JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# SOIL MOISTURE RETRIEVAL USING GROUNDWATER USING MACHINE LEARNING

**[1]Varadha Neelima Devi, [2] Kakara Sai Meghana**
[1]Assistant Professor, [2]MCA 2nd year
[2]Master of Computer Applications,
[2]Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

***ABSTRACT***
Soil moisture plays a crucial role in various agricultural, hydrological, and environmental applications. Traditional methods for soil moisture retrieval often rely on in-situ measurements, which can be time-consuming and expensive. In recent years, machine learning techniques have emerged as a promising approach to estimate soil moisture using remote sensing data. However, most existing studies focus on using satellite-based observations, neglecting the valuable information provided by groundwater levels. This study proposes a novel approach for soil moisture retrieval using machine learning, specifically leveraging ground water data. The goal is to exploit the relationship between groundwater levels and soil moisture content to develop a reliable and accurate estimation model. The proposed method utilizes a combination of historical groundwater measurements and corresponding soil moisture data collected from in-situ sensors.

***INDEX TERMS*: Soil moisture retrieval, groundwater dataset, machine learning, feature engineering, spatiotemporal dynamics, remote sensing, water resource management.**

## I.INTRODUCTION

The use of deep learning, specifically Convolutional Neural Networks (CNNs), has shown promise in retrieving soil moisture from passive microwave remote sensing data. Soil moisture is a critical variable in the water cycle and has implications for various applications such as agriculture, drought prediction, and engineering construction projects.

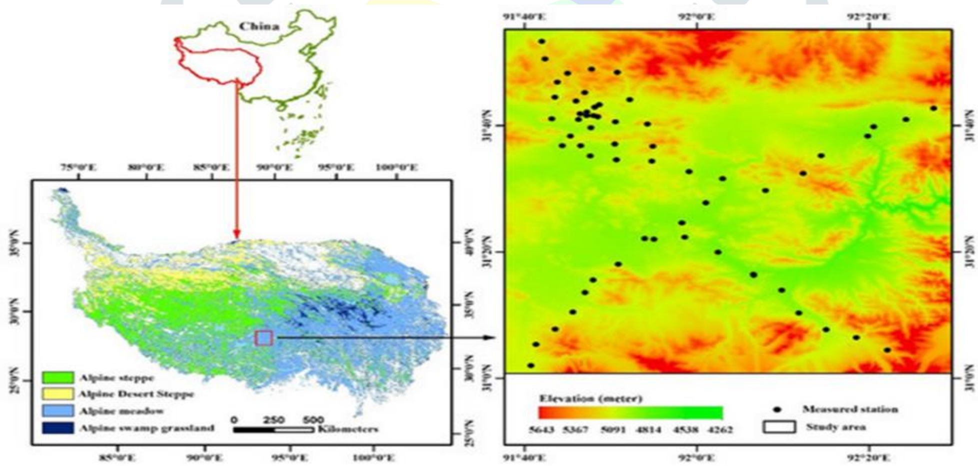


**Figure 1:** study area and soil moisture stations distribution

Traditional soil moisture retrieval algorithms, including SMOS Level 2 and Level 3 operational algorithms, SMAP Level 2 operational algorithm, dual channel algorithm (DCA), land parameter retrieval model (LPRM), and multi-orbit retrievals of soil moisture and optical depth (MT-DCA), have been proposed. These algorithms have provided valuable insights but may not be suitable for handling the increasing volume of remotely sensed data, commonly referred to as remote sensing big data.

In this context, the paper proposes the use of a deep learning-based empirical model for soil moisture retrieval. Deep learning is a type of machine learning algorithm that can learn complex features from data and perform complex tasks. With the exponential growth of remote sensing data, traditional retrieval methods, whether empirical or physical models, may struggle to handle the high complexity and non-linearity of the retrieval problems. Deep learning techniques, particularly CNNs, offer several advantages for soil moisture retrieval. They can effectively learn and extract features from the input data, enabling them to capture the complex relationships between the microwave measurements and soil moisture. Additionally, CNNs are well-suited for handling large-scale datasets due to their ability to parallelize computations and leverage the power of modern hardware accelerators.

