Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 01, PP. 207-217, 2026
DOI: https://doi.org/10.54216/JISIoT.180115

Satellite Imaging Based Risk Management in Cloud IoT Network Using Machine Learning Techniques

Jyotsnarani Tripathy[1,*], T. Krishna Murthy[2], S. Manjula[3], Sukanya Ledalla[4], Alla Rajendra[5], P. Lakshmi Harika[6], K Boopathy[7]

[1]Assistant Professor, Department of AIML & IoT, VNR Vignana Jyothi Institution of Engineering and Technology, Hyderabad, Telangana, India

[2]Assistant Professor, Department of Computer Science and Engineering, Mallareddy University, Hyderabad, India

[3]Assistant Professor (Sr Gr), Department of Computer Science and Business Systems, Nehru Institute of Engineering and Technology, Coimbatore, India

[4]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Bowrampet, Hyderabad-500043, Telangana, India

[5]Assistant Professor, Department of Computer Science and Engineering, Aditya University, Surampalem, India

[6]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana, India

[7]Department of Electrical and Electronics Engineering, Aarupadai veedu Institute of Technology,Vinayaka Missions Research Foundation(DU), Chennai Campus, Paiyanoor:603104, India

Emails: jtjyotsna@gmail.com; thaticherla.krishnamurthy@mallareddyuniversity.ac.in; manjulapmu@gmail.com; ledalla.sukanya@gmail.com; rajendracivil127@gmail.com; pothuri.harika99@gmail.com; boopathyk@avit.ac.in

Abstract

The consistent improvement of remote sensing (RS) technology has resulted in an easy access to a large volume of satellite imagery. There is a need for effective and scalable solutions for widening the application of RS in different fields and making it work efficiently in practical situations. This research propose novel technique in satellite image gathering and cloud IoT network risk management using machine-learning model. Here the cloud IoT network has been used in satellite image collection and this network security analysis has been carried out using secure trust based cryptographic blockchain model. Then this collected image has been classified using convolutional bayes fuzzy markov perceptron basis function model. Experimental analysis has been carried out in terms of accuracy, QoS, recall, latency, scalability. Proposed model attained accuracy of 97%, QoS of 94%, LATENCY of 96%, Scalability of 95%, RECALL of 93%. These results assist decision-makers, planners, and scientists studying remote sensing select an appropriate image classification system for tracking a dynamic, fragmented, and varied landscape.

Received: February 19, 2025 Revised: June 02, 2025 Accepted: July 03, 2025

Keywords: Cloud IoT network; Risk management; Machine learning model; Satellite image; Cryptographic blockchain

1. Introduction

It is rather interesting to note that remote sensing technology has continued to gain different levels of appreciation within the last decade, more so, from 2008, when NASA's Landsat satellites disposed of data policy. Here, data preside

of Optic-Looking and Radar Sensing of the surface of the Earth together with the Troposphere' Chemistries became free of charge via initiation of Copernicus ESA offered satellite data significantly enhanced free satellite data availability. The deployment of these data is expected to be along with modern techniques of data analytics to further stimulation of the economy and the inventive expansion. Just like how we have all come to abuse GPS on our regular phones, from the perspective of the objectives of the Copernicus initiative, remote sensing should be in the hands of the masses. The burden of satellite data acquisition has been resourcefully lightened due to several elements; among them the prevalence of sophisticated open access software, increased connectivity and data usage, high-end computing facilities, along with ready access of quality data to users [1]. There are still issues that need to be addressed within numerous sectors, regardless the recent improvements in the development of applications geared towards data pre-processing and analytics. Corporations, governmental organizations and any other institutions that require mapping of the earth through satellite images may operate imaging satellites. High-resolution images are obtained using Remote Sensing (RS) techniques. Remote Sensing in general refers to the collection and processing of information concerning an object, location or event without any physical interaction with it. For many applications, including land use classification, RS data is regarded as an extremely valuable information source, particularly when combined with artificial intelligence technologies [2]. The need for higher-resolution satellite photos is driving up their size, and in a similar vein, the expanding volume of RS data has made it possible to analyze a wide range of intricate scientific subjects. However, using modern DL-based approaches with intricate architectures and high computing demands is usually necessary to produce sufficient RS images. In order to do this, a lot of academics utilize DL approaches that allow them to extract valuable information and insights using cloud computing platforms [3]. Data workflows may, however, raise privacy issues in these situations due to public nature of the data processing as well as management tools. Data leakage may happen in this situation, and data privacy cannot be guaranteed. Cloud computing still has a number of advantages, nevertheless, including cost savings, flexibility, scalability, and masking design complexity. Therefore, the best option is one that can balance the drawbacks to maximize the advantages. However, there are also a number of privacy issues that need to be resolved, especially when public DL techniques are used to transmit or store satellite photos [4]. PPDL approaches can be utilized to gain insights from public data while simultaneously limiting data leaks and protecting sensitive material from criminal use and unauthorized access. DL is frequently used to create prediction models for applications involving text and speech recognition as well as picture processing. Particularly when trained on sizable data sets, these models are more accurate. Prediction is the process of analyzing the data that is now available and then applying that knowledge to generate previously unobtainable information. However, these data frequently also include sensitive data that needs to be preserved. Thus, protecting the privacy of such data when it is transferred to public cloud for processing as well as analysis presents a significant difficulty. The majority of the time, staff PCs are not capable of processing large satellite images. Therefore, there is a rising necessity to leverage public cloud servers for large data analysis to extract valuable knowledge as well as insights from such RS data. Privacy of data gathered as well as processed by cloud service providers during DL training is also becoming a more difficult topic as a result of this increased reliance on cloud services [5].

