Department of Computer Science
Texas State University

Deep Learning Project Report

Bao Hoang Duc Vo
bao.vo@txstate.edu

Javad Mokhtari Koushyar
koushyar@txstate.edu

Abstract

This project utilizes a dataset to predict room occupancy using environmental observations such as temperature, humidity, and CO2 levels. Effective prediction of occupancy from such data can significantly enhance the efficiency of energy usage in buildings and improve automated system responses to human presence. We have implemented five different models to address this problem: RandomForest Classifier, Multilayer Perceptron, CNN, LSTM and Transformer.

1 Introduction

The use of sensors like thermostats to gather environmental data is common; however, these systems can often continue to operate based on previous settings even when a room is unoccupied. This scenario underscores the need for intelligent systems capable of recognizing occupancy patterns to optimize energy usage. This project aims to develop a predictive model that not only accurately determines room occupancy but also has the potential to contribute to more energy-efficient building management.

1.1 Data Description

Source: The dataset used in this project is sourced from the UCI Machine Learning Repository. The dataset comprises several environmental attributes collected from sensors, structured as follows:

Table 1: Description of dataset attributes

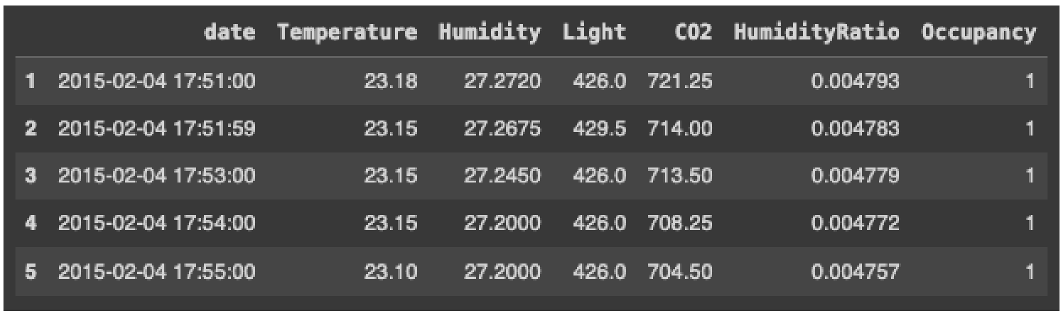
Attribute	Description
Date	Records the exact date and time of the data entry.
Temperature	Indicates the ambient temperature of the environment.
Relative Humidity	Shows the amount of moisture in the air.
Light	Represents the light level of the environment.
CO2	Can indicate human occupancy and affect the perceived air quality.
Humidity Ratio	This metric is useful for understanding the air's capacity to hold moisture.
Occupancy	This is the target variable for prediction.

1.2 Data Exploration

The training data snapshot reveals the following structure and attributes:

Table 2: Description of data exploration

Attribute	Exploration
Date	Recorded on February 4, 2015, with a one-minute interval between each entry.
Temperature	temperature readings are provided in degrees Celsius, around 23.15 to 23.18°C.
Relative Humidity	captured in percentage terms, displaying a slight variation around the 27.27 %.
Light	measured in Lux, remaining constant at 426.0 Lux.
CO2	Carbon dioxide concentrations with values in the vicinity of 714 to 721 ppm.
Humidity Ratio	The values here are in the range of approximately 0.0047.
Occupancy	Indicates the occupancy status of the room, where '1' denotes occupancy.



	date	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
1	2015-02-04 17:51:00	23.18	27.2720	426.0	721.25	0.004793	1
2	2015-02-04 17:51:59	23.15	27.2675	429.5	714.00	0.004783	1
3	2015-02-04 17:53:00	23.15	27.2450	426.0	713.50	0.004779	1
4	2015-02-04 17:54:00	23.15	27.2000	426.0	708.25	0.004772	1
5	2015-02-04 17:55:00	23.10	27.2000	426.0	704.50	0.004757	1

Figure 1: Data Frame Head

1.3 Exploratory Data Analysis

In the exploratory data analysis, we will examine the distribution and time series trends of the data, as well as analyze the correlations between variables.

The line graph displays room temperature changes from February 5th to February 10th, 2015. Sharp spikes in temperature suggest occupancy, likely due to body heat from individuals present in the room. Analyzing these variations aids in developing energy-efficient environmental control systems by predicting the presence or absence of occupants.