By leveraging deep learning in the context of remote sensing big data, the proposed methodology aims to overcome the limitations of traditional retrieval methods and provide a fast and convenient model for soil moisture retrieval. This approach has the potential to improve the accuracy and efficiency of soil moisture estimation, thereby enhancing our understanding of the water cycle, supporting agricultural practices, and aiding in various engineering projects.

## II.PROBLEM STATEMENT

By using groundwater data, this research attempts to provide a machine learning-based system to estimate soil moisture levels. The issue entails gathering historical groundwater data from numerous sources, including groundwater levels and matching measures of soil moisture. In order to capture temporal patterns and relationships, relevant meteorological information and environmental factors will be taken into account while designing features. The ability of several machine learning techniques to forecast soil moisture levels—including regression models, support vector machines, and neural networks—will be tested. To train and fine-tune the selected models in order to increase their performance, the acquired dataset will be divided into training and validation sets. The ultimate objective is to deliver precise and fast information regarding soil moisture conditions, which has uses in hydrology, environmental monitoring, and agriculture.

### II.A)EXISTING SYSTEM

Support Vector Regression (SVR) is indeed a popular approach in the field of geo-/bio-physical parameter retrieval, including soil moisture retrieval. SVR is known for its good intrinsic generalization ability and robustness to noise, especially when there is limited availability of reference samples. It has been successfully applied in various applications, including remote sensing and environmental sciences.

However, traditional models, such as SVR, may have limitations in terms of their flexibility to learn more complex feature information. This is where deep learning approaches can offer advantages. Deep learning models, particularly neural networks, have shown great potential in capturing intricate relationships and patterns in data. They can automatically extract relevant features and learn from large datasets, which can potentially improve the accuracy and performance of soil moisture retrieval.

#### II.A.1)DISADVANTAGES OF EXISTING SYSTEM

1.Poor Performance

### II.B)PROPOSED SYSTEM

The paper you described proposes a deep learning approach for the inversion of soil moisture content using remote sensing data. The authors argue that deep learning models can capture complex features more effectively than classical algorithms, enabling real-time retrieval of soil moisture. The specific deep learning model used in the research consists of three pairs of convolutional layers and pooling layers, followed by a fully connected layer. The activation function of the top layer is changed from softmax loss to Euclidean loss. The input to the model is the brightness temperature images from the AMSR-E sensor, and the ground truth for soil moisture is obtained from the ECMWF model, which is considered to provide accurate soil moisture values.

To train the deep learning model, the authors utilize one month's worth of global data, comprising 30 pairs of images. This data is used to train the model, which is then employed to predict soil moisture maps for the following month. The performance of the deep learning model is evaluated by comparing the root-mean-square error (RMSE) and the R-square (R^2) with a Support Vector Regression (SVR) model.

The results of the experiment demonstrate that the deep learning method outperforms traditional retrieval algorithms, as indicated by lower RMSE values and higher R^2 values. The authors conclude that the deep learning model can better learn the complex relationship between observations and the ground truth and exhibits improved generalization performance for soil moisture retrieval compared to conventional algorithms like SVR.

Overall, the paper highlights the potential of deep learning techniques in soil moisture inversion from remote sensing data, showcasing their ability to handle complex features and achieve superior performance compared to traditional methods.

#### II.B.1)ADVANTAGES OF PROPOSED SYSTEM

Enormous amounts of data must be inverted. In less than 10 seconds after the model has been trained, a global soil moisture map can be forecasted. Moreover, the deep learning-based technique for retrieving soil moisture can acquire intricate texture properties from vast amounts of remote sensing data. The findings of this experiment show that the CNN used to recover global soil moisture can perform better than the support vector regression (SVR) for soil moisture retrieval.

## III.FEASIBILITY STUDY

Based on the provided information, it appears that a preliminary feasibility study has been conducted for a proposed system. Here's a breakdown of the feasibility aspects mentioned:

### III.A)Economic Feasibility:

The economic feasibility evaluates whether the development cost of the system is justified by the expected benefits. In this case, the system is considered economically feasible because it doesn't require additional hardware or software. The development is done using existing resources and technologies, resulting in nominal expenditure. The benefits derived from the system are expected to equal or exceed the costs.