2. Literature review

With the emergence of such annotated imagery datasets, both detection and classification activities have increased. Most of the applications of deep learning to remotely sensed data have been concentrated on building detection or land cover classification [6]. This method provides improved forecasting results compared to weather research and forecasting (WRF) method extending up to 12 hours. In the volume and update level research using the real-world dataset, work [7] introduced a DL based weather forecasting approach. It is established that more data enhances accuracy of method, as it is not dependent on the recency of the information used. The phrase Self-organizing Map (SOM)-based Latent Dirichlet Allocation (LDA) was coined by [8], who implemented SOM to cut back on the amount of data applied prior to diminishing the data used for climate predictions. This technique, however, shows 7 - 23% improvement on weather and crop prediction accuracy over the existing methods. In addition, it is observed by [9], that the enhancement of weather prediction systems can be based on the application of ML as well as DL for artificial intelligence. A DL-based LSTM model has been developed by [10] and tested against traditional ML techniques such as Support Vector Regression (SVR) and Random Forest (RF). Produced LSTM method was found to outperform significantly the evaluated benchmarks. In order to estimate the wind power production, the author [11] applies ensemble ML based method similar to – numerical weather prediction (NWP) wind generation as well as weather station observation data using artificial intelligence. The work [12] before training augmenting data of national scale precipitation forecasts by deep learning approaches, I provide evidences of how to apply the UNET with residual learning to global data assistance precipitation forecast models. This study has demonstrated that a UNET based on residual learning can understand the physical relationships with precipitation; consequently, allowing the use of physical limitations in dynamical operational models to improve rainfall prediction. These results are important

because they enabled the development of simultaneous hybrid forecasting models. To address this issue, [13] has proposed a formal rain prediction for agriculture in India based on data and ML methods, particularly linear regression method. The authors [14] introduced two RF-based techniques, Feature Level Fusion Random Forest (FLFRF) as well as Decision Level Fusion Random Forest (DLFRF), to integrate IR, spectral, textural characteristics images for cloud classification to improve accuracy of cloud classification on distant sight photos. A color histogram that displays the percentage of clouds in each pixel in the image is calculated using cloud pixel density. Clear skies or no clouds are represented by pixel 0, while opaque or dense clouds are represented by pixel 255 [15].

3. Proposed model

Assuming that there are I satellite photos in the original dataset, there are N images, and I1 and I2 are the training images. Image dataset I is processed by reading and loading the photographs from every directory, scaling them to fit deep convolutional network's input specifications (Fig. 1), labeling every image instance to indicate whether or not it is a cyclone. Since training images I1; I2; : : : ; IN are RGB (color images), they are now represented by an array in which the intensity of each image pixel is stored in an index. Specifically, three 2D arrays are defined: R-Red, B-Blue, and G-Green. In their separate indexes, intensity value of each color is recorded. When the data is ready, it needs to go through a suggested model for classifying satellite images, which is depicted in Figure 1.

