IoT-based Water Disaggregation in IWS System using ML and DL Techniques

S. Padole, P. S. Reddy, R. Yelekar, N. Nilesh, S. Chaudhari
International Institute of Information Technology, Hyderabad, India
Emails: {sahil.padole, ritik.yelekar, nitin.nilesh}@research.iiit.ac.in, sachin.chaudhari@iiit.ac.in

2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS) | 979-8-3315-3119-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/COMSNETS63942.2025.10885550

***Abstract*—This paper addresses the challenge of water disaggregation in intermediate water supply (IWS) systems using water level data. The water level node, in conjunction with four automatic labelling nodes, was deployed to collect data on appliance usage, no activity periods, and tank filling. Data is collected and transmitted to the cloud at two-second intervals via Wi-Fi networks. The collected water level data are used to train various machine learning (ML) and deep learning (DL) models for disaggregation in filling, appliance use such as geyser, flush, washing machine, and periods of no activity. These models are then evaluated to determine the most effective approach for water disaggregation, with performance assessed using accuracy and F1 score metrics. Out of the tested models, the Long Short-Term Memory (LSTM) model emerged as the best-performing algorithm, achieving an accuracy of 94.09% and an F1 score of 0.88.**

***Index Terms*—Classification, DL, IWS, IoT, ML, Water Level.**

I. INTRODUCTION

Water scarcity is a growing global concern, with only 2.5% of the world's water being freshwater, and this limited supply is diminishing due to increasing demand and contamination [1]. Effective water management is crucial, requiring detailed monitoring and analysis of consumption patterns. The use of IoT-based water monitoring enables real-time data collection of water consumption and levels, providing valuable insights for optimal water conservation efforts [2], [3].

Conventional water meters do not provide enough information for efficient water conservation. Water end-use disaggregation breaks down consumption into specific uses, such as toilets, washing machines, and geysers, and identifies wasteful practices like unnecessary faucet use. This detailed data helps users and policymakers develop effective water conservation strategies.

Recent years have seen significant advancements in water disaggregation techniques [4]–[9]. These studies have explored various sensing technologies and algorithmic approaches, each with its own advantages and limitations. Microphone-based sensing systems [4] offer non-intrusive monitoring but are susceptible to environmental noise, limiting their practical application. Pressure sensors [5] have been investigated for end-use appliance detection but face challenges due to reading variability based on sensor location. Flow sensors [6]–[8] provide accurate measurements but are costly, complex to install, and do not provide information about water levels in storage tanks.

Numerous studies have investigated various machine learning (ML) and deep learning (DL) techniques for water disaggregation [4]–[7], [9]. In the initial study [6], the authors used Hidden Markov Models (HMMs). The HMMs require extensive manual feature engineering and are limited in identifying complex relationships. In [7], ML methods such as Random Forest, Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP) were applied to water disaggregation, utilizing feature engineering and time series analysis on flow sensor data. DL techniques have been explored by creating sophisticated architectures using Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) models. The study [9] implemented DL architectures such as Neural Fourier Energy Disaggregation (NFED), Self-Attentive Energy Disaggregation (SAED), and Weighted Gated Recurrent Unit (WGRU) by applying transfer learning from electricity data models. Although electricity and water flow data patterns correlate, water level data exhibits distinct trends compared to the more constant patterns observed in electricity usage, making transfer learning ineffective. Some studies [4], [5] lack explicit mention of classification algorithms and comparative analysis of various models. Manual segmentation of events [5] can yield high-quality labels but may limit scalability in complex, real-world scenarios.

Recent research on water disaggregation [4]–[9] has been focused on Western countries where continuous water supply (CWS) systems are prevalent. CWS systems [10] maintain a more stable and reliable water distribution by providing a direct supply from the mains, ensuring consistent pressure and flow rates across all points of use. This stability facilitates accurate measurement and disaggregation of water consumption. In contrast, India predominantly uses the IWS system for water management and distribution. In IWS systems, the supply to the consumers is not available round the clock through the mains [10]. To overcome this in IWS systems, consumers typically rely on consumer-side storage tanks to regulate the water supply during non-supply hours. Thus the methods used for disaggregation in CWS are not directly applicable, as the end consumption happens through the water tanks In this approach, water pressure within the tanks fluctuates, and as the water level in the storage tank decreases, flow rates may drop as well. These variations in pressure and flow make it challenging to use flow or pressure sensors for accurate water disaggregation in IWS systems.

To address the challenges of IWS, the paper proposes using smart water level nodes inside overhead tanks (OHT). These non-intrusive, cost-effective devices can be easily installed in tanks without requiring modifications to the existing infrastructure. They enable real-time monitoring and continuous data collection, providing insights into water disaggregation patterns.