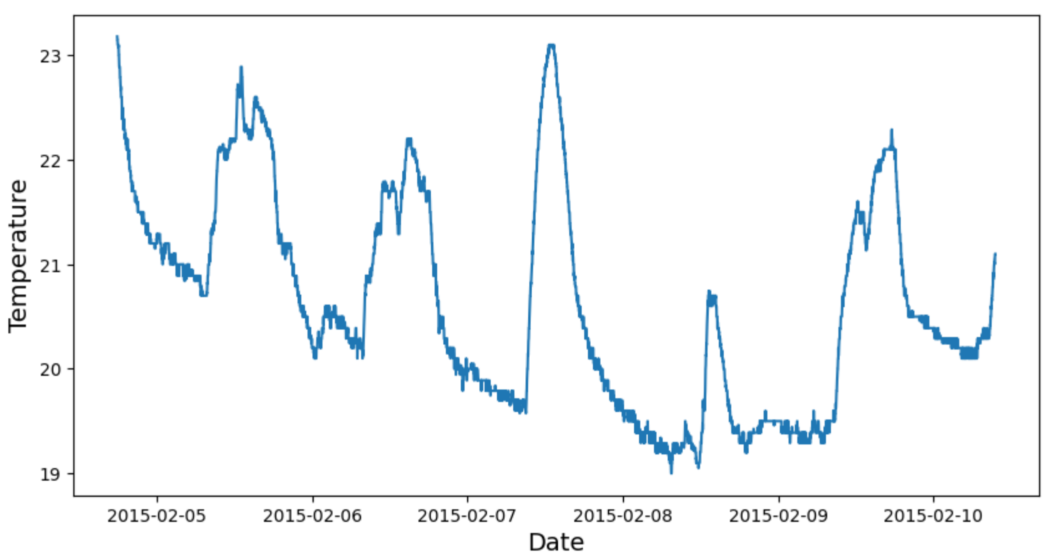


Figure 2: Temperature over Time

The line graph reflects light levels in a room from February 5th to 10th, 2015. Looking closely, we can see that the number of peaks in this graph and the temperature graph are the same. The persistent high light levels on February 10th may be due to factors beyond regular occupancy, necessitating further investigation into environmental or usage pattern changes. This is a good indicator of the occupancy of the room.

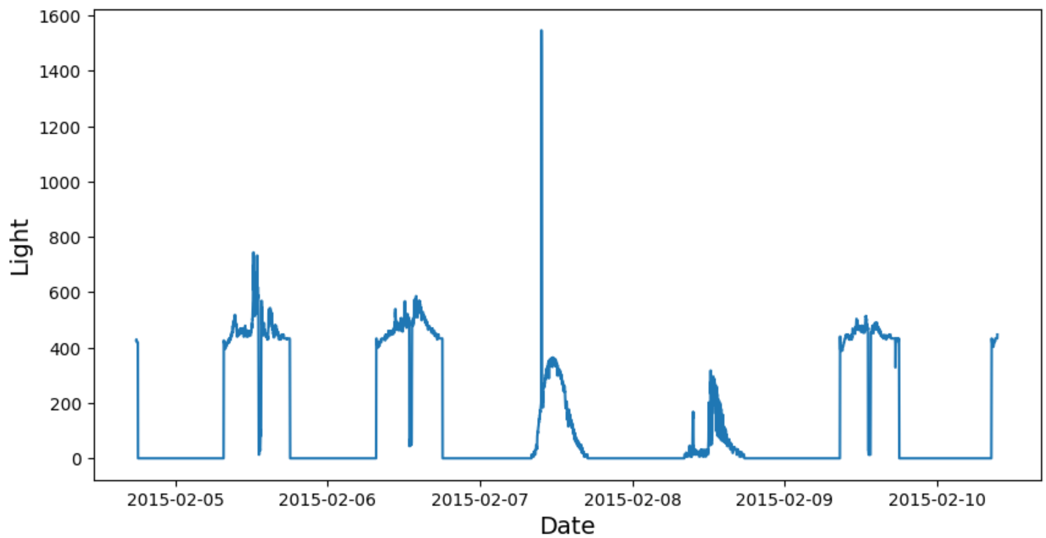


Figure 3: Light Intensity over Time

The line graph charts CO2 concentration levels from February 5th to February 10th, 2015. Pronounced spikes in CO2 levels could indicate occupancy, as human respiration is a primary source of CO2 in enclosed spaces. These spikes align with increases seen in the temperature and light graphs, reinforcing the likelihood of human presence. A notable dip in CO2 levels from February 7th to 9th suggests the room was likely unoccupied, a conclusion that seems to diverge from patterns observed in the humidity and temperature data. This discrepancy might imply varying activities or environmental conditions influencing each parameter differently, warranting a more nuanced analysis of occupancy patterns.

