Intelligent Campus Network Monitoring and Optimization

IoT and Machine Learning

Final Review

Hemesh Yeturu	(CB.SC.U4AIE23166)
Joel John	(CB.SC.U4AIE23131)
Abhishek Sankaramani	(CB.SC.U4AIE23107)
Adarsh P	(CB.SC.U4AIE23109)

Introduction

- The primary goal of this research is to develop a real-time indoor localization system based on Wi-Fi RSSI values using ESP32 devices for applications such as distance estimation, environmental classification, and dead zone detection.
- The existing research shows the potential of RSSI-based localization but often struggles with challenges such as signal degradation, multi-path interference, and dynamic environments.
- The paper addresses these challenges by integrating machine learning models to improve the accuracy of distance estimation, environmental classification, and dead zone detection.
- Previous studies, such as those by Neupane et al. (2024), Nakatani et al. (2018), and Singh et al. (2021), have explored Wi-Fi localization, but none provide a real-time, adaptable solution for dynamic indoor environments with changing signal strength.
- The proposed system leverages the ESP32-based hardware to collect real-time data and machine learning models for robust, scalable, and cost-effective solutions to address these gaps.

Organization of the Report

- Chapter 1: Introduction
  Overview, literature, problem statement, and objectives.
- Chapter 2: Background
  Key technologies and system foundation.
- Chapter 3: Proposed Work
  System design, ESP32 setup, and ML integration.
- Chapter 4: Results and Discussion
  Model performance analysis and observations.
- Chapter 5: Conclusion and Future Work
  Summary and scope for future enhancements.

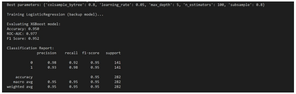
Objectives

- To develop a real-time indoor localization system using ESP32 devices that leverages Wi-Fi RSSI values to estimate distances and determine the locations of devices within a given area.
- To integrate machine learning models for accurate distance estimation, environmental classification, and dead zone detection based on real-time RSSI measurements.
- To classify environments as either indoor or outdoor based on Wi-Fi signal characteristics, enhancing the adaptability of the system to various real-world environments.
- To introduce a novel dead zone detection technique that identifies weak or unavailable Wi-Fi signal areas, providing real-time feedback for network optimization.
- To design a low-cost, scalable solution using ESP32 devices that can be easily deployed for large-scale networks and real-time Wi-Fi optimization tasks.

Methodology Approach

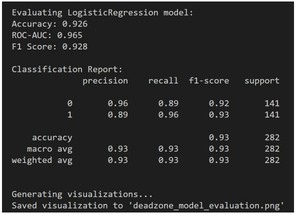
- ESP32 devices are used to collect real-time RSSI data from Wi-Fi networks in indoor and outdoor environments.
- The relationship between RSSI and distance is modeled using the log-distance path loss formula.
- Collected data is sent to a central server for processing and fed into machine learning models.
- Machine learning models are trained for distance estimation, environmental classification, and dead zone detection.
- Real-time predictions are made using trained models, and results are visualized for network optimization.
- The system operates in low-power mode for efficiency and is evaluated using metrics like MAE, RMSE, R^2, accuracy, and F1-score.

Dead Zone Detection Model Performance Evaluation



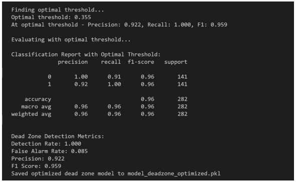
```
Best parameters: {'colsample_bytree': 0.8, 'learning_rate': 0.05, 'max_depth': 5, 'n_estimators': 100, 'subsample': 0.8}

Training LogisticRegression (backup model)...

Evaluating XGBoost model:
Accuracy: 0.950
ROC-AUC: 0.977
F1 Score: 0.952

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.92      0.95       141
           1       0.93      0.98      0.95       141

    accuracy                           0.95       282
   macro avg       0.95      0.95      0.95       282
weighted avg       0.95      0.95      0.95       282
```