The specific contributions of this paper are:

- An IoT-based mechanism is proposed to estimate water disaggregation in the IWS system using water level measurements.
- For this, an IoT-based water level node (also called a disaggregation node) and four automatic labeling nodes were developed and deployed to collect relevant parameters at the test lab on our campus in India for over five weeks. The disaggregation categories considered were filling, geyser, washing machine, flush and no activity.
- Different ML and DL algorithms are employed for water disaggregation using the water level data. ML algorithms considered are Random Forest, Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) [11] while DL algorithms considered include Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), Bidirectional LSTM (BiLSTM), and CNN-LSTM [12].
- A slope feature has been introduced to improve the performance of the models.
- The trained models are evaluated using performance parameters of accuracy, macro F1 score, and confusion matrix.

The structure of the rest of the paper is as follows. Section II covers the details of an IoT-based water level monitoring device (the disaggregation node). Section III discusses the data measurement campaign, while Section IV details the data analysis and methods. Experimental results are presented in Section V, and Section VI concludes the paper.

II. IoT-Based Water Level Monitoring

Figs. 1, 2, and 3 shows the block architecture, circuit diagram, and the designed water level node. This node is designed for water level monitoring in tanks, particularly in OHT. At its core, the system utilizes an ESP32 microcontroller [13], chosen for its powerful processing capabilities, integrated Wi-Fi module, and low power consumption. Table I shows the specification of sensors used in the deployed sensor node. For water level measurement, the node employs the JSN SR04T ultrasonic sensor [14]. This waterproof sensor offers an extended range of up to 6 m and higher accuracy compared to standard ultrasonic sensors, making it ideal for use in tanks or water-exposed environments. It also includes the DS18B20 temperature sensor [15], which is also a waterproof sensor, providing a digital output and low power consumption. The temperature data is crucial for improving the accuracy of water level measurements through temperature compensation, as done in [16]. The ESP32 microcontroller processes data from both sensors and, using Wi-Fi transmits it to ThingSpeak [17], a cloud-based IoT platform. Power management is handled by an HLK-10M05 AC to DC converter [18], which supplies a consistent 5V, 2A power to all components. For protection and durability, all components are enclosed in an IP65-rated polycarbonate box as illustrated in Fig.**??**, ensuring the system's resilience in various environmental conditions and are designed for deployment in OHTs.

TABLE I: Specification of sensors used in the deployed sensor node

Sensor	Parameters	Accuracy	Resolution
JSN-SR04T	Distance	±1 cm	1 mm
DS18B20	Temperature	±0.5°C	0.1°C

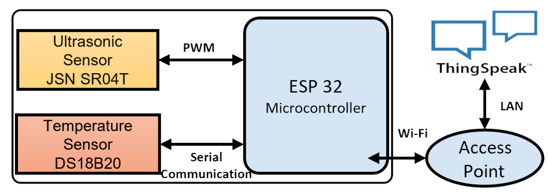


Fig. 1: Block architecture of water level monitoring node

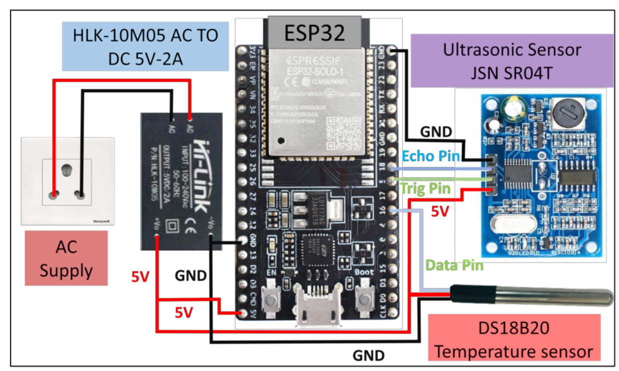


Fig. 2: Circuit diagram of water level monitoring node

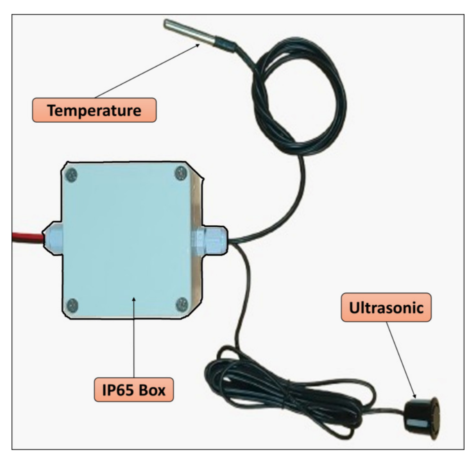
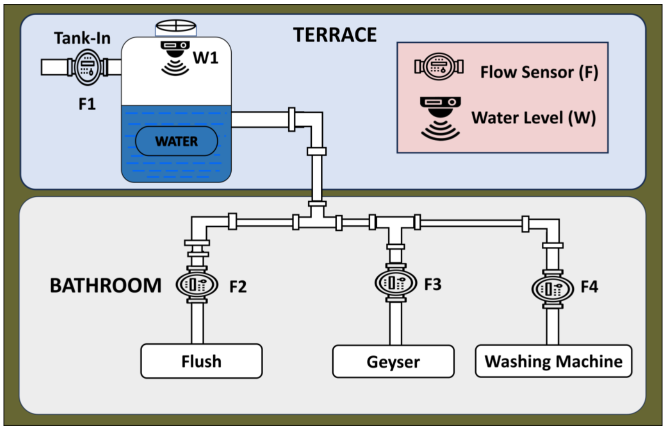
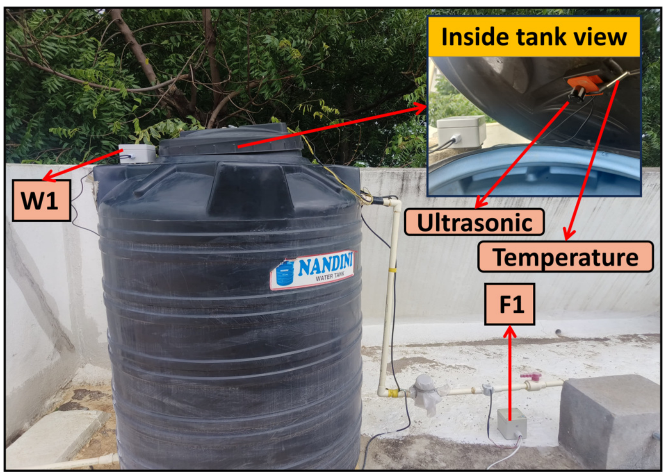


