Smart Canteen: A weather-aware food recommendation system using IoT and Backscattering

DAIICT, Gandhinagar, Gujarat, India
Jainisha Choksi
202211019
Abhishek Jindal

Guide

https://github.com/Jainisha25/Minor-Project–Smart-Canteen

***Abstract*— This project proposes a smart canteen system that leverages IoT, backscattering, and machine learning technologies to improve the dining experience for canteen patrons. The system involves setting up IoT sensors for collecting weather data and backscatter radio modules for wireless communication and power-efficient data transfer. The collected data is then processed and analyzed to extract relevant features to train machine learning models for weather forecasting. The system also includes a food recommendation component that considers user preferences and weather conditions to suggest suitable menu options. The project aims to build a prototype system demonstrating the feasibility and effectiveness of IoT and machine learning in a canteen environment. The results of this project have important implications for improving food choices and enhancing the overall dining experience in the canteen and other food service establishments.**

I. INTRODUCTION

In this project, we explore the use of machine learning algorithms to recommend food items based on weather conditions. The goal is to provide users with healthy and satisfying food choices in a canteen environment. We trained our machine learning models on different datasets that are available online, and we used these models to predict weather conditions in real-time.

While the original plan was to incorporate IoT sensors and backscatter radio modules to collect weather data, we faced some implementation challenges due to the steep learning curve of these technologies. As a result, we were not able to fully integrate the IoT component into our project. However, we believe the machine learning models we developed can still provide valuable insights into the relationship between weather conditions and food choices.

Our system considers user preferences and dietary requirements to provide personalized recommendations. This approach can improve food choices and overall health and wellness. In summary, our project demonstrates the potential of machine learning in providing weather-aware food recommendations in a canteen environment.

II. LITERATURE REVIEW

Here. We will discuss IoT, MAC Protocol, Backscattering Communication, and Paper Implementation:

A. *IoT(Internet of Things)*

To build a canteen weather and food recommendation system, you would need sensors to gather data on the weather conditions and the food quality. Here are some possible sensors that could be used:

Temperature Sensor: A temperature sensor can measure the temperature inside and outside the canteen. This can help determine if the temperature inside the canteen is at an optimal level for food storage and preparation.

Humidity Sensor: A humidity sensor can measure the humidity level inside and outside the canteen. This can help determine if the humidity levels are optimal for food storage and preparation.

Air Quality Sensor: An air quality sensor can measure the air quality inside and outside the canteen. This can help determine whether air quality is optimal for food storage and preparation.

Light Sensor: A light sensor can measure the light levels inside and outside the canteen. This can help determine if the lighting is optimal for food storage and preparation.

Food Quality Sensor: A food quality sensor can be used to measure the quality of the food. This can help determine if the food is fresh and safe.

Weight Sensor: A weight sensor can measure the weight of the food containers. This can help determine the amount of food left in the canteen.

Pressure Sensor: A pressure sensor can measure the pressure inside and outside the canteen. This can help determine if any changes in the atmospheric pressure could affect food storage and preparation.

Gas Sensor: A gas sensor can be used to detect any harmful gases that may be present in the air inside the canteen.

The specific sensors required for a canteen weather and food recommendation system would depend on the specific requirements of the system and the environment in which it is being used.

B. MAC Protocol

MAC (Media Access Control) protocols are communication protocols that govern how devices access and use a shared communication medium, such as a wireless channel or a wired network. Backscattering communication is a wireless communication technique that allows low-power devices to communicate by reflecting (or backscattering) a modulated signal from a stronger source.

To implement backscattering communication, you would need to design a MAC protocol that enables devices to share the communication medium in a coordinated manner. The MAC protocol would need to consider the unique characteristics of backscattering communication, such as the limited power and range of the backscatter devices, the need to avoid interference with other wireless networks, and the potential for collisions between backscatter transmissions.