### III.B)Operational Feasibility:

Operational feasibility assesses whether the proposed system can meet the organization's operational requirements and if there is sufficient support from management and users. According to the provided information, the management issues and user requirements have been taken into consideration during the planning phase. It is stated that there won't be any resistance from users that could undermine the potential benefits of the system. A well-planned design is expected to ensure optimal utilization of computer resources and improve performance.

### III.C)Technical Feasibility:

Technical feasibility examines whether the necessary technology exists to implement the proposed system effectively. It also considers factors like equipment capacity, system response, upgradeability, accuracy, reliability, ease of access, and data security. The information states that no system existed previously to cater to the needs of the "Secure Infrastructure Implementation System," but the current system developed is technically feasible. It is a web-based user interface for audit workflow and provides easy access to users. The database and software requirements for the project are available in-house or as open source. The system is expected to provide technical guarantees of accuracy, reliability, and security, and the necessary bandwidth exists to handle multiple users and provide fast feedback.

## IV.SYSTEM SPECIFICATIONS

### IV.A)Hardware Requirements:

System : Pentium i3
Hard Disk : 500 GB

Monitor : 14' Colour Monitor
Mouse : Optical Mouse
Ram : 4GB

**IV.B)Software Requirements:**

0perating system : Windows 7 OR ABOVE
Coding Language : PYTHON
Back end : Dataset

## V.CODE EDITORS

### V.A)PyCharm

PyCharm is an integrated development environment (IDE) used In computer programming, specifically for the Python language. It is developed by the Czech company Jet Brains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and

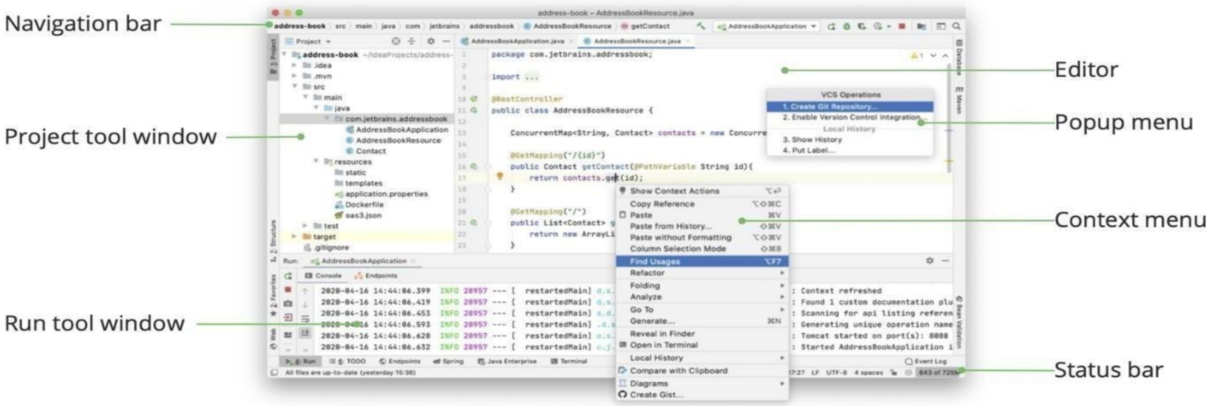


**Figure 2**: PyCharm screen

supports web development with Django as well as data science with Anaconda

•Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
•Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
•Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
•Integrated Python debugger
•Integrated unit testing, with line-by-line code coverage
•Google App Engine Python development
•Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
•Support for scientific tools like matplotlib, numpy and scipy [professional edition only]

PyCharm provide an API so that developers can write their own plugins to extend PyCharm features. Several plugins from other JetBrains IDE also work with PyCharm. There are more than 1000 plugins which are compatible with PyCharm

## VI.SOFTWARE VALIDATION

Software validation is an essential process in the software development life cycle (SDLC) that ensures the software meets the specified requirements and performs its intended functions accurately and reliably. It involves evaluating and verifying the software system to determine if it satisfies the user's needs and expectations. By conducting comprehensive validation activities, organizations can enhance the quality, reliability, and usability of their software products.