Fig. 3: Designed water level node

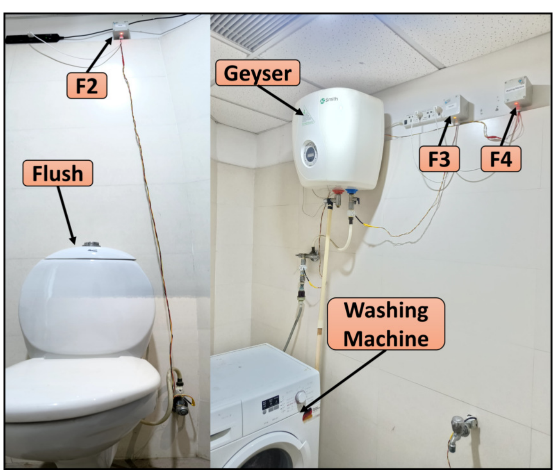
III. Data Measurement campaign



(a) Node deployment schematic



(b) Overhead tank deployment



(c) Indoor deployment

Fig. 4: Node deployment schematic and locations

Fig. 4 shows the deployment of water monitoring nodes at the smart home test lab spanning strategic locations. On the terrace, a Tank-In flow meter node is installed at the inlet pipe, while a Water level node is positioned on top of the 500 L OHT. Individual flow meter nodes in the bathroom are fixed to the flush, geyser, and washing machine appliances. This setup, illustrated in Fig. 4a, shows a node deployment schematic for monitoring water filling, tank water storage levels, and appliance-specific consumption, providing a holistic view of the water usage patterns within the test environment. Figs. 4b and 4c show overhead tank and indoor deployment.

Data was collected from two types of nodes:

- *Disaggregation Node*: The node on the OHT (W1) continuously tracks filling events, periods of inactivity, and overall water consumption patterns.
- *Automatic Labeling Nodes*: The flow sensor nodes tank-in (F1), flush (F2), geyser (F3), and washing machine (F4) for labels. Each node is equipped with an ESP-32 microcontroller [13] and a YS-F201 flow sensor [19] to capture flow readings, creating a labeled dataset essential for training ML and DL models.

Data was collected from June 4 to July 9, 2024, using deployed nodes. Measurements were taken at 2-second intervals during weekday daytime hours. The tank is filled once per day. Four readings are taken for each appliance, including the geyser, flush, and washing machine. Between two consecutive readings of activity data, a break time has been incorporated to create a period of no activity. Two datasets were created: a disaggregation dataset and an automatic labeling dataset. Each water level dataset entry contains a timestamp and a water level, while each automatic labeling dataset entry contains only a timestamp. A combined water flow dataset from various appliances was used to label the water level dataset during preprocessing. Table II presents the number of data points collected for each class in both datasets.

TABLE II: Number of samples in each class in raw data.

Nodes	No. of data points
Flush	7405
Washing Machine	7291
Geyser	5161
Filling	45669
No Activity	89567
Total	**155093**

IV. Data Analysis and Methods

A. Data Preprocessing

Fig. 5 illustrates the essential preprocessing steps applied to raw sensor data, preparing it for model training.

1) Outlier Removal

Removing outliers from the raw disaggregation dataset is essential, as extreme values can significantly skew the analysis.

