).

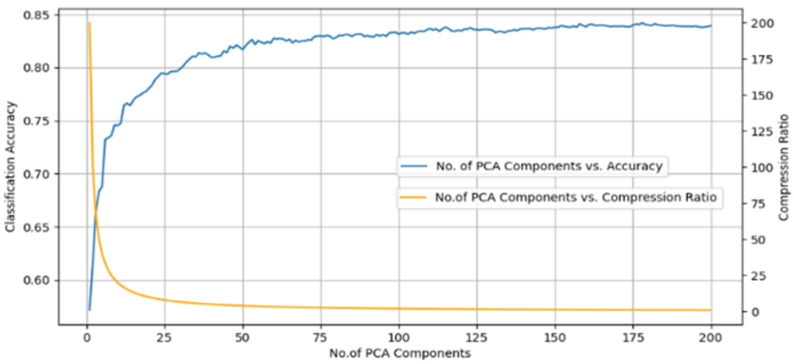

Figure 4. *Component number versus accuracy.*

WSN performance

Proposing that we have a space ship that is equipped with a camera and sensor node(s) that are fixed in place. This spaceship has traveled far away from Earth to capture a hyperspectral image of a particular region that scientists want to study. In order to evaluate the performance of algorithms and wireless sensor networks (WSN) within the space ship, we use two images-the Indian Pines and Salinas. The use of hyperspectral imaging and wireless sensor networks in space research is an exciting field with a lot of potential for discovery and innovation. As technology continues to advance, we can expect to see even more powerful tools and techniques developed to help us explore and understand the universe around us. Once a Hyperspectral Image (HSI) is captured, it must be classified to extract meaningful information. This classification can be done either in a space station or on an earth station. To ensure that the classification process can be executed efficiently on low-resource platforms, we have chosen three algorithms for classification: Support Vector Machines (SVM), Logistic Regression, and Random Forest. SVM is a supervised learning algorithm that uses a set of training examples to classify new examples. It works by finding the best hyperplane that separates the different classes in the feature space. Logistic Regression is also a supervised learning algorithm that predicts the probability of an example belonging to a certain class. It uses a logistic function to map the input variables to the output. Random Forest, on the other hand, is an ensemble learning algorithm that uses multiple decision trees to classify examples. Each decision tree is trained on a subset of the data, and the final classification is based on the majority vote of all the trees.

These three algorithms have been chosen because they are suitable for execution on low-resource platforms. SVM and Logistic Regression are relatively simple algorithms that require only a small amount of memory and computation power, making them ideal for resource-constrained environments. Random Forest, while more complex, can also be implemented efficiently on low-resource platforms because the decision trees can be constructed in parallel. These algorithms provide a robust and efficient solution for classifying HSIs in space or on Earth. We explored two potential scenarios for Hyper Spectral Image (HSI) classification. The first scenario involved utilizing edge computing, where the captured image would be processed at the edge, specifically at the cluster head. This approach allowed for real-time processing and analysis of the image data, with the results being transmitted to the cloud for further analysis and refinement. The second scenario we considered involved sending the captured HSI data in its raw format directly to the cloud for classification. While this approach may have been less efficient in terms of processing time, it had the advantage of being more accurate due to the cloud's greater computational power and access to advanced ML algorithms. The choice of which scenario to utilize would depend on the specific needs and requirements of the application. For applications where real-time processing and low latency are critical, edge computing may be the preferred approach.

Figure 5 illustrates the process that data would follow to reach the cloud for further processing. The initial step in this process is to capture data using a Hyperspectral camera. Unfortunately, this type of camera was not available for use in our study, so we instead used images that had been previously captured, specifically, the Indian pines and Salinas images. Once the images are captured, they can be either processed locally or sent to the cloud in their raw format for further analysis. This decision is often dependent on the available resources, as well as the requirements of the specific application. In the first scenario, the program is split into two distinct parts. The first part is dedicated for running classification algorithm that analyze the data and provide classification results. The second part of the program is responsible for transmitting the result for further analysis. Once the local processing is complete, the result is transmitted to a satellite communication system to send it from space station to the earth station. As the result arrives at the earth station, it is transmitted to a router or gateway to send it to later the internet. After that, internet would deliver the data to the cloud for further analysis by other computing systems.