Here are some key aspects of software validation:

**VI.A)Requirement Validation:** This phase involves validating the software requirements to ensure they are complete, consistent, and aligned with the user's needs. It includes reviewing the requirements documentation, conducting meetings with stakeholders, and resolving any ambiguities or conflicts in the requirements.

**VI.B)Design Validation:** Design validation focuses on verifying the software's architectural and detailed design. It ensures that the design effectively translates the requirements into a system that can be implemented. This involves reviewing the design documents, conducting design inspections, and assessing the design's compliance with best practices and standards.

**VI.C)Test Planning and Execution:** Testing is a crucial component of software validation. It involves creating test plans, test cases, and test scripts to verify that the software functions as expected. Test planning includes defining test objectives, identifying test scenarios, and selecting appropriate testing techniques. Test execution involves running the tests, recording the results, and analyzing any deviations or failures.

**VI.D)Verification and Validation:** Verification and validation (V&V) activities are performed to ensure that the software meets the specified requirements. Verification focuses on checking whether the software is built correctly, while validation ensures that the right software is being built. This includes conducting inspections, walkthroughs, and reviews, as well as using various testing techniques such as unit testing, integration testing, system testing, and acceptance testing.

**VI.E)Compliance Validation:** In certain industries, software needs to comply with specific regulations, standards, or industry-specific requirements. Compliance validation involves assessing the software's adherence to these standards and ensuring that all necessary compliance measures are implemented. This may include performing audits, documenting compliance evidence, and conducting assessments against relevant regulations.

**VI.F)User Acceptance Testing:** User acceptance testing (UAT) involves validating the software from the end-user's perspective. It allows users to test the software in a real-world environment and provide feedback on its usability, functionality, and overall satisfaction. UAT helps identify any gaps between user expectations and the software's actual performance, ensuring that the software meets the needs of its intended users.

**VI.G)Documentation and Reporting:** Throughout the software validation process, it is essential to maintain proper documentation of the validation activities performed. This includes documenting test plans, test cases, test results, and any deviations or issues encountered during testing. Comprehensive reporting enables stakeholders to track the validation progress, make informed decisions, and ensure that the software is ready for release.

## VII.FUNCTIONAL REQUIREMENT

Functional requirements are an essential part of software engineering and systems engineering. They define the specific functions or behaviors that a system or its components should exhibit. Functional requirements describe the expected behavior between inputs and outputs, and they encompass calculations, technical details, data manipulation, and processing necessary for the system to achieve its intended goals.

Behavioral requirements, on the other hand, capture the various scenarios in which the system utilizes the functional requirements. These scenarios are typically documented in use cases, which illustrate how users interact with the system to accomplish specific tasks. Functional requirements are complemented by non-functional requirements, also known as quality requirements, which impose constraints on the system's design or

implementation. Non-functional requirements encompass aspects such as performance, security, reliability, and other characteristics that affect the system's overall quality.

Functional requirements are usually expressed as statements like "the system must do <requirement>," highlighting the specific actions or functions that the system needs to perform. In contrast, non-functional requirements are expressed as statements like "the system shall be <requirement>," focusing on the desired qualities or attributes of the system. The implementation plan for functional requirements is detailed in the system design, whereas non-functional requirements are addressed in the system architecture.

In the realm of requirements engineering, functional requirements outline the specific outcomes or results that the system should deliver. This is distinct from non-functional requirements, which encompass broader characteristics like cost and reliability. Functional requirements play a crucial role in shaping the application architecture of a system, while non-functional requirements guide the technical architecture.

During the requirements gathering and validation process, a requirements analyst may generate use cases based on the collected functional requirements. The sequence of activities involved in functional requirements collection and refinement typically follows this hierarchy: user/stakeholder request → analyze → use case → incorporate. Stakeholders articulate their requirements, and systems engineers engage in discussions, observations, and analysis to understand the different aspects of these requirements. Use cases, entity relationship diagrams, and other models are constructed to validate and verify the requirements. Once documented and approved, the requirements are implemented or incorporated into the system.

Use cases often serve as a starting point for requirements elicitation. Analysts derive the functional requirements necessary to support each use case, enabling users to perform their intended actions.