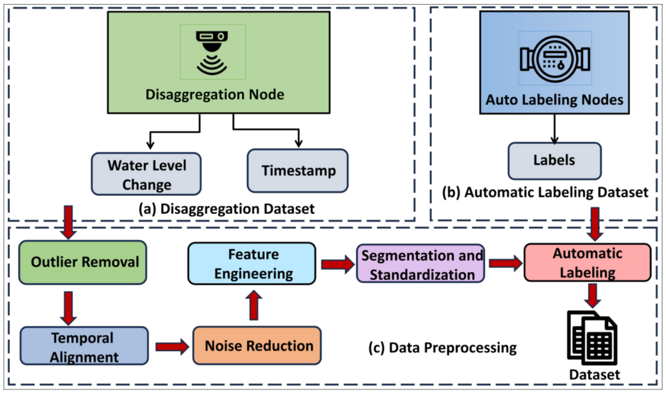


Fig. 5: Data collection and processing pipeline

The residuals were extracted from the time series using Seasonal and Trend decomposition using Loess method [20]. Outliers were then identified and removed based on residuals exceeding three times the standard deviation. In this process, 3.396% of the data was removed as outliers.

2) Temporal Alignment

To ensure the data used for model training was temporally aligned, we implemented a specific process. The original high-resolution data was resampled to create three distinct datasets with intervals of 2 s, 4 s, and 6 s to check the model performance for different sampling intervals.

3) Noise Reduction

The raw disaggregation dataset consists of noise due to ripples formed in water. A low-pass filter [21] was implemented to reduce noise in the data. The cutoff frequency of 0.008 Hz was established at the point where 97% of the signal power was retained, effectively filtering out the remaining 3% considered high-frequency noise.

4) Feature Engineering

Feature extraction involved slope calculation through first-order difference data. Fig. 6 employs box plots to illustrate the unique slope distributions corresponding to the different labels in the final dataset. This feature captured water level change rates, providing a distinctive feature for differentiating various water usage events.

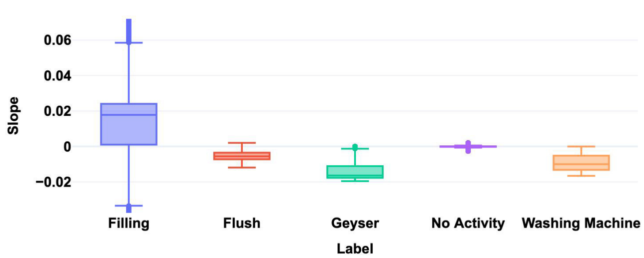


Fig. 6: Boxplot of slope values for different labels

Fig. 7: Feature vector representation across multiple data windows

5) Segmentation and Standardization

The data was segmented into time series windows of size N data points. These windows were designed to capture both historical context and temporal patterns, providing a comprehensive view of each water usage event. Fig. 7 shows the final feature vector of the dataset, where N is the size of the window. Finally, standardization was implemented, setting the features to a mean of 0 and a standard deviation of 1. This standardization is crucial for ML and DL models, as it prevents features with larger ranges from dominating the objective function and ensures the algorithm treats all features equally.

6) Automatic Labeling

The data was labeled by correlating timestamps of the automatic labeling dataset with the water level dataset. This yielded five distinct labels: flush, geyser, washing machine, filling, and no activity.

B. Model Training

To comprehensively capture the diverse features of water usage patterns, a range of learning algorithms were employed. This approach encompassed both traditional ML techniques and advanced DL architectures, each selected for its specific strengths in handling time series data and multi-class classification problems. The models were trained using 80% of the data, while the remaining 20% was used for testing.

1) ML Algorithms

In ML [11], the models Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest were implemented. Logistic Regression was chosen for its direct probability prediction capabilities and interpretability, utilizing the LBFGS solver for rapid and reliable convergence in high-dimensional datasets. SVM, implemented with an RBF (Radial Basis Function) kernel and one-vs-one (OvO) strategy, excelled at managing complex, non-linear relationships crucial for distinguishing closely related water usage categories. KNN was selected for its simplicity and effectiveness in scenarios where data points exhibit clear class distinctions based on similarity, making it well-suited for analyzing well-defined usage patterns. Random Forest was employed for its robustness in handling large, high-dimensional datasets. Its ensemble approach, combining multiple decision trees, enhanced classification accuracy and captured intricate feature relationships while mitigating overfitting.

2) DL Algorithms

DL architectures [12], [22] considered were LSTM, GRU, CNN, BILSTM, and CNN-LSTM. LSTM and GRU networks were chosen to address the vanishing gradient problem in traditional Recurrent Neural Networks, enabling the retention of long-term dependencies in sequential water level data. While LSTM offers a more complex gating mechanism, GRU's streamlined architecture provides faster training and lower memory requirements, making it ideal for deployment in resource-constrained environments common in water monitoring applications. The BiLSTM model was implemented, extending LSTM's capability by processing input sequences in both directions to enhance the capture of contextual information crucial for distinguishing water usage patterns. A one-dimensional CNN was implemented to capture patterns in the data, recognizing local relationships in sequential water level data. Finally, a hybrid CNN-LSTM model was developed, combining the strengths of convolutional layers for local pattern extraction with LSTM's ability to model long-term dependencies.