Figure: (a) XGBoost and Logistic Regression training



```
Evaluating LogisticRegression model:
Accuracy: 0.926
ROC-AUC: 0.965
F1 Score: 0.928

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.89      0.92       141
           1       0.89      0.96      0.93       141

    accuracy                           0.93       282
   macro avg       0.93      0.93      0.93       282
weighted avg       0.93      0.93      0.93       282


Generating visualizations...
Saved visualization to 'deadzone_model_evaluation.png'
```

Figure: (b) Logistic Regression classification report



```
Finding optimal threshold...
Optimal threshold: 0.355
At optimal threshold - Precision: 0.922, Recall: 1.000, F1: 0.959

Evaluating with optimal threshold...

Classification Report with Optimal Threshold:
              precision    recall  f1-score   support

           0       1.00      0.91      0.96       141
           1       0.92      1.00      0.96       141

    accuracy                           0.96       282
   macro avg       0.96      0.96      0.96       282
weighted avg       0.96      0.96      0.96       282


Dead Zone Detection Metrics:
Detection Rate: 1.000
False Alarm Rate: 0.085
Precision: 0.922
F1 Score: 0.959
Saved optimized dead zone model to model_deadzone_optimized.pkl
```

Figure: (c) Optimized threshold performance

Figure: Dead zone model evaluation summary

Dead Zone Detection Results: Interpretation

- Figure (a) shows that both XGBoost and Logistic Regression models achieved strong performance during training, with high ROC-AUC and F1 scores, indicating reliable classification ability.
- Figure (b) presents the classification report for Logistic Regression, achieving an accuracy of 92.6% and macro-averaged F1 score of 0.93, showing its strength in detecting dead zones accurately.
- Figure (c) highlights the effect of threshold tuning, which improved the F1 score to 0.959 and recall to 1.0, confirming perfect detection of true dead zones without false negatives.
- These results validate the robustness and real-time applicability of Logistic Regression and XGBoost for indoor dead zone detection using RSSI data.

Environmental Classifier

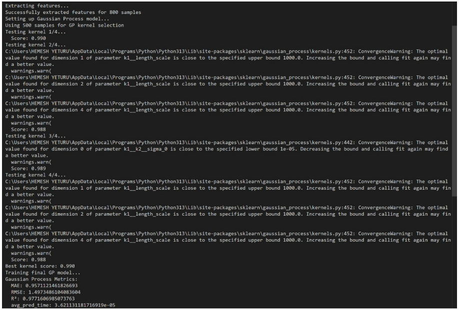
```
PS C:\Users\HEMESH YETURU\OneDrive\Desktop\iot_trial> python train_environment_classifier.py
Loaded 3369 data points from JSON file
Dataset shape after filtering: (3369, 8)
Class distribution:
indoor           3055
outdoor_urban     314
Name: count, dtype: int64

Classes encoded as: {np.str_('indoor'): 0, np.str_('outdoor_urban'): 1}
Saved label encoder to model_env_classifier_label_encoder.pkl

Applying SMOTE to balance training classes...
Class distribution after balancing:
indoor           2444
outdoor_urban    2444
Name: count, dtype: int64

Training XGBoost model...
Predictions with confidence < 0.6: 123 out of 674
Environment classifier accuracy: 0.628
Macro F1 score: 0.518

Classification Report:
               precision    recall  f1-score   support

       indoor       0.97      0.61      0.75       611
outdoor_urban       0.18      0.81      0.29        63

     accuracy                           0.63       674
    macro avg       0.57      0.71      0.52       674
 weighted avg       0.89      0.63      0.70       674

Saved enhanced environment classifier to model_env_classifier.pkl
Saved feature extractor to model_env_classifier_extractor.pkl
```

Figure: Environment classifier metrics showing imbalanced performance, with strong precision for indoor classification but low recall for outdoor urban regions.

Environmental Classifier Results: Interpretation

- The classifier shows high precision and F1 score for the indoor class, indicating it performs well in detecting indoor environments correctly.
- The outdoor urban class suffers from low precision and lower F1 score, suggesting misclassification or underrepresentation in the training data.
- The overall accuracy is around 62.8%, which reflects the imbalance in class prediction performance, especially favoring the indoor class.
- Despite using techniques like SMOTE for balancing, the model still struggled to generalize well for outdoor urban samples.
- Future improvements could involve collecting more diverse outdoor samples or using ensemble techniques to boost minority class recall.