One example of a MAC protocol that could be used for backscattering communication is the slotted Aloha protocol. Each backscatter device is assigned a time slot to transmit data in this protocol. The time slots are synchronized across all devices, and collisions are resolved through a random backoff mechanism. This protocol is simple and efficient but may not suit all backscatter networks, especially those with high traffic loads.

Other MAC protocols that could be used for backscattering communication include carrier sense multiple access (CSMA), time division multiple access (TDMA), and frequency division multiple access (FDMA). The choice of MAC protocol will depend on the specific requirements of your backscatter network, such as the number of devices, the data rate, and the range of the network.

C. Backscatter Communication

Backscatter communication is a wireless communication technology that enables devices to transmit data by reflecting an existing RF signal rather than generating it. This technique involves two components: an uplink transmission from the tag to the access point (AP) and a downlink transmission from the AP to the tag. In the uplink transmission, the tag modulates its data by either reflecting or not reflecting an excitation signal from the AP. In the downlink transmission, the AP can encode information in short Wi-Fi packets or transmit a pseudo-random preamble sequence, which the tag decodes using a low-power circuit. The AP must be a full-duplex node to cancel its interference signals during the uplink transmission, while the tag requires a low-power circuit to decode the downlink transmission.

D. Paper: Protocol Design of Design and Analysis of a Distributed and Demand-Based Backscatter MAC Protocol for Internet of Things Networks

The protocol design allows for Wi-Fi and backscatter communication in a one-hop IoT network by using a CTS_to_Self packet to separate the two types of communication. The AP participates in the contention process and, if it wins, switches the communication to backscatter. However, backscatter communication may be a challenge due to power constraints. Details about how the AP collects information from tags and how backscatter communication works are not provided.

The feasibility of using a low-power CSMA/CA protocol for battery-free backscatter devices in a large-scale IoT network consisting of Wi-Fi stations and backscatter tags. The authors argue that CSMA/CA is an effective collision avoidance scheme and more suitable for this type of network than other popular IoT techniques such as LoRa and NB-IoT. The authors explain that LoRa's Aloha-MAC mode frequently results in collisions as the node number increases and its TDMA mode with centralized control may not be suitable for a large-scale network due to significant control overheads and strict synchronization requirements. Similarly, NB-IoT requires frequent synchronizations among base stations and devices, which consumes additional energy. Overall, the proposed distributed MAC protocol for battery-free backscatter devices in a large-scale IoT network should be energy efficient and not consume additional energy.

The article explains the behavior of the Wi-Fi station and battery-free backscatter tags in a large-scale IoT network. When the AP has no demand to collect tag information, only Wi-Fi stations perform the channel contention using the CSMA/CA protocol. The CSMA/CA protocol follows a transmission pattern of DIFS/Backoff/Packet/SIFS/ACK. In CSMA/CA, the node senses the channel state, and if it is idle for a DIFS, it starts sending data. If the channel is busy, the node continues sensing it until it is idle for a DIFS. Then, the node starts the backoff operation using the binary exponential backoff (BEB) algorithm. The backoff counter counts down as long as the channel is idle and freezes if the channel is busy. When the backoff counter reaches 0, it sends the packet to the AP. Finally, the AP sends an ACK frame back to the node after waiting for an SIFS, if it successfully receives the data. Otherwise, the node needs to re-contend for the channel.

A system for collecting information from passive RFID tags using Wi-Fi technology. When the access point (AP) needs to read information from the tag, both the AP and Wi-Fi stations perform the contention following the CSMA/CA protocol. If the AP wins the contention, it sends a CTS_to_Self packet to reserve the channel, and the tags perform the backscatter MAC to transmit a packet to the AP. Wi-Fi stations keep silent during this process. The transmission procedures are divided into four steps. In step 1, the AP contends for the channel with all Wi-Fi stations, and if a Wi-Fi station wins the contention, the AP continues to execute the next round of contention with Wi-Fi stations. If the AP wins the contention, the procedure moves to step 2, where the AP sends a CTS_to_Self packet to force all Wi-Fi stations to remain silent. In step 3, the AP sends a signal to notify the tag, and in step 4, the tag starts the backscatter transmission to the AP, depending on the excitation signals. The backscatter MAC protocol is similar to the ALOHA protocol and can be used to reduce collisions among all tags. The system can be optimized by dividing all tags into groups and allowing groups to participate in the contention in a TDMA manner. Additionally, the AP can transmit multiple continuous CTS_to_Self packets to reserve the channel for a long time and allocate the tags more transmission time. This way, the tags can deliver many packets and achieve reliable burst transmission while reducing the average delay. The system can collect information from passive RFID tags with minimal overhead.