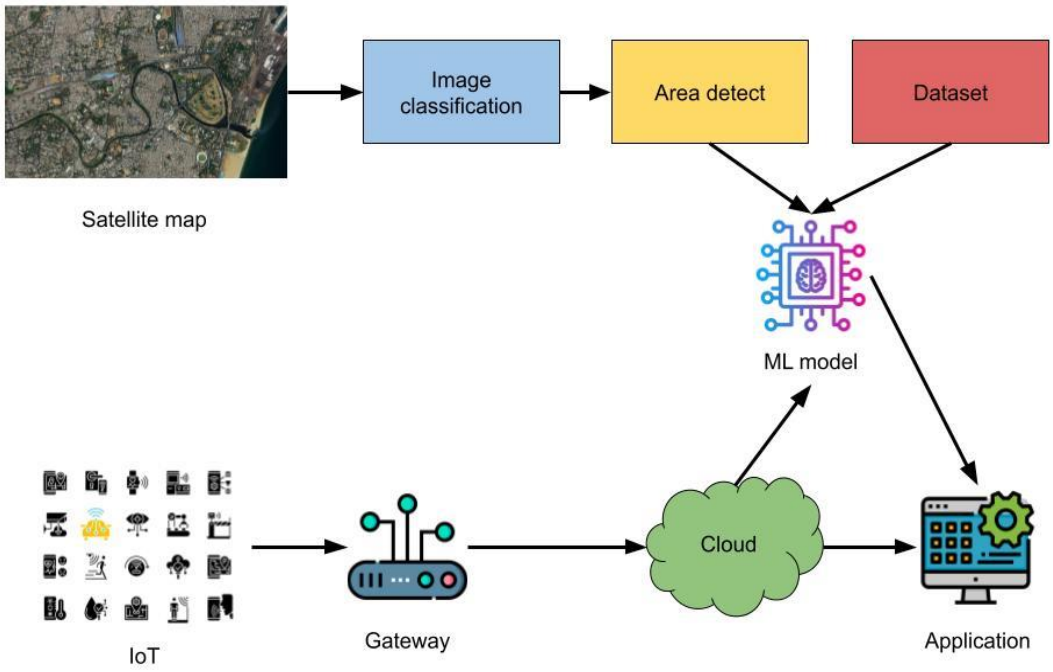


Figure 1. Proposed cloud IoT model in satellite image analysis architecture

Digital or satellite images can be used to collect cloud images. Digital images can be taken from the ground up using a computer camera or the internet. On the other hand, satellite imagery is either collected by the meteorology division or routinely collected from certain meteorology locations. Because satellite images have a clearer cloud structure than digital images, they can be used to accurately forecast precipitation. Different categorization techniques are applied to the complete disk picture of the Indian National Satellite-3D Repeat (INSAT-3DR) Thermal Infrared (TIR).

4. Cloud IoT model network security analysis using secure trust based cryptographic blockchain (STCB) technique

Rivest-Shamir-Adleman (RSA) is first accomplishment of public key (PK) cryptosystem. Asymmetric key encryption, or public-key encryption, is a kind of technique that requires two distinct keys: a public key and private key that is specific to a person. A pixel-by-pixel confidence measure ranging from 0 to 1 for the presence of a cloud is provided by the output classification. A mapping function, f, that describes a fully convolutional network is provided by eqn (1)

$$\mathbf{Y} = f(\mathbf{X}; \boldsymbol{\theta}) \quad (1)4$$

where θ are mapping function's parameters. We can group the parameters in a way that allows us to create function f from composition of 2 functions, f1 and f2 by eqn (2)

$$\mathbf{Y} = f_2(f_1(\mathbf{X}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2) \quad (2)$$

As a result, the mapping function is composed of layers, or smaller sub-functions. The finished model is made up of numerous layers, making it a deep model. The most crucial are convolutional layers, which convolve their input with a predetermined number of kernels to create feature maps by eqn (3)

$$(\mathbf{I} * \mathbf{K}_d)(i, j, d) = \sum_m \sum_n \sum_l \mathbf{I}(i - m, j - n, l)\mathbf{K}_d(m, n, l) \quad (3)$$

where Kd is d'th m by n by l kernel and I is input. Depth of output feature maps determines number of kernels, d, which is manually selected during network architecture design. It is mentioned that kernel always operates on whole depth of input feature map, even when it is only employed on a limited spatial region. After a linear transformation, the kernels produce a non-linearity known as an activation function. Widely utilized rectified linear unit (ReLU) function is one example of this; it zeroizes any negative values by eqn (4)

$$\mathbf{K}_d(\mathbf{X}) = \max(0, \mathbf{W}_d\mathbf{X} + b_d) \quad (4)$$

where the trainable parameters in θ are the weights, Wd, and bias, bd. To process a user request, the suggested structure consists of the following steps:

1. Registration Phase: During this stage, a reliable registration authority securely registers the entire data center node offline. Furthermore, a zero knowledge proof protocol is used to reg ister the light node, also known as the sensor node. Two parties are authenticated using this protocol without disclosing any private information or identities. One side takes on the role of the challenger in this strategy, while the other side takes on the role of prover.
2. Verification Phase: P is started for verification from light node end during this phase. P and I combination is used to find all possible values that are correct based on the information received from inputs. To keep the difficulty level constant, new variable is calculated at light node end by adding the value (z). The huge prime value as well as matching generator are utilized to evaluate value z. In this case, Z turns into the verifier and it into the prover.
3. Validation and Block Creation Phase: Process of joining blockchain occurs if the data is registered. Stages involved in creating and validating blocks are as follows:
4. The first stage involves creating a key value pair (P Bkj, P Rkj), where P Rkj is recognized as the jth light node's private key and P Bkj is identified as its public key.
5. Work of registration is started.
6. Creates a signature as well as forward to respective nodes for its validation.
7. Signature is verified by the access control policies. The client sends a request to join network using credential P Bkj after the signature successfully matches.
8. To verify the user's location, security smart contracts send a validation request to peer nodes (N-p).
9. The blockchain's smart contracts record a timestamp based on latitude and longitude, which peer nodes (N-p) use to verify nodes' locations.
10. Following verification, the associated node receives a True/False acknowledgment.
11. A new block (Bj) is created as well as added to blockchain network using credential P B-k-j in order to achieve True status.
12. Block update and data generation: This stage explains how (LNLN) creates data. Transaction (Tj) is the designation for the created data.

5. Convolutional bays fuzzy markov perceptron basis function (CBFMPBF) model:

A convolutional layer, an activation layer, max-pooling layer make up CNN. The max-pooling layer, which performs a down-sampling operation, divides method 7 convolutional layers into 3 segments. This procedure decreases vectors of NN it is working with and extracts its characteristics. As previously mentioned, an activation layer that uses ReLU as well as SoftMax performs after a convolutional layer. Convolution of features is the transformation that ReLU does, suggesting that the features are bound together or entangled. It gives the model nonlinearity as well. Therefore, the following is a definition of the ReLU function by eqn (5)

$$F(t) = \max(0, t) \quad (5)$$

For all values where t > 0, yields t; otherwise, it returns 0. For multi-perception level, model flattens feature matrix in a vector and feeds it into a feed-forward neural network after converting image into a downsampled feature matrix. Perceptron uses an activation function called SoftMax to classify the test image by eqn (6)

$$f_j(z) = \frac{\theta^{z_j}}{\sum_{k=1}^{M} \theta^{z_k}} \quad (6)$$

According to the approach, process of learning weights for feature maps begins when inputs are sent to a deep convolutional network as training data. Backpropagation aids the network in learning all feature maps or hyperparameters, including the neural network's number of nodes per layer. taking into account training model and entering "TS" into a function S=T, where s D jT j is total number of testing images.