### VII.A)NON-FUNCTIONAL REQUIREMENT

Non-functional requirements (NFRs) are an essential part of software development. They define the attributes that are crucial for the success of a software system, focusing on aspects such as performance, usability, security, reliability, maintainability, and more. NFRs specify how the system should behave and how well it should perform in different conditions.

The example you provided, "how fast does the website load?" is a common non-functional requirement related to performance or responsiveness. Other examples of non-functional requirements could include:

**1.Usability:** The system should have an intuitive and user-friendly interface, allowing users to perform tasks efficiently and with ease.
**2.Security:** The system should implement robust security measures to protect sensitive data and prevent unauthorized access.
**3.Portability:** The system should be easily deployable and runnable on different platforms or environments without significant modifications.
**4.Scalability:** The system should be able to handle increasing workloads and adapt to accommodate a growing number of users or data.
**5.Reliability:** The system should operate consistently and reliably, minimizing the occurrence of failures or errors.

When defining non-functional requirements, it is important to make them measurable and specific. This allows for effective evaluation and verification of whether the system meets those requirements. Additionally, describing non-functional requirements with clear and concise language is crucial for effective communication among stakeholders, development teams, and testers.

By considering non-functional requirements alongside functional requirements, software development teams can ensure that the resulting system not only meets the intended functionality but also satisfies the expectations and needs of users in terms of performance, usability, security, and other important aspects.


**VII.A.1)ADVANTAGES OF NON-FUNCTIONAL REQUIREMENT**

The positive aspects of non-functional requirement are:

1.The software system complies with legal and compliance standards thanks to the nonfunctional requirements.
2.They guarantee the software system's dependability, accessibility, and effectiveness.
3.They make sure the software runs smoothly and the user experience is good.
4.The security policy of the software system is created with their assistance.

**VII.A.2)DISADVANTAGES OF NON-FUNCTIONAL REQUIREMENT**

The negative aspects of the non-functional requirement are:

1.The many high-level software subsystems may be impacted by no functional necessity.
2.Costs go up because they demand special attention throughout the software architecture/high-level design phase.
3.Once you have passed the architecture stage, it is difficult to modify non-functional code since their implementation typically does not map to the specific software sub-system.

**VIII.PREPROCESSING OF THE IMAGES**

**1.Image Registration:** This step involves aligning the AMSR-E data from consecutive days to obtain average images. For one day's data cannot cover the entire world, registration helps create composite images by combining data from multiple days.

**2.Spatial Interpolation:** It appears that the brightness temperature data and soil moisture truth values from ECMWF have different resolutions. To ensure compatibility, the soil moisture truth value images are projected onto an equal area grid with the same resolution as the brightness temperature data (0.25°). Spatial interpolation techniques are typically used for this purpose.

**3.Normalization:** The inputs (brightness temperature data and soil moisture truth values) and outputs (predicted soil moisture) are normalized to a range of [0, 1]. Normalization is a common practice to scale the data and bring it within a consistent range, which can benefit the training process.

**4.Generation of Patch Images:** After the preprocessing steps, the data is further processed to generate patch images of size 9×9. Patching is often employed to extract smaller image patches from the larger images, which can enhance the ability of the model to capture local spatial patterns.

**5.SVMAlgorithm**: Support Vector Machine (SVM) is a popular classification algorithm. In this step, the dataset is split into training and test sets. The SVM model is then trained using the training set, and the trained model is applied to the test data to calculate and evaluate the prediction accuracy.

**6.KNNAlgorithm:** K-Nearest Neighbors (KNN) is another classification algorithm. Similar to the SVM step, the dataset is split into training and test sets. The KNN model is trained using the training set, and then the trained model is applied to the test data to calculate and evaluate the prediction accuracy.

It's worth noting that the steps described above are just a high-level overview of the process, and there may be additional details or specific parameters that are not mentioned. The exact implementation and configurations of the algorithms may vary based on the specific CNN model and software framework being used.