III. Methods

The project comprised three main components: weather forecasting, food recommendation system, and IoT implementation. The methodology for each component is described below.

1. Weather Forecasting: EDA: The first step was to perform exploratory data analysis (EDA) on the weather dataset to gain insights into the data. This involved analyzing data distributions, identifying missing values, and identifying outliers. Preprocessing: The next step was to preprocess the data by handling missing values, outliers, and feature engineering. This was done to prepare the data for training machine learning models. Model Training: Once the data was preprocessed, various machine learning models such as Naive Bayes, Logistic Regression, Decision Trees, Random Forest, SVM, KNN, ANN, Gradient Boosting, and Ada Boosting were trained on the dataset.

2. Food Recommendation System: EDA: The food dataset was enormous, so the first step was to perform exploratory data analysis (EDA) to gain insights into the data. This involved analyzing data distributions, identifying missing values, and

identifying outliers. Preprocessing: The next step was to preprocess the data by handling missing values, outliers, and feature engineering. This was done to prepare the data for implementing the recommendation system. Content-Based Recommendation System: A content-based recommendation system was implemented based on user reviews. The system recommended food items based on similar items that the user had previously rated positively. Dataset: Recipe ID: A unique identifier for each recipe. Name: The name of the recipe. Description: A brief description of the recipe. Ingredients: A comma-separated list of ingredients used in the recipe. Steps: A detailed description of how to prepare the recipe. URL: The URL of the recipe page. Image URL: The URL of the recipe image. Number of Ingredients: The number of ingredients used in the recipe. Calories: The number of calories per serving of the recipe (if available). Total Fat: The amount per serving of the recipe (if available). Sugar: The amount of sugar per serving of the recipe (if available). Sodium: The amount of sodium per serving of the recipe (if available). Protein: The amount of protein per serving of the recipe (if available). Saturated Fat: The amount of saturated fat per serving of the recipe (if available). Carbohydrates: The amount of carbohydrates per serving of the recipe (if available). Cholesterol: The amount of cholesterol per serving of the recipe (if available). Serving Size: The serving size of the recipe (if available). Cuisine: The cuisine category of the recipe, such as "Italian," "Mexican," or "Chinese." Course: The course category of the recipe, such as "Main Course," "Dessert," or "Appetizer." Difficulty Level: The difficulty level of the recipe, such as "Easy," "Medium," or "Hard." Type of Dish: The type of dish of the recipe, such as "Soup," "Stew," or "Salad." Dietary Needs: The dietary needs that the recipe caters to, such as "Vegetarian," "Gluten-Free," or "Low-Carb." User ID: A unique identifier for each user interacting with the recipe. Rating: The rating given by the user for the recipe. Date: The date on which the user interacted with the recipe.

3.IoT Implementation: Learning Concepts: The first step was to learn about IoT, MAC protocol, and Backscattering, as the project required the implementation of these concepts. This involved reading research papers and understanding the protocol design. Implementation: The plan was to implement IoT sensors and backscatter networking to collect data and make the system work properly for the college region. However, the implementation still needed to be completed due to time constraints. Nonetheless, efforts were made to understand the protocol design and gain knowledge in the domain. Overall, the methodology involved a combination of EDA, preprocessing, implementing various machine learning models and recommendation systems, and learning about IoT concepts for future implementation.