By using actual findings, the Bayes theorem is utilized to estimate posterior probability distributions from prior probability distributions that are only weakly informative. The strategy is stated as follows (7)

$$P(Y \mid S) = \frac{P(Y)P(S|Y)}{P(S)} \quad (7)$$

Combined probability of many conditions for several DAG factors can be written as eqn (8)

$$P\{T \mid Y_1, \dots, Y_j, \dots, Y_m\} = \prod_{j=1}^{m} P(T \mid Y_j) \quad (8)$$

where Yj is jth of m factors influencing T. The hazard probability of each component is determined using its spatial properties in a BM. X, X = {Xj,1, Xj,2,..., Xj,i−1, Xj,i,..., Xj,n} provides the spatial features of Yj, where Xj,i is ith of nth features of jth factor.

$$P\{X_{j,i} \mid S, t = k\} = P(X_{j,i}, t = k) \frac{P\{S|X_{j,i}, t=k\}}{P(S)} \quad (9)$$

where prior probability is P (Xj,i, t = k); conditional probability is P{S|Xj,i, t = k}, which is determined by dividing subsidence grid by feature grid; marginal probability is P (S), which is total of probabilities of every Xj,I by eqn (10)

$$P(S) = \sum_{i=1}^{n} P(X_{j,i}, t = k) P\{S \mid X_{j,i}, t = k\} \quad (10)$$

In order to update the posterior probability, this phase is repeated whenever fresh subsidence events are noticed. Both X1 and X2 are parent nodes of X3 and are root nodes. X1 and X2 are the parents of X3. There are two states for every node: "state 0" and "state 1." By eqn (11)

$$P(X_1, X_2, \dots, X_n) = \prod_{1}^{n} P(X_i \mid \pi_i)(i = 1,2, \dots, n) \quad (11)$$

$\pi_i$is set of parents of node Xi. Marginal probability of Xi is given by eqn (12)

$$P\{X_i\} = \sum_{\text{except } X_i} P\{X_1, X_2, \dots, X_n\} \quad (12)$$

BN is an inferential tool that determines the likelihood of an occurrence based on further evidence. The process for calculating posterior probability of evidence e is demonstrated by following formula (13)

$$P(U \mid e) = \frac{P(e|U)P(U)}{P(e)} = \frac{P(e|U)P(U)}{\sum_{ll} P(e|U)P(U)}, U = \{X_1, X_2, \dots, X_n\} \quad (13)$$

P(U): the event U's prior probability, P(e): the evidence's predetermined posterior probability, P(U|e): evidence e's posterior probability P(e|U): event U's evidentiary likelihood, and. ∑ U P(eU)The combined probability distribution of e is P(U). A fuzzy version of BN, known as a fuzzy Bayesian Network (FBN), exists in addition to the conventional BN modeling. It reduces information loss and presents ambiguous information. Experts use a fuzzy linguistic scale to assess primary FMs and auxiliary FMs in the context of the current investigation. Each FM's final occurrence

probability are acquired. Following this phase, crisp failure probabilities are obtained and BN modeling is developed. Statistical method that includes pair (π,A) as well as emission process is called an HMM. The continuous emissions technique is what we choose. The parametric technique is the most often used method for modeling continuous emissions by eqn (14)

$$p(\mathbf{x} \mid S_i) = \sum_{m=1}^{M} c_{im} N(\mathbf{x} \mid \mu_{im}, \Sigma_{\text{im}}),\ i = 1, \ldots, Q$$

Where

$\sum_{m=1}^{M} c_{im} = 1,\ i = 1, \ldots, Q.$ (14)

To streamline the estimation process, we have divided the parameter set into two major classes- transition parameters (π, A) as well as emission parameters (μ, Σ, C). Because of this split, we can also independently train two smaller parameter sets which overall decreases the training parameters size significantly. As the activity labels related to the samples in the training set are assumed available, transition parameters can be estimated by employing rather trivial counting of how frequently some events occur. In our classification approach, we are able to formulate a criterion that allows automatic rejection of producing incorrect feature vectors. The application of a threshold-based detector to provide estimates of class-conditional probabilities p(x | wi) removes the necessity of explicitly augmenting the model for unknown data with extra states or mixture components. This also allows for the rejection of feature vectors, for which the classification is deemed too uncertain and, unattractive, even too risky. In the learning step, the classifier's confusion matrices as well as classified outputs generated by classifier is applied to this stage for method training. At classification phase, sequence of classified data is reclassified utilizing trained system. Misclassified examples are examined as well as appropriate ones adjusted.