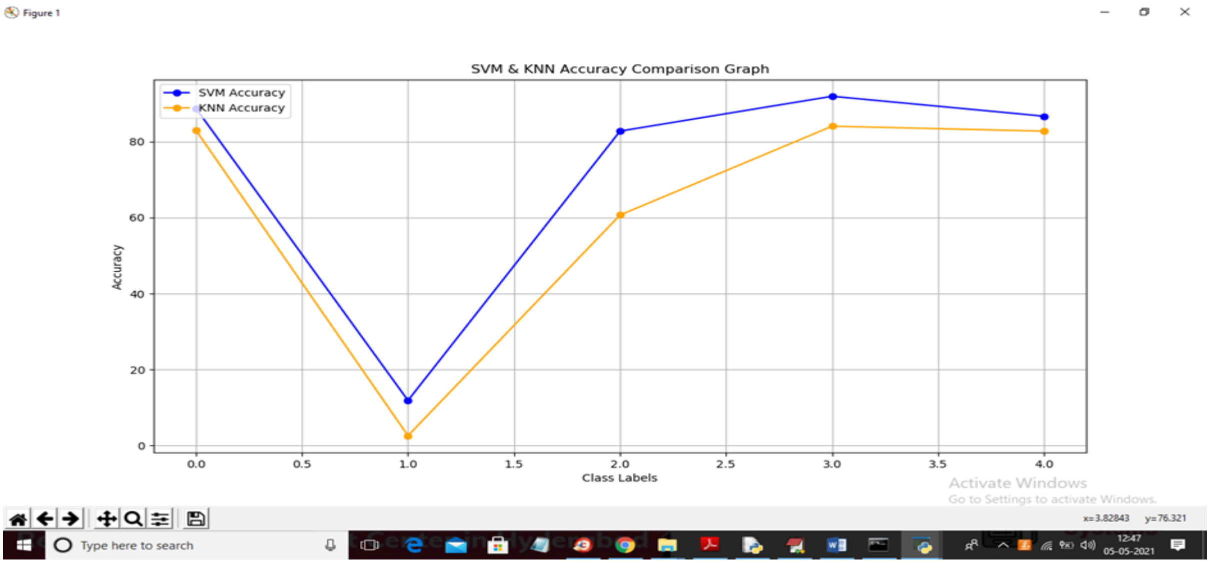


**Figure4:** SVM and KNN Accuracy Comparison Graph

## IX.PURPOSE

Using groundwater data, this research aims to create a machine learning-based system that can accurately estimate soil moisture levels. The project's goal is to establish the link between soil moisture levels and groundwater levels in order to offer useful data for a variety of industries, including agriculture, hydrology, and environmental monitoring. When it comes to planning irrigation, managing water resources, and maximising crop output, accurate and timely soil moisture measurement may help. This research uses machine learning to enhance our knowledge of soil moisture dynamics, support sustainable land use, and maximise the use of available water resources.

## X.CONCLUSION

In this project, a Convolutional Neural Network (CNN) was utilized to retrieve global soil moisture daily from AMSR-E brightness temperature images. The CNN took image patches of brightness temperature data as input and directly outputted the corresponding soil moisture values. Compared to the classical Support Vector Regression (SVR) approach, the CNN method achieved soil moisture results that were closer to the ground truth map. The training process of the CNN involved 31 images and took approximately 2 hours. Predicting the soil moisture for a single image required less than 10 seconds using an Nvidia GTX 1080Ti graphics card. With more powerful or a higher quantity of graphics cards, the training and prediction times could be significantly reduced. This means that once the model between brightness temperature and soil moisture is trained, the soil moisture retrieval can be efficiently parallelized across multiple GPUs.On the other hand, using SVR for predicting soil moisture on one image took more than two minutes, indicating a significant time difference compared to the CNN approach. Therefore, when it comes to the retrieval of soil moisture from large remote sensing data, CNNs offer a substantial advantage in terms of both prediction accuracy and computational cost when compared to traditional regression approaches.