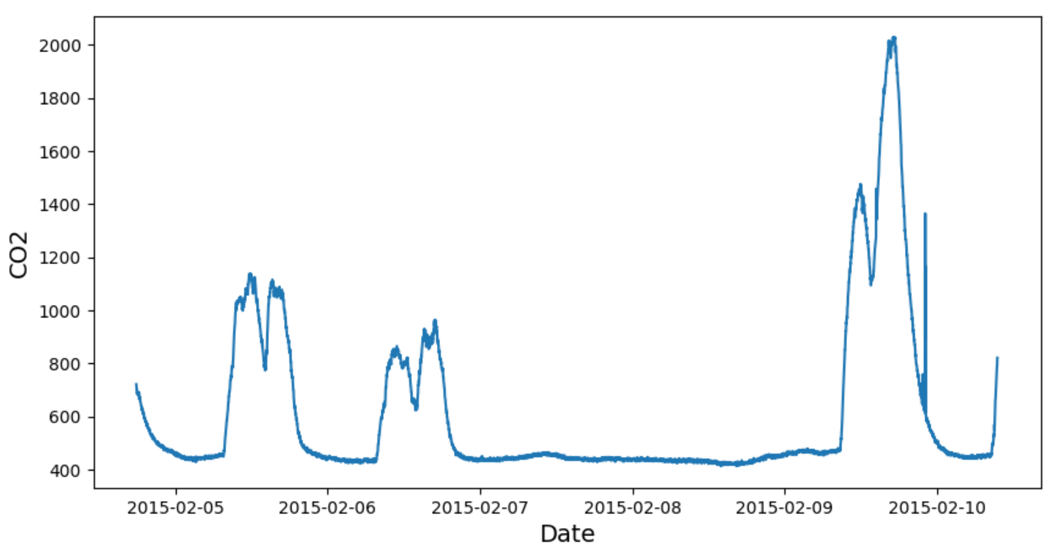


Figure 4: CO2 Levels over Time

1.3.1 Correlation Between the Variables

The heatmap visualizes the correlation matrix for various environmental factors and their relationship to room occupancy. It highlights a strong correlation between light intensity and occupancy, signifying that light is a significant indicator of whether a space is being used. Additionally, there is a robust correlation between CO2 levels and occupancy, aligning with the expectation that occupied spaces tend to have higher CO2 concentrations due to exhalation.