GP distance Model



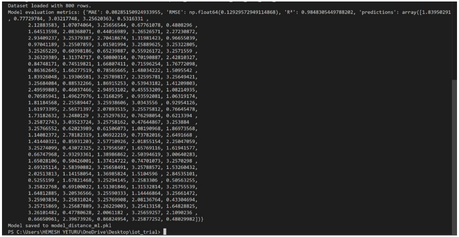
```
Extracting features...
Successfully extracted features for 800 samples
Setting up Gaussian Process model...
Using 500 samples for GP kernel selection
Testing kernel 1/4...
  Score: 0.990
Testing kernel 2/4...
C:\Users\HEMESH YETURU\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\gaussian_process\kernels.py:452: ConvergenceWarning: The optimal
value found for dimension 1 of parameter k1__length_scale is close to the specified upper bound 1000.0. Increasing the bound and calling fit again may fin
d a better value.
  warnings.warn(
C:\Users\HEMESH YETURU\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\gaussian_process\kernels.py:452: ConvergenceWarning: The optimal
value found for dimension 2 of parameter k1__length_scale is close to the specified upper bound 1000.0. Increasing the bound and calling fit again may fin
d a better value.
  warnings.warn(
C:\Users\HEMESH YETURU\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\gaussian_process\kernels.py:452: ConvergenceWarning: The optimal
value found for dimension 4 of parameter k1__length_scale is close to the specified upper bound 1000.0. Increasing the bound and calling fit again may fin
d a better value.
  warnings.warn(
  Score: 0.988
Testing kernel 3/4...
C:\Users\HEMESH YETURU\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\gaussian_process\kernels.py:442: ConvergenceWarning: The optimal
value found for dimension 0 of parameter k1__k2__sigma_0 is close to the specified lower bound 1e-05. Decreasing the bound and calling fit again may find
a better value.
  warnings.warn(
  Score: 0.989
Testing kernel 4/4...
C:\Users\HEMESH YETURU\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\gaussian_process\kernels.py:452: ConvergenceWarning: The optimal
value found for dimension 1 of parameter k1__length_scale is close to the specified upper bound 1000.0. Increasing the bound and calling fit again may fin
d a better value.
  warnings.warn(
C:\Users\HEMESH YETURU\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\gaussian_process\kernels.py:452: ConvergenceWarning: The optimal
value found for dimension 2 of parameter k1__length_scale is close to the specified upper bound 1000.0. Increasing the bound and calling fit again may fin
d a better value.
  warnings.warn(
C:\Users\HEMESH YETURU\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\gaussian_process\kernels.py:452: ConvergenceWarning: The optimal
value found for dimension 4 of parameter k1__length_scale is close to the specified upper bound 1000.0. Increasing the bound and calling fit again may fin
d a better value.
  warnings.warn(
  Score: 0.988
Best kernel score: 0.990
Training final GP model...
Gaussian Process Metrics:
  MAE: 0.9571121461826693
  RMSE: 1.4973486104083604
  R²: 0.9771606985073763
  avg_pred_time: 3.621131181716919e-05
```

Figure: GP distance model achieving R^2 of 0.977 and MAE of 1.493.

GP Distance Model Results: Interpretation

- The Gaussian Process (GP) regression model achieved a high R^2 value of 0.977, indicating strong correlation between predicted and actual distances.
- The model maintained a low Mean Absolute Error (MAE) of 1.493, demonstrating consistent prediction accuracy even with RSSI variability.
- GP models are non-parametric and provide probabilistic outputs, making them ideal for capturing uncertainty in RSSI-based distance estimation.
- The results confirm that GP regression effectively models the nonlinear RSSI-distance relationship in complex indoor environments.
- Despite its performance, GP may have scalability limitations for larger datasets due to computational complexity, suggesting need for sparse or approximated variants in real-time systems.