The categorical cross-entropy loss and the Adam optimizer were employed to train the DL models. To refine the training process and prevent overfitting, the implementation of several callbacks, including early stopping was done. This technique halts training when the model's performance on the validation set ceases to improve, effectively balancing model complexity and generalization capability.

C. Model Evaluation

Performance assessment of water disaggregation models employed two primary metrics: accuracy and macro F1 score [11]. Accuracy quantifies overall classification performance, while macro F1 score assesses the model performance across all categories in multi-class classification. The macro F1 Score is calculated by computing the harmonic mean of precision and recall for each class and averaging the values of all classes. The metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2}$$

where Precision and Recall are given by,

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

In the above equations, True Positive (TP) denotes correctly predicted positive instances, False Positive (FP) represents negative instances incorrectly classified as positive, True Negative (TN) indicates correctly predicted negative instances, and False Negative (FN) signifies positive instances incorrectly classified as negative.

To ensure robust evaluation and mitigate potential overfitting, a 5-fold cross-validation methodology was implemented in ML models. This technique involves partitioning the dataset into five subsets, training the model on four subsets, and validating the remaining subset. The confusion matrices were generated for each model to provide a visual representation of performance across all classes. These matrices facilitate the identification of classes where the model excels, classes prone to misclassification, and patterns in errors that may indicate model bias or limitations.

V. Results

The time series plot of the disaggregation dataset, shown in Fig. 8, illustrates changes in water levels over time, showcasing the impact of various household activities on water consumption. The graph reveals patterns of rapid filling (steep upward slopes), periods of no activity (flat lines), flushing usage (gradual decline), washing machine operation (slightly steeper decline) and geyser (steepest downward slope). The visualization provides valuable insights into water consumption patterns in a household setting, showing how different appliances and activities contribute to water usage.

Fig.9 shows the performance of models with and without slope at 4 s sampling interval and N=250 window size. It illustrates that incorporating the slope feature enables the models to identify classes more accurately, resulting in improved water disaggregation. The GRU model performs well even without the slope feature. After adding the slope feature, the machine learning models' performance approaches that of the deep learning models. Suggesting that extensive feature engineering and proper data preprocessing can significantly enhance ML model performance, potentially minimizing the performance gap between ML and DL models.

Experiments were conducted to evaluate the performance of ML and DL models across various sampling intervals and window sizes. The performance of the ML and DL at different sampling intervals with a window size of N=250 is shown in Fig. 10. All the models performed very close to each other at various sampling intervals. The sampling does not affect the performance of the model much. Fig. 11 demonstrates the performance of ML and DL with various window sizes using a 4 s sampling interval. The model's performance shows minimal variation across different window sizes.

The accuracy and F1 score performance of ML and DL models at 4 s sampling interval and window size of N=250 are presented in Table III. The LSTM model exhibited the highest performance among the evaluated models, attaining an accuracy of 94.09% and an F1 score of 0.88.

In ML model comparison, Random Forest outperformed with an accuracy of 93.75% and an F1 score of 0.87 than other ML models due to its ensemble learning. Random Forest can handle high-dimensional datasets and capture complex, non-linear relationships between features. The SVM displayed the next-best performance with an accuracy of 90.96% and an F1 score of 0.81. SVM with an RBF kernel can efficiently map input features to a higher-dimensional space, allowing for better separation of classes in complex, non-linear data distributions. On the other hand, the KNN algorithm showed poor performance, with an accuracy rate of just 53.92% and an F1 score of 0.55. The main reason for this inadequate performance