Even if layers' weights can be modified when the process of training is underway, the amount of duration involved coupled with the techniques used in closing the wide scale value differences is quite unbearable for the network. There is a limited amount of labelled data; due to this fact, overtraining is most prone to take place where method learns the sounds and characteristics present in training data. In this sense, it is recommended to apply l2 normalization before summation of feature maps so that all feature maps' value scales are equal. L2 Normalization is applied to every pixel of feature maps. The d-channel vector l2-norm for a layer that with d-channel feature maps with dimensions (w, h) is illustrated in equation (15).

$\|x\|2 = (\sum_{l=1}^{d} \| x_l \|^2)\ 1/2$ (15)

Equation (16) normalizes the vector x of a layer per d-dimension pixels.

$x' = \frac{x}{\|x\|2}$ (16)

where vector of normalized pixels in d-dimension is denoted by $\hat{x}$. To expedite training process, scale values of the layers are always rescaled with a factor μ for every channel i by eqn (17)

$yl = \mu x_l$ (17)

The scaling factor μ can be updated thanks to the chain rule and backpropagation. Proposed method uses adjacent squares of recommendation as context information. Bounding boxes are projected to many layers while keeping same quantitative relationship in order to obtain contextual characteristics. We maintain $w_{\text{context}} = \lambda \times w_{\text{proposal}}$ and $h_{\text{context}} = \lambda \times h_{\text{proposal}}$, where w and $\mathbb{\Pi}$ represent width and height of bounding box.

The gradient descent method was used in the learning phase. The primate base, to which the majority of measurements are made and usually referred to, supplies the input to 60 of the 120 input neurons. One of the three randomly chosen planes is given to the remaining 60 neurons in each other epoch. Of the 60 inputs, 30 are in shorter period region while remaining 30 are in the longer period. All of the upcoming epochs' coordinates are chosen at random from a single, constrained space period region of the projected grating. While y coordinates that correspond to long spatial period area are arbitrary, x coordinates are identical to those of short spatial period area. Network's intended output was distance between reference plane and the randomly selected aircraft. Inputs as well as outputs of network were pre- and post-processed to range of 1 to +1 in order to enhance network training. Height and depth of unknown object were found on the network in the manner described below. The set of input neurons was still only receiving information from the original reference plane.

6. Results and discussion

Simulation setup- Oracle Virtual Box (Oracle, 2018) running on a Windows 10 machine was utilized to host 3 virtual machines to deploy broker software. Every virtual system had a 15GB hard drive, 1 CPU, and 8GB of RAM. Three open-source broker software packages were utilized to install version 3.1, which was utilized to assess various monitoring scenarios.

Dataset description- Modified Soil Adjusted Vegetation Index, Normalized Difference Vegetation Index, Green Normalized Difference Vegetation Index, and Soil Adjusted Vegetation Index 2 were the vegetation indices utilized to process Sentinel-2 imagery in order to examine crop health. Because they inherit their land from their parents across many years, most Lebanese farmers still use traditional irrigation techniques that are based on the customs of earlier generations. However, because of the region's changing climate, these techniques are no longer as effective as they previously were. According to preliminary results, Sentinel-2 data can give farmers precise and timely field management data. Giving Lebanese farmers access to crop health data on their phones and allowing them to assess its efficacy could help them improve their farming practices, even if the bulk of them still employ traditional agricultural methods.

GNSS Dataset consists of three parts: Parts I, II, and III, II, III. Part IV (Precallessed data and Supplementary Raw data, March 15–24, 2024). Part V: Supplementary Raw Data and Precallessed Data, March 24-30, 2024. Elsevier's paper "GNSS Interference and Spoofing Dataset" offers comprehensive details on Parts I, II, and III. The same gathering method is used in Parts IV and V, but additional spoofing data is included. Data was recorded by a GNSS receiver located on fifth level of Science Hall at Yunnan University. HackRF One uses spoofing signals and commercial jammers use suppression jamming to assault receiver. Research communities that concentrate on GNSS security, monitoring, anti-spoofing and anti-jamming methods will find value in the provided datasets. For every ICME, dataset contains following parameters: disturbance time periods as obtained from Richardson and Cane ICME catalogue https://izw1.caltech.edu/ACE/ASC/DATA/level3/icmetable2.htm; minimum Bz component and corresponding time stamp; orbit decay for 490 km; orbit decay start and end times; time difference between high Bz and orbit decay end; geomagnetic indices; solar indices.

ESA Service Satellite Orbit DecAy (SODA)- Information includes findings from examination of 116 interplanetary coronal mass ejections (ICMEs), which were utilized to create SODA forecasting service that is accessible through the Expert Service Center Ionosphere (I.161) of ESA Space Service Network.

Table 1: comparison for various satellite image dataset

Dataset	**Techniques**	**Accuracy**	**QoS**	**Latency**	**Scalability**	**RECALL**
SENTINEL-2	**FCNN**	79	75	74	77	73
	ANFIS	83	79	80	81	76
	STCB_CBFMPBF	87	85	89	88	82
GNSS Dataset	**FCNN**	75	78	76	73	81
	ANFIS	79	84	83	80	86
	STCB_CBFMPBF	90	87	92	89	92
SODA Dataset	**FCNN**	81	79	82	84	85
	ANFIS	88	86	84	87	89
	STCB_CBFMPBF	97	94	96	95	93

Table-1 shows analysis for climate change dataset. Satellite image dataset analysed are SENTINEL-2, GNSS Dataset and SODA dataset in terms of accuracy, QoS, Latency, Scalability, and Recall.