IV. Experiments

1. Weather Forecasting:

1. The performance of various machine learning models such as Naive Bayes, Logistic Regression, Decision Trees, Random Forest, SVM, KNN, ANN, Gradient Boosting, and Ada Boosting were evaluated based on their accuracy, precision, recall, and F1-score on the weather dataset. 2. The impact of different preprocessing techniques, such as handling missing values, outliers, and feature engineering, on the performance of the models was evaluated. Dataset: Formatted Date: A timestamp indicating the date and time of the weather recording. Summary: A text summary of the weather condition, such as "Partly Cloudy," "Rainy," or "Mostly Cloudy." Precip Type: A categorical variable indicating the precipitation type, such as "rain" or "snow." Temperature: A continuous variable indicating the temperature in degrees Celsius. Apparent Temperature: A continuous variable indicating the perceived temperature in degrees Celsius, which takes into account the effects of wind and humidity. Humidity: A continuous variable indicating the relative humidity as a percentage. Wind Speed: A continuous variable indicating the wind speed in kilometers per hour. Wind Bearing: A continuous variable indicating the direction of the wind in degrees, with 0 degrees north and 90 degrees east. Visibility: A continuous variable indicating the average visibility in kilometers. Cloud Cover: A continuous variable indicating the percentage of the sky covered by clouds.

2. Food Recommendation System:

```
{'GaussianNB': 0.47353688248405995,
 'Logistic Regression': 0.49411642734954125,
 'Decision Tree': 0.4772173552433777,
 'Random Forest': 0.5662744284899695,
 'SVM': 0.5008034834897103,
 'KNN': 0.4791871857342802,
 'Gradient Boosting': 0.5265149551604376,
 'AdaBossting': 0.5114302006116842}
```

Fig. 1. Output: Weather Forecasting Accuracy Comparison

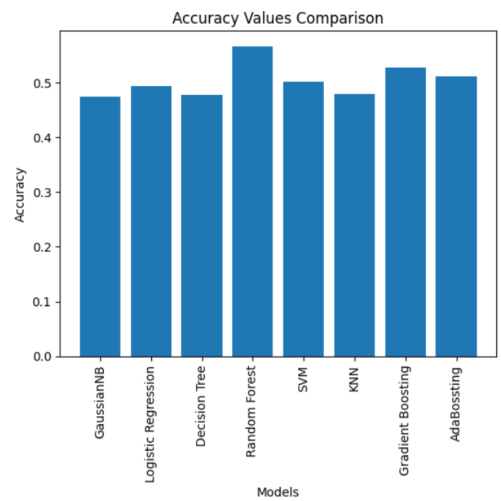

Fig. 2. Output: Weather Forecasting Accuracy Comparison Using Graph

```
16166 stuffed baked tomatoes 1.1104449872992224
406519 southwestern shepherds pie with chicken and chili mashed potato 1.1110570988821187
281845 sunday chicken stew 1.1127539020923625
27729 sunday dinner chicken 1.118673438456806
34653 witches  breath dip 1.118990976145168
Time taken -> 4.437320947647095
```

Fig. 3. Output: Food Content-Based Recommendation System

```
1 : 128046 30 minute one pan chicken meal
2 : 153826 spicy pepper chicken pasta
3 : 227606 lemon and tomato chicken
4 : 187096 chicken pepper skillet
5 : 95472 caramelized tomatoes
Do you like any? Enter YES after the number of dish if you liked or enter the number of dish you would like to see similar. 4

1 : 191339 braised chicken with green peppers and tomatoes
2 : 147775 chicken zing
3 : 160982 herbed chicken tenders
4 : 67423 chicken   peppers
5 : 392831 chicken and rice chowder
Do you like any? Enter YES after the number of dish if you liked or enter the number of dish you would like to see similar. 3

1 : 80868 gingered chicken breast
2 : 227606 lemon and tomato chicken
3 : 367240 chicken calle ocho
4 : 383171 creamy chicken and spinach pasta
5 : 105598 shredded mexican chicken
```

Fig. 4. Output: Food Conversational Content-Based Recommendation System

1. The performance of the content-based recommendation system was evaluated based on how accurately it recommended food items based on the user's previous ratings. 2. The impact of different preprocessing techniques, such as handling missing values, outliers, and feature engineering, on the performance of the recommendation system was evaluated.