## XI.REFERENCES

[1] Zeng, J., Li, Z., Chen, Q., Bi, H., Qiu, J., & Zou, P. (2015). Evaluation of remotely sensed and reanalysis soil moisture products over the tibetan plateau using in-situ observations. Remote Sensing of Environment,163, 91-110.
[2] Tuttle, S. E., & Salvucci, G. D. (2014). A new approach for validating satellite estimates of soil moisture using large-scale precipitation: comparing amsr-e products. Remote Sensing of Environment,142(3), 207-222.
[3] Dall'Amico, J. T., Loew, A., Schlenz, F., & Mauser, W. (2009). SMOS rehearsal campaign 2008: radiometer data analysis and soil moisture retrieval using the LPRM. Earth Observation and Water Cycle conferenceEarth Observation and Water Cycle conference.
[4] Cui, H., Jiang, L., Du, J., Wang, G., & Lu, Z. (2016). Assessment of QP model based two channel algorithm with JAXA, LPRM soil moisture products over Genhe area in China. Geoscience and Remote Sensing Symposium(pp.1663-1666). IEEE.
[5] Lu, Z., Chai, L., Zhang, T., Cui, H., & Li, W. (2017). Evaluation of amsr2 retrievals using observation of soil moisture network on the upper and middle reaches of heihe river basin. Remote Sensing Technology & Application.
[6] Kerr, Y. H., Waldteufel, P., Richaume, P., Wigneron, J. P., Ferrazzoli, P., & Mahmoodi, A., et al. (2012). The smos soil moisture retrieval algorithm. IEEE Transactions on Geoscience & Remote Sensing,50(5), 1384-1403.
[7] Wagner, W., Hahn, S., Kidd, R., Melzer, T., Bartalis, Z., & Hasenauer, S., et al. (2013). The ascat soil moisture product: a review of its specifications, validation results, and emerging applications. Meteorologische Zeitschrift,22(1), 5-33.
[8] Wang, S., Mo, X., Liu, S., Lin, Z., & Hu, S. (2016). Validation and trend analysis of ecv soil moisture data on cropland in north china plain during 1981–2010. International Journal of Applied Earth Observation & Geoinformation,48(48), 110-121.
[9] Buizza, R., Milleer, M., & Palmer, T. N. (1999). Stochastic representation of model uncertainties in the ecmwf ensemble prediction system. Quarterly Journal of the Royal Meteorological Society,125(560), 2887–2908.
[10] Wigneron, J. P., Jackson, T. J., O'Neill, P., Lannoy, G. D., Rosnay, P. D., & Walker, J. P., et al. (2017). Modelling the passive microwave signature from land surfaces: a review of recent results and application to the l-band smos & smap soil moisture retrieval algorithms. Remote Sensing of Environment,192, 238-262.
[11] Ali, I., Greifeneder, F., Stamenkovic, J., Neumann, M., & Notarnicola, C. (2015). Review of machine learning approaches for biomass and soil moisture retrievals from remote sensing data. Remote Sensing,7(12), 221-236.
[12] Durbha, S. S., King, R. L., & Younan, N. H. (2007). Support vector machines regression for retrieval of leaf area index from multiangle imaging spectroradiometer. Remote Sensing of Environment,107(1), 348-361.
[13] Rodríguez-Fernández>, N. J., Aires, F., Richaume, P., Kerr, Y. H., Prigent, C., & Kolassa, J., et al. (2015). Soil moisture retrieval using neural networks: application to smos. IEEE Transactions on Geoscience & Remote Sensing,53(11), 5991-6007.
[14] Kolassa,J., Gentine, P., Prigent, C., Aires, F., & Alemohammad, S. H. (2017). Soil moisture retrieval from amsr-e and ascat microwave observation synergy. part 2: product evaluation. Remote Sensing of Environment,195, 202-217.

[15] Slingo, J. M. (1987). The development and verification of a cloud prediction scheme in the ecmwf model. Quart.j.roy.meteor.soc,113.

**BIBILIOGRAPHY**

Varadha Neelima Devi working as Assistant professor in Master of computer application (MCA) sanketika vidya parishad engineering college, Visakhapatnam, Andhra Pradesh. with 8+ years of experience in area of interests c, cpp, java, python, sql, software engineering, data mining, cloud computing, operating systems, coding.

Kakara Sai Meghana studying her 2nd year, Master of Computer Applications in Sanketika Vidya Parishad Engineering College, affiliated to Andhra University, accredited by NAAC. With her interest in machine learning method and as a part of academic project, she used Soil moisture retrieval using groundwater using machine learning. As a result of a desire to comprehend. A completely developed project along with code has been submitted for Andhra University as an Academic Project. In completion of her MCA.