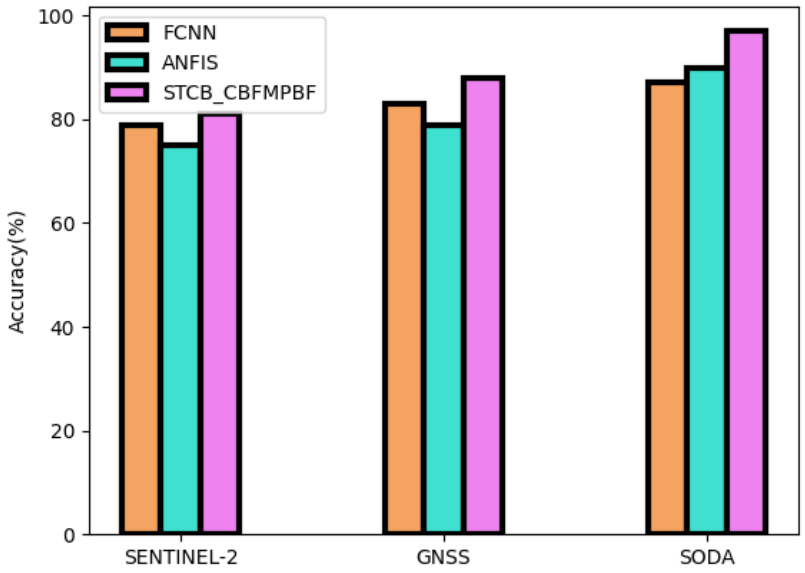


Figure 2. Comparison of Accuracy

Analysis for accuracy is shown in Figure 2. Proposed technique 87% accuracy, 79% existing FCNN, 83% ANFIS for SENTINEL-2 dataset; for GNSS Dataset, proposed technique 90% accuracy, 75% existing FCNN, and 79% ANFIS; for SODA, proposed technique 97% accuracy, 81% existing FCNN, 88% ANFIS.

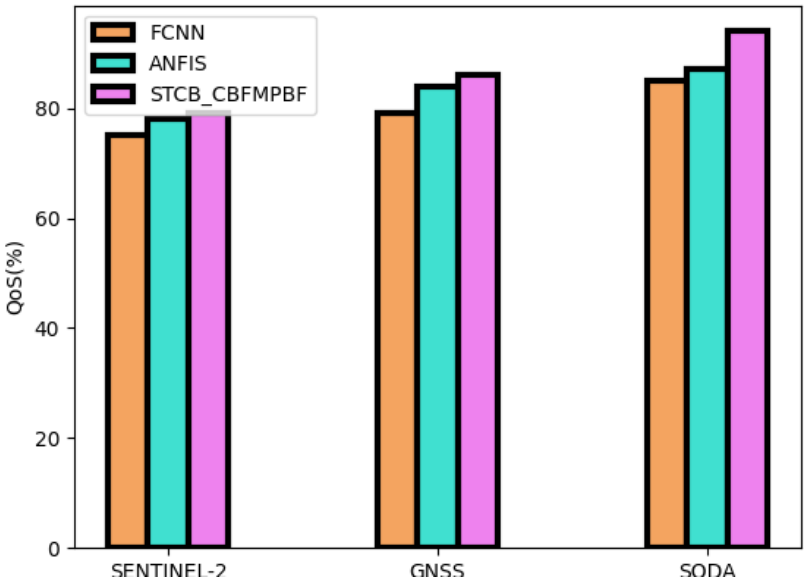


Figure 3. Comparison of QoS

Figure 3 shows analysis in QoS. Proposed technique QoS of 85%, existing FCNN 75%, ANFIS 79% for SENTINEL-2 dataset; for GNSS Dataset proposed technique QoS of 87%, existing FCNN 78%, ANFIS 84%; proposed technique QoS of 94%, existing FCNN 79%, ANFIS 86% for SODA .