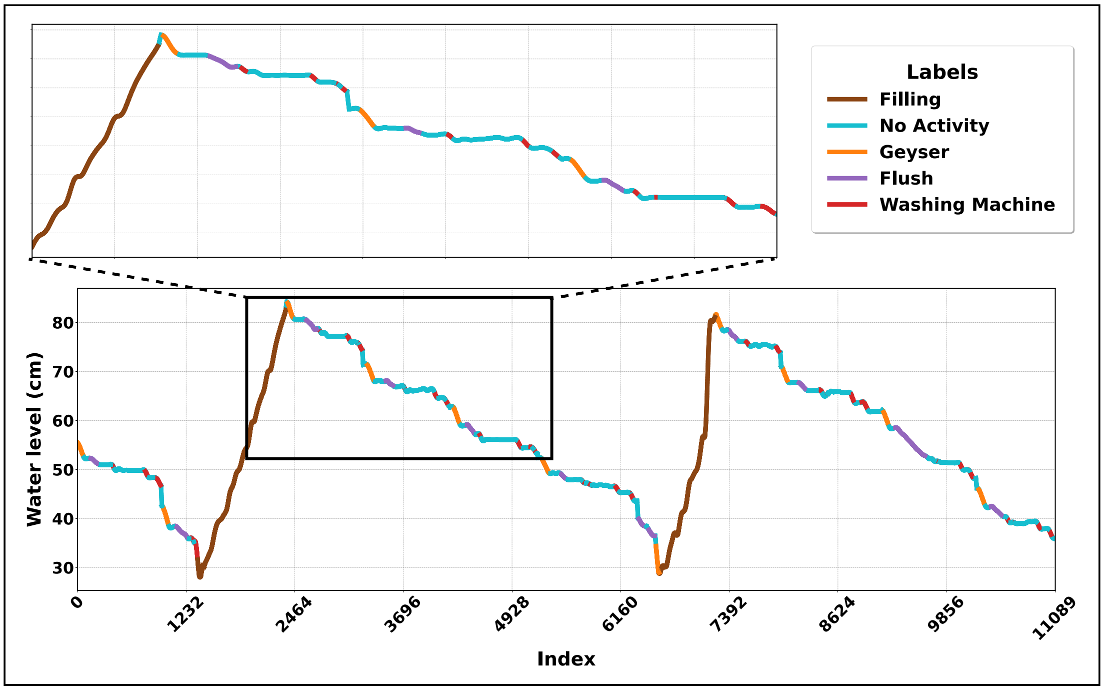


Fig. 8: Time series plot of disaggregation dataset

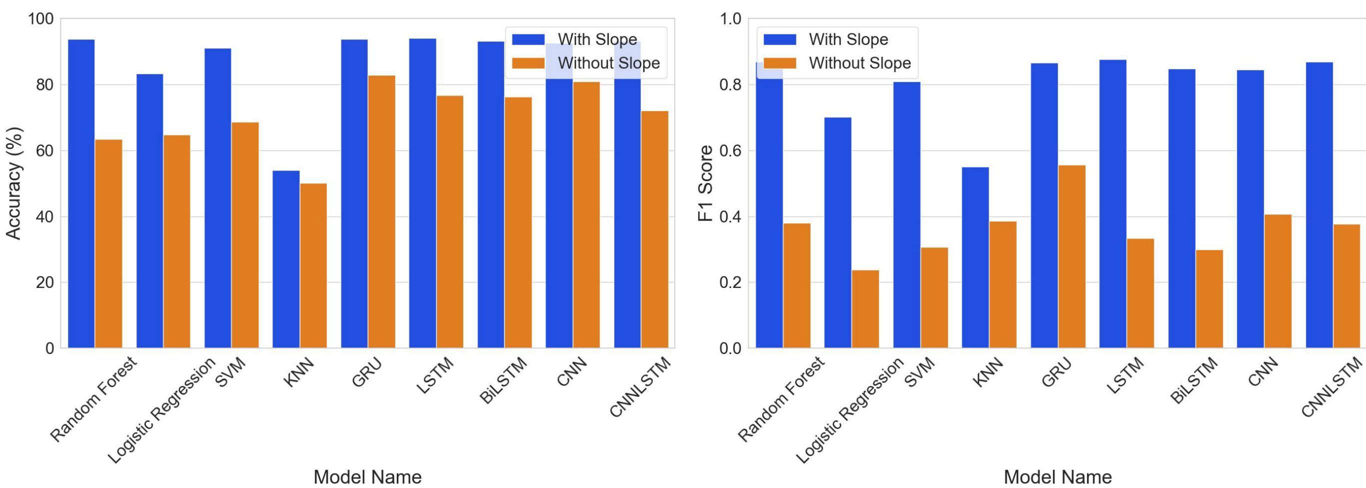


Fig. 9: Performance of models in terms of Accuracy and F1 score, with and without slope as a feature

is KNN algorithm relies on measuring distances between data points or identifying nearest neighbors for classification. As the number of dimensions increases, data points become more spread out, and the nearest neighbors identified by KNN may not be truly relevant for accurate classification.

In DL model evaluation, the models LSTM, GRU, CNN, BiLSTM, and CNN-LSTM models achieved similar performance, with accuracies ranging from 92.54% to 94.09% and F1 scores between 0.84 and 0.88. The LSTM, GRU, and BILSTM models capture sequential patterns, manage non-linear relationships, and effectively retain important information over long sequences. The CNN models focus on capturing spatial patterns in the data. The hybrid CNN-LSTM model exhibited slight progress than the CNN model. The confusion matrix of the LSTM model at 4 s sampling interval with window size of N=250 is presented in Fig. 12. From the observation of the confusion matrix, the classifier identifies no activity and filling states accurately. At some data points of flush, washing machine, and geyser activities are misclassified as no activity. This misclassification occurs because the initial data points for these activities may have characteristics similar to the no activity state. In some data points, the change in slope for flush and washing machine activities is similar, leading to misclassification between these labels.