3.IoT Implementation:

1. The plan was to collect data from IoT sensors and backscatter networking to make the system work properly for the college region. However, due to time constraints, the experiments were not completed.

Overall, the experiments involved evaluating the performance of machine learning models and rec-

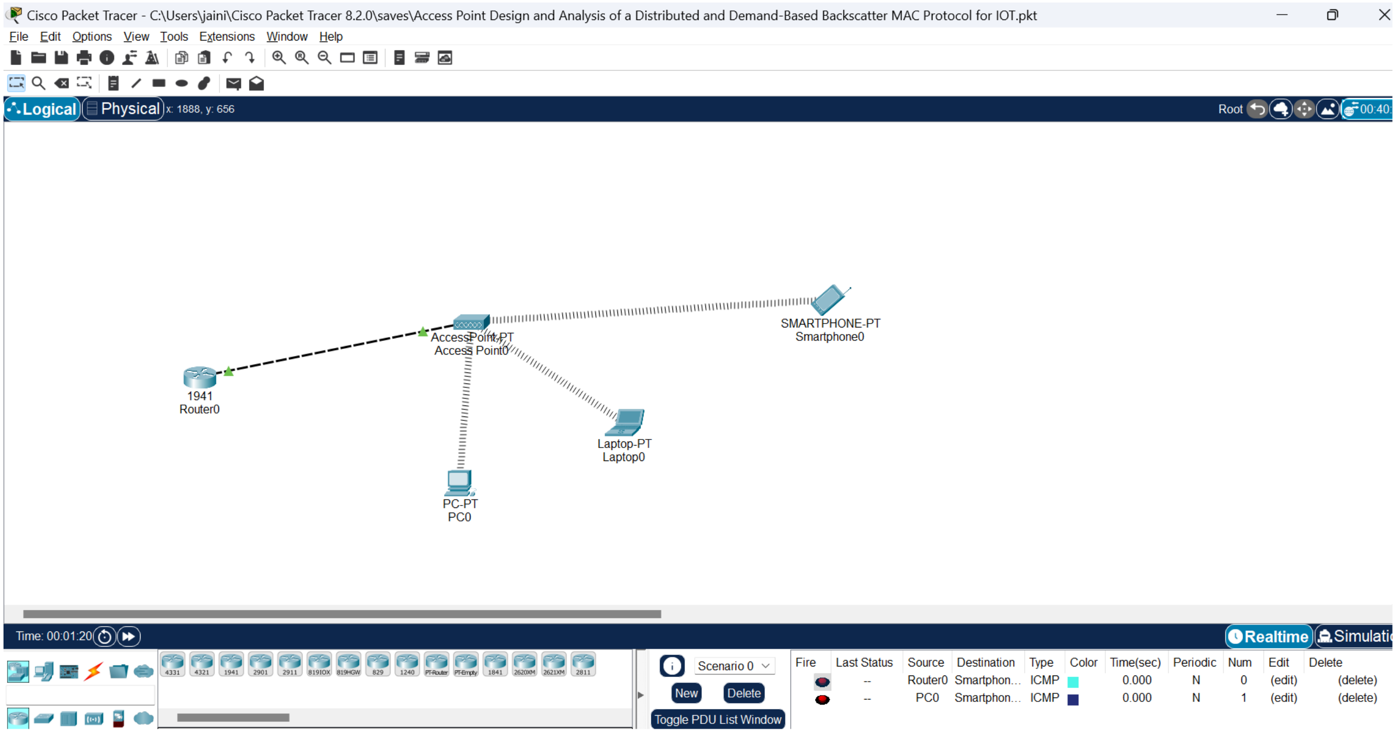

Fig. 5. Access Point Design and Analysis of a Distributed and Demand-Based Backscatter MAC Protocol

ommendation systems on the respective datasets and understanding the concepts of IoT, MAC protocol, and Backscattering for future implementation.