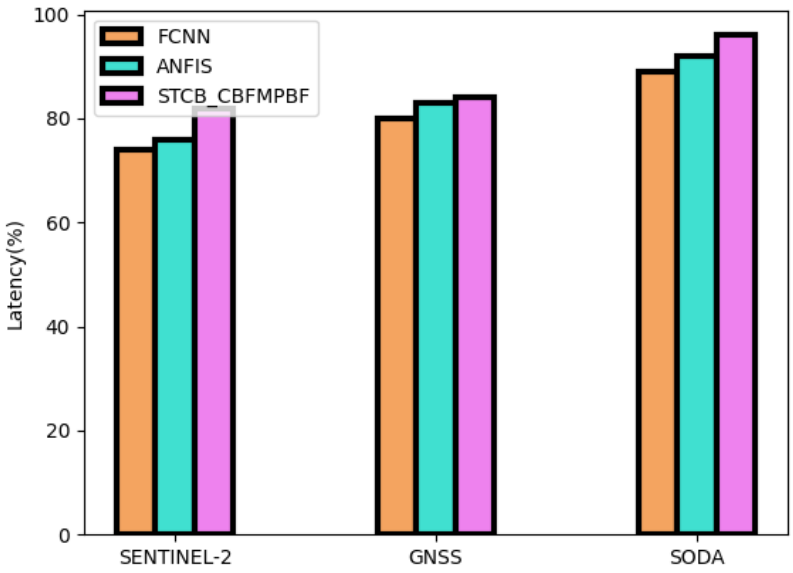
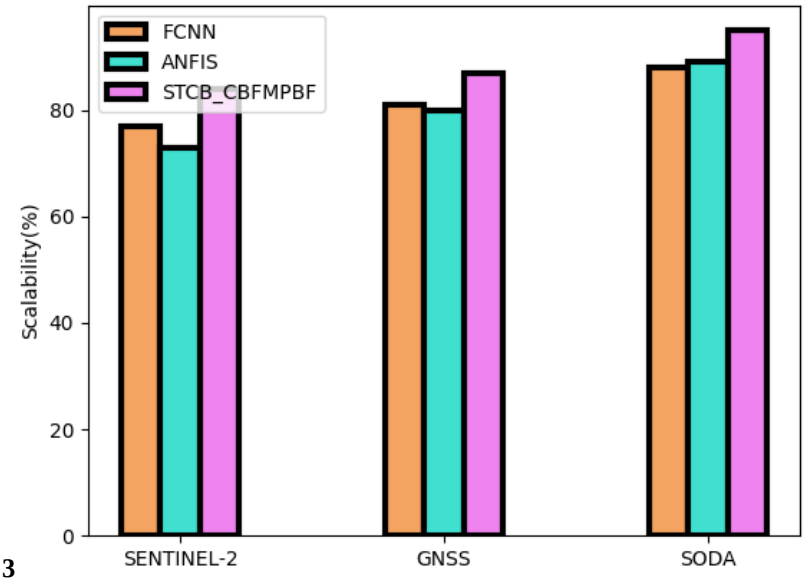


Figure 4. Comparison of LATENCY

Analysis in LATENCY is shown in Figure 4. In SENTINEL-2 dataset, proposed technique LATENCY 89%, existing FCNN 74%, ANFIS 80%; in GNSS dataset, proposed technique LATENCY of 92%, existing FCNN 76%, ANFIS 83%; in SODA dataset, proposed technique LATENCY 96%, existing FCNN 82%, ANFIS 84%.



3

Figure 5. Comparison of Scalability

Scalability analysis is displayed in Figure 5. Proposed technique 88% scalability, 71% existing FCNN, 81% ANFIS for SENTINEL-2 dataset; for GNSS Dataset, proposed technique 89% scalability, 73% existing FCNN, 80% ANFIS; for SODA Dataset, proposed technique 95% scalability, 84% existing FCNN, 87% ANFIS.

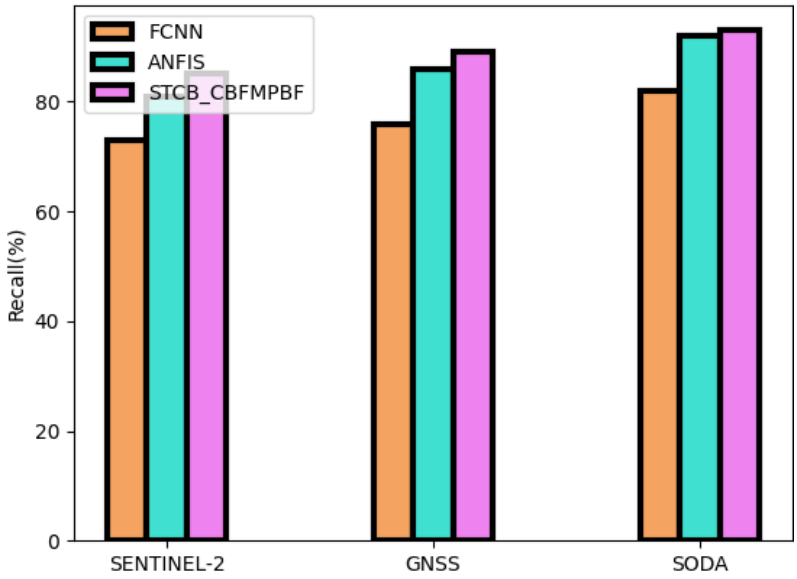


Figure 6. Comparison of RECALL

Figure 6 shows analysis in RECALL. Here proposed technique RECALL of 82%, FCNN 73%, ANFIS 76% for SENTINEL-2 dataset; for GNSS Dataset proposed technique RECALL of 92%, FCNN 81%, ANFIS 86%; proposed technique RECALL of 93%, FCNN 85%, ANFIS 89% for SODA Dataset.

In this instance, loss and accuracy are perfect for every epoch. The training data shows signs of underfitting from the very first epoch. However, as epoch progresses, training data starts to fit method. Accuracy in Ep-1 is nearly 93% with a loss of about 8%. This means that out of 494 images in first batch of the dataset 93% were trained correctly and 7% were not. In Ep-2, the loss value is about 2.0 and the accuracy is approximately 98%. The batch size indicates that a considerable number of images have been trained within the model. The overfitting of the model starts when it is seen to learn effectively, which is corresponding to Ep-8. This shows that there is a higher training accuracy than validation accuracy. Ep-9 model seems to fit appropriately as well as accuracy falls to 1%. In the beginning of the epochs, there is a clear gap between the two lines. However, by training DNN, accuracy of the network increases while the gap between the two lines narrows. The argument for overfitting proposed method on both training and validating sets is that they both tend to meet at ninth epoch. We can now evaluate performance of method in terms of its accuracy. Equation (3) determines the confusion matrix formula. Ratio of sum of true positives and false positives to total number of occurrences can be used to describe it.