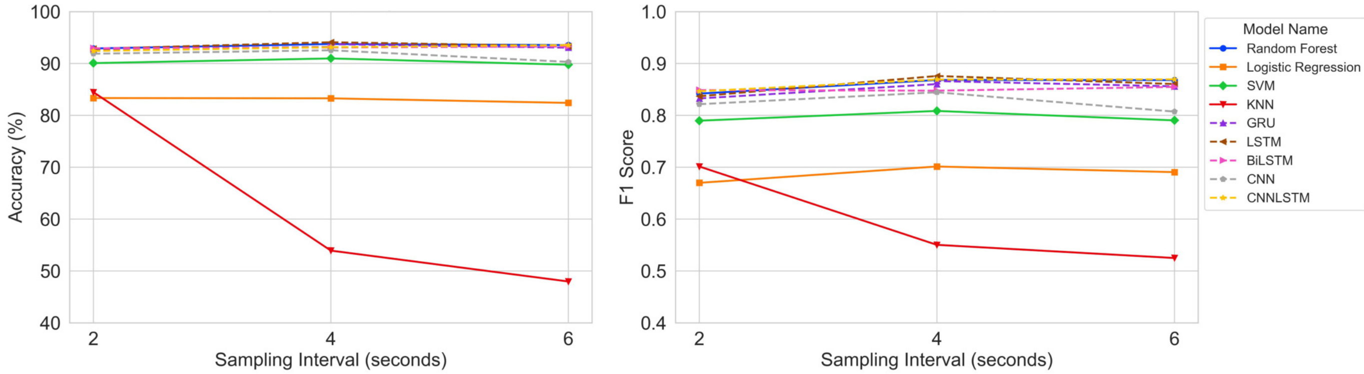


Fig. 10: Performance of models in terms of Accuracy and F1 score across different sampling intervals.

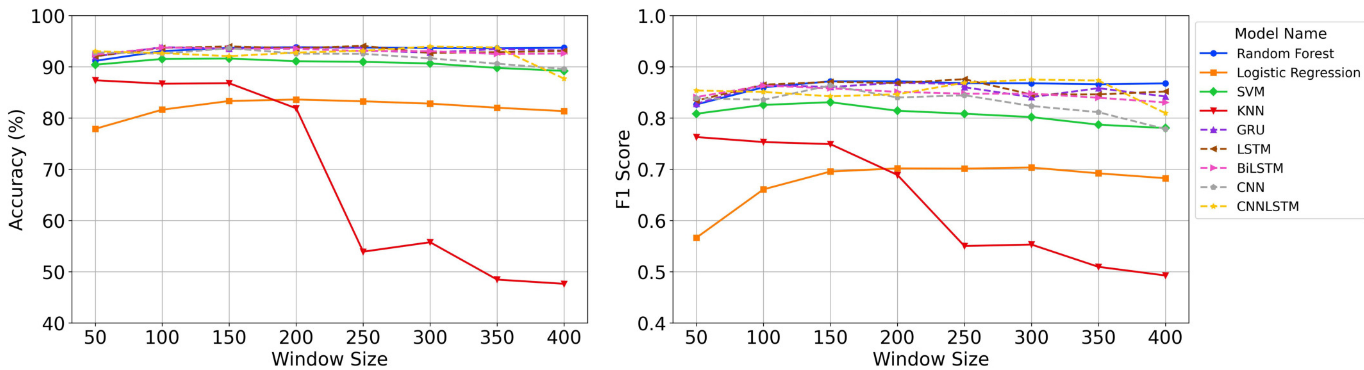


Fig. 11: Performance of models in terms of Accuracy and F1 score across various window sizes.

TABLE III: Performance of ML/DL algorithms for water disaggregation

	Model	Accuracy(%)	F1 Score
ML	Random Forest	**93.75**	**0.87**
	Logistic Regression	83.27	0.7
	SVM	90.96	0.81
	KNN	53.92	0.55
DL	GRU	93.71	0.87
	LSTM	**94.09**	**0.88**
	BiLSTM	93.08	0.85
	CNN	92.54	0.84
	CNN-LSTM	93.13	0.87

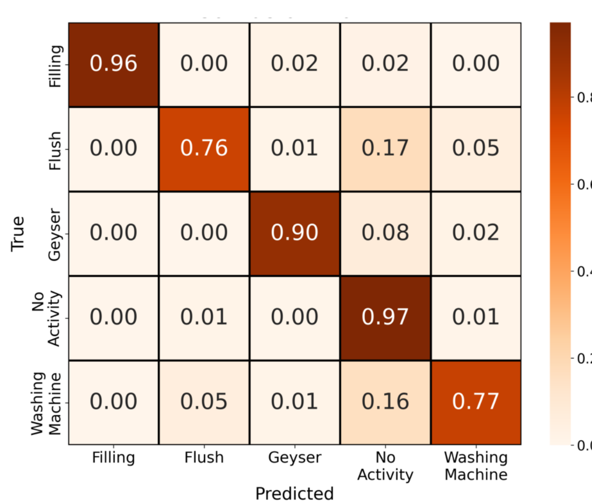


Fig. 12: Confusion matrix of classification of individual appliance usage

VI. Conclusion

This paper demonstrates that water level data from an overhead tank of the IWS system can be used to disaggregate water usage in a home setting. It shows that water level can identify the usage of different devices, water-filling occurrences, and periods of non-use using ML and DL algorithms. The results indicate that all models performed well with an accuracy ranging from 83.27% to 94.09% except KNN, with LSTM giving the best accuracy of 94.09%. In general, ML models showed close performance to DL models. The variations in sampling intervals and window sizes did not significantly impact model performance. Additionally, through effective feature engineering, such as the addition of slope feature, the models could distinguish between different classes and enhance the performance of models in water disaggregation.