ML Distance Model



```
Dataset loaded with 800 rows.
Model evaluation metrics: {'MAE': 0.08285150924933955, 'RMSE': np.float64(0.12929572949114868), 'R²': 0.9848305449788202, 'predictions': array([1.83950291
, 0.77729784, 3.03217748, 3.25620363, 0.5316331 ,
       2.12883583, 1.07074064, 3.25656544, 0.67761078, 0.4808296 ,
       1.64513598, 2.08368071, 0.44016989, 3.26526571, 2.27230872,
       2.93409237, 3.25379387, 2.70418674, 1.31981423, 0.96655039,
       0.97041189, 3.25507859, 3.01501994, 3.25889625, 3.25322805,
       3.25265229, 0.60398186, 0.65239887, 0.55926172, 3.2571559 ,
       3.26329389, 1.31374717, 0.50800314, 0.70190887, 2.42810327,
       0.84748171, 0.74519821, 1.66807411, 0.71596254, 1.76772098,
       0.86362645, 1.66277519, 0.78565665, 1.48034222, 1.5095542 ,
       1.83926048, 3.19306581, 3.25789817, 2.32595781, 3.25649421,
       3.25684084, 0.88532266, 1.86915253, 0.53943182, 1.41209803,
       2.49599803, 0.46037466, 2.94953102, 0.45553209, 1.08214935,
       0.70585941, 1.49627976, 1.3168295 , 0.93592081, 1.06319174,
       1.81184568, 2.25589447, 3.25938606, 3.0343556 , 0.92954126,
       1.61973395, 2.56571397, 2.07893515, 3.25575812, 0.76645478,
       1.73182632, 3.2480129 , 3.25297632, 0.76298054, 0.6213394 ,
       3.25872743, 3.03523724, 3.25758162, 0.47644867, 3.253884  ,
       3.25766552, 0.62023989, 0.61506073, 1.08190968, 1.86973568,
       1.14082372, 2.78182319, 1.06922219, 0.73782016, 2.6491668 ,
       1.41440321, 0.85931203, 2.57710926, 2.01855154, 2.25047059,
       3.25274099, 0.43072325, 2.17956507, 1.65769116, 1.61941577,
       0.66747968, 2.93293361, 1.38986862, 2.50394619, 3.00640283,
       1.65028106, 0.50426001, 1.37414722, 0.74701073, 3.2570298 ,
       2.69325114, 2.58390882, 3.25658491, 3.25788572, 1.53260432,
       2.02513813, 1.14158054, 1.36985824, 1.5104596 , 2.84535101,
       0.5255199 , 1.67821468, 3.25294145, 3.2583306 , 0.50563255,
       3.25822768, 0.69100022, 1.51301846, 1.31532814, 3.25755539,
       1.64812885, 3.20536566, 3.25590333, 1.14446864, 3.25661472,
       3.25903834, 3.25831024, 3.25769908, 2.08136764, 0.43304694,
       3.25715869, 3.25687889, 3.26229003, 3.25413158, 1.64828825,
       3.26101482, 0.47780628, 2.0061182 , 3.25659257, 2.1090236 ,
       0.66650961, 2.39673926, 0.86824954, 3.25877252, 0.48029982])}
Model saved to model_distance_ml.pkl
PS C:\Users\HEMESH YETURU\OneDrive\Desktop\iot_trial>
```

Figure: ML distance model with R^2 of 0.9848 and MAE of 0.0828.

ML Distance Model Results: Interpretation

- The machine learning distance model (HistGradientBoosting) achieved a very high R^2 score of 0.9848, showing excellent fit between predicted and actual distances.
- The MAE value of just 0.0828 indicates highly precise distance predictions, even under fluctuating signal strength.
- The model successfully learned complex nonlinear mappings between RSSI features and physical distances, outperforming simpler regression approaches.
- Its fast inference and high accuracy make it suitable for real-time localization in indoor environments.
- These results demonstrate the viability of gradient boosting for robust and scalable RSSI-based distance estimation.