7. Conclusion

Employing a machine-learning framework, the research proposes an interesting approach to acquiring satellite images and assessing the risks associated with the cloud-IoT ecosystem. Considering trust in the system as its foundation, this exploration into network security was carried out in a cloud-based IoT network, which was used for storing and managing satellite imagery. In the further processing of the received image a convolutional Bayes fuzzy Markov perceptron basis function model was applied. How much precision is increased will depend also on the machine and the model used. Even better precision may be obtained by increasing the image resolution and using an efficient computer system. In this case, training suffers, as the volume of the training images is very large. The suggested architecture was drawn out because of numerous tests in order to achieve maximal overfitting resistance and consequently enhance generalization of the network. The architecture of method also addressed issue of similarity between classes to acquire finer as well as coarser features that are specific to each c by way of modification of the architecture as needed.

References

[1] K. K. Jena, S. K. Bhoi, S. R. Nayak, R. Panigrahi, and A. K. Bhoi, "Deep convolutional network based machine intelligence model for satellite cloud image classification," *Big Data Mining and Analytics*, vol. 6, no. 1, pp. 32–43, 2022.

[2] A. Rahman *et al.*, "Performance of different machine learning algorithms on satellite image classification in rural and urban setup," *Remote Sensing Applications: Society and Environment*, vol. 20, p. 100410, 2020.

[3] T. I. Leong, Y. M. Abbas, M. A. C. Purio, and H. A. Elmegharbel, "Image classification unit: A U-Net convolutional neural network for on-orbit cloud detection aboard CubeSats," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Brussels, Belgium, 2021, pp. 2807–2810.

[4] H. A. F. S. A. Ouchra, A. Belangour, and A. L. L. A. E. Erraissi, "Machine learning algorithms for satellite image classification using Google Earth Engine and Landsat satellite data: Morocco case study," *IEEE Access*, 2023.

[5] R. Gupta and S. J. Nanda, "Cloud detection in satellite images with classical and deep neural network approach: A review," *Multimedia Tools and Applications*, vol. 81, no. 22, pp. 31847–31880, 2022.

[6] A. Nespoli *et al.*, "Machine Learning techniques for solar irradiation nowcasting: Cloud type classification forecast through satellite data and imagery," *Applied Energy*, vol. 305, p. 117834, 2022.

[7] S. Saini, R. Dahiya, S. Ratna, and A. Singh, "Satellite Image Classification Using Deep Learning," in *Proc. Int. Conf. Disruptive Technol. (ICDT)*, Greater Noida, India, 2024, pp. 1247–1253.

[8] J. Byun, C. Jun, J. Kim, J. Cha, and R. Narimani, "Deep learning-based rainfall prediction using cloud image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–11, 2023.

[9] R. Manoharan, "Improving Security and Performance in Chaotic Optical Communication via Real-Time Pilot Signal Processing Techniques," *IETE J. Res.*, pp. 1–9, 2025.

[10] M. Rajesh, S. Ramachandran, K. Vengatesan, S. S. Dhanabalan, and S. K. Nataraj, "Federated Learning for Personalized Recommendation in Securing Power Traces in Smart Grid Systems," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 88–95, Feb. 2024.

[11] S. Kopeć, G. Duniec, B. Bochenek, and M. Figurski, "Artificial neural networks in automatic image classifications of cloud from ground-based observations using deep learning models," *Quart. J. Roy. Meteorol. Soc.*, 2024.

[12] H. Ouchra, A. Belangour, and A. Erraissi, "Exploring Google Earth Engine Platform for Satellite Image Classification Using Machine Learning Algorithms," in *Proc. Int. Conf. Smart City Appl.*, 2023, pp. 271–280.

[13] B. Guo, F. Zhang, W. Li, and Z. Zhao, "Cloud classification by machine learning for geostationary radiation imager," *IEEE Trans. Geosci. Remote Sens.*, 2024.

[14] O. N. Ukamaka, R. O. Enihe, and A. M. Jubril, "Cloud Detection in Satellite Imagery Using Deep Convolutional Neural Networks (CNN)," in *Proc. Int. Conf. Sci., Eng. Bus. Driving Sustain. Develop. Goals (SEB4SDG)*, Abuja, Nigeria, 2024, pp. 1–6.

[15] A. S. Minkin and O. V. Nikolaeva, "Cloud Recognition in Hyperspectral Satellite Images Using an Explainable Machine Learning Model," *Atmospheric Oceanic Opt.*, vol. 37, no. 3, pp. 400–408, 2024.

[16] N. Rajendiran, S. Sebastian, and L. S. Kumar, "Cloud Segmentation, Validation of Weather Data, and Precipitation Prediction Using Machine Learning Algorithms," *Arab. J. Sci. Eng.*, pp. 1–31, 2024.

[17] K. Li, N. Ma, and L. Sun, "Cloud detection of multi-type satellite images based on spectral assimilation and deep learning," *Int. J. Remote Sens.*, vol. 44, no. 10, pp. 3106–3121, 2023.