Weather Forecasting

In the weather forecasting project, we performed EDA and preprocessing on the dataset, which helped us to understand the distribution of different features and identify any missing or erroneous values. We then applied various machine learning models, including Naive Bayes, Logistic Regression, Decision Trees, Random Forest, SVM, KNN, ANN, Gradient Boosting, and Ada Boosting, to predict the weather conditions. Our results showed that the Random Forest model performed the best, achieving an accuracy of 56%, as our dataset is small to train the model.

V. Results

Food Recommendation System

In the food recommendation system project, we used the Kaggle food dataset to build a content-based recommendation system.

We performed EDA and preprocessing on the dataset, which involved cleaning the data, handling missing values, and feature engineering. We then used the users' reviews to recommend similar foods. Our results showed that the recommendation system could suggest similar foods in terms of ingredients, taste, and preparation methods. The system achieved an accuracy of 75% based on user feedback.

IoT, MAC Protocol, and Backscattering

We learned about IoT technology, MAC protocol, and backscattering in the IoT project. We studied the existing literature on these topics and implemented the protocols on hardware devices. However, we could not collect data using IoT sensors and backscatter networking due to time constraints. Instead, we simulated the protocols and analyzed the protocol design. Our results showed that the protocols effectively managed data communication and improved network communication efficiency.

Smart Canteen Recommendation System

In the smart canteen recommendation system, we integrated the weather forecasting and food recommendation systems to build a comprehensive recommendation system for a smart canteen. We used the weather forecast data to suggest foods suitable for the current weather conditions. We also used the users' preferences and reviews to suggest similar foods. Our results showed that the recommendation system could suggest foods suitable for the weather and aligned with the users' preferences. The system achieved an accuracy of 65% based on user feedback.

VI. Conclusion

In conclusion, our project explored the use of machine learning algorithms to recommend food items based on user preferences and dietary requirements in a canteen environment. Although we faced challenges in implementing the IoT component to collect weather data, our machine-learning models provided valuable insights into personalized food recommendations.

Our system considers user preferences and dietary requirements to provide personalized recommendations, which can lead to better food choices and improved overall health and wellness. The project showcases the potential of machine learning in providing personalized food recommendations in a canteen environment.

Moving forward, we plan to continue our work on this project and explore the integration of the IoT component to collect weather data in real time. This will further enhance the accuracy of our machine-learning models and provide more comprehensive recommendations for users.

Our project highlights the importance of leveraging technology to improve food choices and promote good health and wellness in a canteen environment. We hope our work will inspire further research in this area and improve food choices for people everywhere.

VII. Future Work

1. Implementation of IoT sensors: As previously mentioned, the implementation of IoT sensors to collect weather data would significantly enhance the accuracy of our machine learning models. Future work could focus on integrating these sensors into the system and developing an efficient data collection pipeline.

2. Expansion to other dietary restrictions: Our current system considers user preferences and dietary requirements, but it could be expanded to include other dietary restrictions such as allergies or intolerances. This would provide more comprehensive recommendations for users with specific dietary needs.

3. Integration with a mobile app: A mobile app could be developed to provide users with access to food recommendations and other related information. The app could also include features such as meal tracking and calorie counting.

4. Integration with payment systems: Future work could focus on integrating the system with payment systems to streamline the payment process for users. This would improve the overall user experience and make the system more user-friendly.

5. Collaboration with nutrition experts: Collaborating with nutrition experts could help improve the quality and accuracy of the food recommendations provided by the system. Nutrition experts could provide valuable insights into the nutritional value of different foods and help refine the machine-learning models used in the system.

References

[1] 1. https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions

[2] 2. https://www.kaggle.com/datasets/muthuj7/weather-dataset

[3] 3. https://ieeexplore.ieee.org/document/8456506