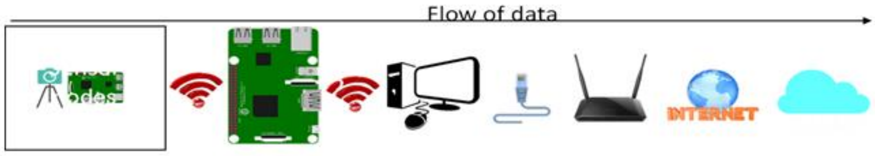

Figure 5. *Data flow in proposed WSN communication system.*

Raw data can be transmitted without compression during cloud computing, resulting in a compression ratio of 0%, while this may require more bandwidth and power than compressed data. Transmitting large amounts of uncompressed data may put a strain on the network, and it's important to consider the bandwidth and power requirements of both the local and internet networks involved in cloud computing. Transmitting data wirelessly in a WSN consumes a significant amount of energy. By reducing the size of transmitted data, the energy required for transmission is also reduced, which can help to extend the battery life of the sensor nodes. Also, reducing the size of data transmitted over a WSN, more data from more sensors can be sent within a given time, which can help to increase the overall capacity of the network. This can be particularly useful in applications where real-time data is required, such as in environmental monitoring or industrial automation. WSNs can be deployed in harsh and remote environments where it may be difficult or costly to maintain or replace sensor nodes. By reducing the size of data, the likelihood of transmission errors or packet loss is reduced, which can help to improve the overall reliability of the network. Reducing the size of data can help to reduce the cost of WSNs. This is because smaller data require less memory and processing power, which can translate to lower hardware costs. Additionally, smaller data size requires less bandwidth, which can result in lower communication costs.

There is always a tradeoff between the time taken to produce the results and the power consumed to execute the algorithm. If the algorithm requires a significant amount of time and energy to execute, then it may not be practical to perform it on the edge device. Instead, it may be better to offload the task to a more powerful centralized computing system. However, this decision should not be made solely based on the time and energy consumption of the algorithm. Other factors such as the amount of data being processed, the size of the model, and the bandwidth available for communication with the centralized system should also be taken into account. In some cases, it may be more efficient to perform the computation on the edge device, even if it consumes more time and energy, due to the reduced latency and bandwidth requirements. Therefore, when deciding whether to perform a computation on the edge or a centralized system, a comprehensive analysis of all relevant factors should be performed to make an informed decision.

Conclusion

Edge computing is a promising solution to address the high energy consumption and communication overheads associated with image processing and classification in wireless sensor networks (WSNs). Our evaluation of different ML algorithms demonstrates that edge computing significantly improves efficiency and reliability, particularly in extreme environments with limited resources. However, it's crucial to ensure that preprocessing time at the edge plus link delay is shorter than sending raw data, and that energy consumption and bandwidth usage are not exceeded compared to cloud computing. These factors should be considered when deciding whether to use edge computing in WSNs.

Acknowledgement

This research is self-funded.

Conflict of interest

The authors declare that there is no conflict of interest involve in this research study.

REFERENCES

[1] Breiman, L. (2001): Random forests. – Machine Learning 45: 5-32.
[2] Gewali, U.B., Monteiro, S.T., Saber, E. (2018): Machine learning based hyperspectral image analysis: a survey. – arXiv preprint arXiv:1802.08701 46p.
[3] Govender, M., Chetty, K., Bulcock, H. (2007): A review of hyperspectral remote sensing and its application in vegetation and water resource studies. – Water Sa 33(2): 145-151.
[4] Haq, M.A., Rehman, Z., Ahmed, A., Khan, M.A.R. (2022): Machine Learning-based Classification of Hyperspectral Imagery. – International Journal of Computer Science & Network Security 22(4): 193-202.

[5] Huang, C., Wylie, B., Yang, L., Homer, C., Zylstra, G. (2002): Derivation of a tasselled cap transformation based on Landsat 7 at-satellite reflectance. – International Journal of Remote Sensing 23(8): 1741-1748.

[6] Mohammed, H.R., Abduljabbar, J.M. (2022): EDGE COMPUTING ENABLED WIRELESS SENSOR NETWORK REQUIREMENT: A SURVEY. – Quantum Journal of Engineering, Science and Technology 3(3): 1-13.

[7] Molnar, C. (2022): Interpretable Machine Learning: A Guide For Making Black Box Models Explainable. – Lulu 328p.

[8] Müller, A.C., Guido, S. (2016): Introduction to machine learning with Python: a guide for data scientists. – O'Reilly Media, Inc. 398p.

[9] Nguyen, T.A., Huynh, N.H., Le, T.N. (2022): Edge Computing for Wireless Sensor Network. In International Conference on Intelligence of Things. – Cham: Springer International Publishing 11p.

[10] Rashvand, H.F., Abedi, A., Alcaraz-Calero, J.M., Mitchell, P.D., Mukhopadhyay, S.C. (2014): Wireless sensor systems for space and extreme environments: A review. – IEEE Sensors Journal 14(11): 3955-3970.

[11] Uno, Y., Prasher, S.O., Lacroix, R., Goel, P.K., Karimi, Y., Viau, A., Patel, R.M. (2005): Artificial neural networks to predict corn yield from Compact Airborne Spectrographic Imager data. – Computers and Electronics in Agriculture 47(2): 149-161.