VII. Acknowledgements

This research was supported by the Ministry of Electronics and Information Technology (MEITY), Govt of India under grant no. 3070665 (2020) as part of the Smart City Living Lab project, with no conflict of interest. The authors thank Mr. Balaji, Senior Technician for discussions on the observations from this work with respect to the actual water infrastructure on the IIIT-H campus.

References

[1] S. R. Carpenter, E. H. Stanley, and M. J. Vander Zanden, "State of the world's freshwater ecosystems: Physical, chemical, and biological changes," *Annual Review of Environment and Resources*, vol. 36, no. 1, pp. 75–99, 2011.

[2] P. Lea, *Internet of Things for Architects: Architecting IoT Solutions by Implementing Sensors, Communication Infrastructure, Edge Computing, Analytics, and Security*. Packt Publishing, 2018.

[3] S. Misra, A. Mukherjee, and A. Roy, *Introduction to IoT*. Cambridge University Press, 2021.

[4] J. Fogarty, C. Au, and S. E. Hudson, "Sensing from the basement: A feasibility study of unobtrusive and low-cost home activity recognition," in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '06, 10 2006, pp. 91–100.

[5] J. Froehlich, E. Larson, T. Campbell, C. Haggerty, J. Fogarty, and S. Patel, "HydroSense: Infrastructure-mediated single-point sensing of whole-home water activity," in *Proceedings of the 11th International Conference on Ubiquitous Computing*, ser. UbiComp '09, 09 2009, pp. 235–244.

[6] B. Ellert, S. Makonin, and F. Popowich, "Appliance water disaggregation via non-intrusive load monitoring (NILM)," in *Proceedings of the 4th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2015)*, vol. 166, 10 2015.

[7] P. V. Pavlou, S. Filippou, S. Solonos, S. G. Vrachimis, K. Malialis, D. G. Eliades, T. Theocarides, and M. M. Polycarpou, "Monitoring domestic water consumption: a comparative study of model-based and data-driven end-use disaggregation methods," *Journal of Hydroinformatics*, vol. 26, no. 4, pp. 709–726, 03 2024.

[8] A. Di Mauro, A. Di Nardo, G. Santonastaso, and S. Venticinque, "An IoT system for monitoring and data collection of residential water end-use consumption," in *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, 08 2019.

[9] N. V. Gkalinikis and D. Vrakas, "Efficient deep learning techniques for water disaggregation," in *2022 2nd International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED)*, 2022, pp. 1–6.

[10] CPHEEO, Ministry of Housing And Urban Affairs, Govt. of India, "Guidelines for planning, design and implementation of 24x7 water supply systems," Ministry of Housing And Urban Affairs, Govt. of India, New Delhi, India, Tech. Rep., Dec. 2021, [Accessed: September 13, 2024]. [Online]. Available: https://mohua.gov.in/pdf/62023520928a0advisory-on-water%20supply-final-new.pdf

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[12] F. Chollet, *Deep Learning with Python*, 2nd ed. Manning, 2021.

[13] *ESP32 Series Datasheet*, Espressif Systems, [Accessed: September 13, 2024]. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[14] *JSN-SR04T-2.0 Ultrasonic Module*, Jaycar Electronics, [Accessed: September 13, 2024]. [Online]. Available: https://www.jaycar.com.au/medias/sys_master/images/images/9404693217310/XC3708-dataSheetMain.pdf

[15] *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*, Maxim Integrated, [Accessed: September 13, 2024]. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

[16] R. Yelekar, T. D. Tency, Aishwaryabharathi, S. Madbushi, and S. Chaudhari, "IoT-based smart water level monitoring," in *2023 IEEE 20th India Council International Conference (INDICON)*, 2023, pp. 1136–1142.

[17] ThingSpeak, "ThingSpeak IoT analytics platform," https://thingspeak.com/, [Accessed: September 13, 2024].

[18] *Hi Link HLK 10M05 – For 5V 2A – AC to DC Power Supply Module*, Hi-Link Electronic, [Accessed: September 13, 2024]. [Online]. Available: http://www.hlktech.net/product_detail.php?ProId=54

[19] *YF-S201 Hall Effect Water Flow Meter / Sensor*, Seeed Studio, [Accessed: September 13, 2024]. [Online]. Available: https://www.seeedstudio.com/G1-2-Water-Flow-Sensor-p-635.html

[20] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2018.

[21] S. Basu and S. Mamud, "Comparative study on the effect of order and cut off frequency of butterworth low pass filter for removal of noise in ECG signal," in *2020 IEEE 1st International Conference for Convergence in Engineering (ICCE)*, 2020, pp. 156–160.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.