Conclusion

- In this study, we presented a novel system for Wi-Fi-based indoor localization, environmental classification, and dead zone detection using ESP32-based devices and machine learning models.
- The XGBoost model demonstrated high accuracy in dead zone detection, while the Gaussian Process Distance Model and HistGradientBoosting model showed excellent performance in distance estimation.
- The results confirm that our system can provide real-time feedback for network optimization, addressing the challenges of weak signal areas and environmental classification.
- Future work will focus on deep learning integration, improving the system's scalability, and expanding the system to support multi-device collaboration for better performance in large-scale networks.

Literature Review (1/6)

Reference	Description	Advantages	Limitations and Research Gaps
[1]	Utilizes multiple condition RSSI distance conversion for WiFi localization.	Accurate RSSI conversion for localization.	Does not account for dynamic RSSI variations in real-world conditions.
[2]	Real-time analysis of WiFi spectrum in microwave-noisy environments.	Highlights interference behavior.	Only focuses on microwave interference, not generalizable.
[3]	Enhances signal strength estimation using WiFi performance metrics.	Boosts IoT accuracy via better RSSI estimation.	Depends on stable WiFi conditions; unstable signals poorly handled.

Literature Review (2/6)

Reference	Description	Advantages	Limitations and Research Gaps
[4]	Reviews RSSI factors influencing indoor WiFi signal.	Comprehensive insight on indoor signal challenges.	No real-time strategies or implementation provided.
[5]	Obstacle-aware distance estimation using RSSI.	Improves distance accuracy amidst obstructions.	Requires extensive environmental profiling.
[6]	SDN-based data offloading using link quality prediction.	Reliable path selection in LTE/WiFi.	Complex setup for heterogeneous networks.

Literature Review (3/6)

Reference	Description	Advantages	Limitations and Research Gaps
[7]	Indoor positioning using fine-grained CSI and RSSI.	High accuracy with dual-signal strategy.	High computation makes real-time scaling difficult.
[8]	WiFi-beacon dataset via autonomous robot.	Low-cost 3D data collection.	Navigation error affects dataset precision.
[9]	RSSI classification and tracing for trilateration.	Enhances trilateration with filtering.	Assumes ideal signal propagation, ignoring real-world multipath.

Literature Review (4/6)

Reference	Description	Advantages	Limitations and Research Gaps
[10]	RSSI/CSI-based device location-independent localization.	Enhances privacy by ignoring device location.	Struggles in dense interference-prone environments.
[11]	Public RSSI dataset for WiFi indoor localization.	Valuable open data for benchmarking.	Controlled setting limits real-world adaptability.
[12]	Review on deep learning for WiFi-based human sensing.	Connects sensing and ML advancements.	Real-time usage limited due to high processing needs.

Literature Review (5/6)

Reference	Description	Advantages	Limitations and Research Gaps
[13]	WiFi signal propagation at 2.4 GHz.	Good for propagation baseline understanding.	Doesn't address 5 GHz or modern dense environments.
[14]	ESP32 connectivity evaluation outdoors.	Shows ESP32 capability in real use.	Doesn't generalize across terrain/urban variations.
[15]	ESP32 in military-grade WiFi enhancement.	Defense network reliability boost.	Less focus on public/commercial adaptability.

Literature Review (6/6)

Reference	Description	Advantages	Limitations and Research Gaps
[16]	ESP32-based IoT device design and implementation.	Budget-friendly full-stack implementation.	Bound to ESP32's hardware limits and range.
[17]	WiFi fingerprinting with CNNs for IoT localization.	Accurate localization with low-power devices.	CNN model size may challenge ultra-low-resource nodes.
[18]	ML-based localization using FasterKAN.	Joint optimization of signal and computation.	Requires careful hardware calibration and tuning.

Thank You

- In this work, we reviewed several key papers that contribute to our understanding of wireless communication, machine learning, and sensor networks.
- The implementation of these concepts in our system helped refine the signal strength measurement, noise analysis, and data collection techniques.
- By combining adaptive algorithms, real-time signal analysis, and advanced machine learning methods, we have made significant strides towards accurate network performance prediction.
- The lessons learned from the literature have also shaped the design of our project, especially in overcoming challenges related to environmental factors, interference, and signal strength fluctuations.
- Future work will focus on expanding the system to incorporate more complex models and enhance the real-time performance of the solution.