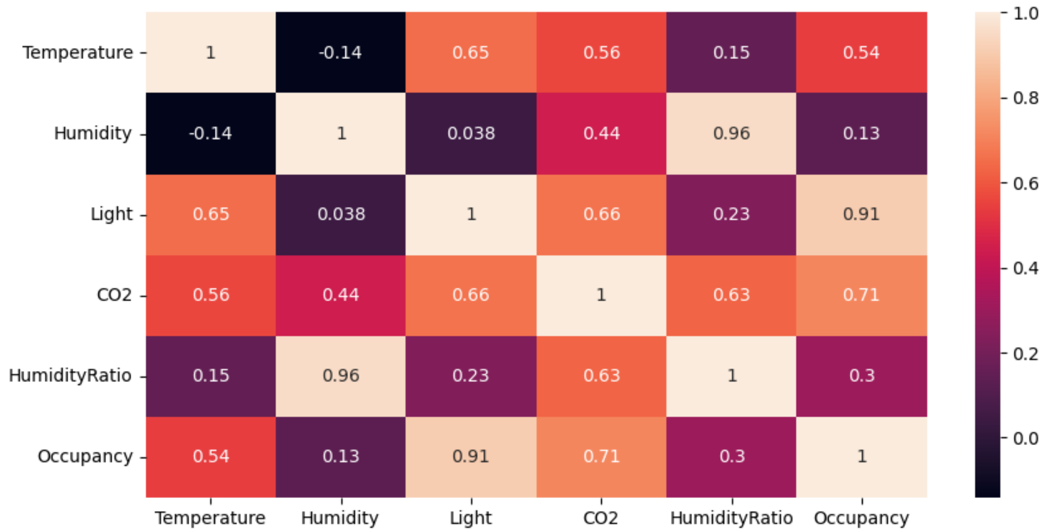


Figure 5: Correlation Heatmap

1.3.2 Distribution

The box plot illustrates the distribution of several environmental features. Temperature and humidity exhibit relatively tight distributions with a few outliers, indicating consistent measurements with some deviations. Light shows a wider range, with many outliers suggesting variable illumination conditions, possibly due to artificial lighting or daylight changes. CO2 levels have a compact box but a vast spread of outliers, indicating occasional spikes likely due to occupancy. The humidity ratio, while having a smaller interquartile range, shows consistency across the dataset with minimal outliers. This plot underscores the diversity in feature behavior, which could be leveraged for occupancy detection.

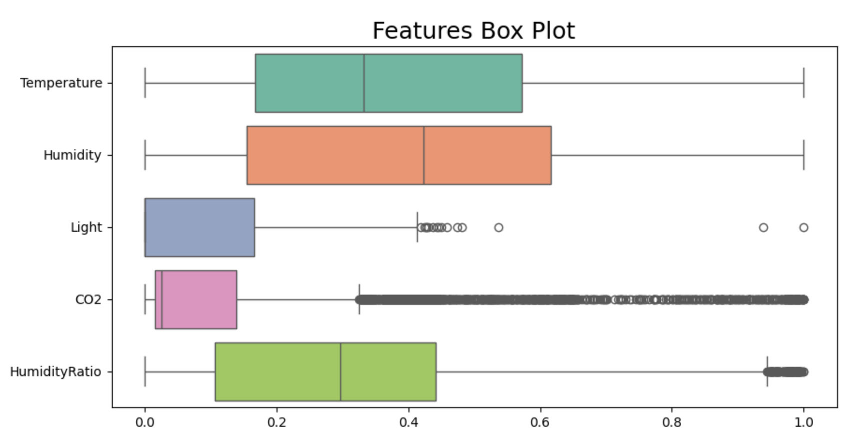


Figure 6: Box Plot Features

The bar chart presents the label distribution for the occupancy detection training dataset. It shows a significant imbalance between the two classes: non-occupied (label '0') and occupied (label '1'). The non-occupied class has a notably higher count, indicating that the data contains more instances where the room is unoccupied compared to when it is occupied. This imbalance in the dataset could affect the performance of machine learning models and may require techniques such as oversampling, undersampling, or weighted loss functions to ensure the model does not become biased towards the majority class.

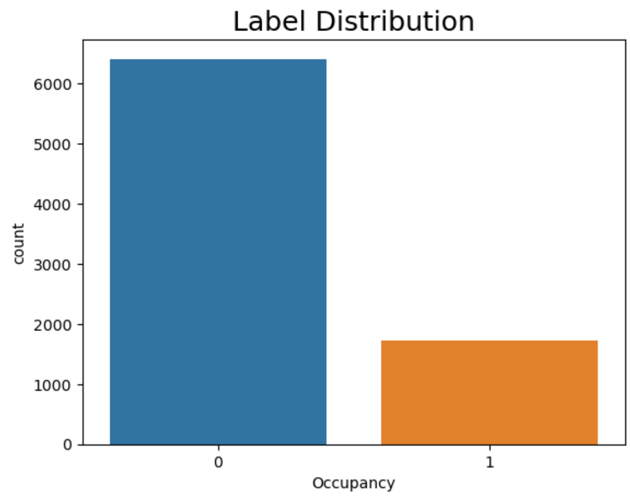


Figure 7: Training Label Distribution

2 Methods

2.1 Model 1: RandomForest Classifier

2.1.1 Architecture

The initial model we explore is the RandomForest Classifier. For our use case, the RandomForest Classifier was configured with 100 trees (n_estimators=100) and a random state of 42 to ensure reproducibility of the results.

2.1.2 Training Process

Training involved fitting the model on the feature set—temperature, humidity, light, CO2, and humidity ratio—to predict the binary occupancy status. The data was preprocessed to convert timestamps into a suitable format and normalize other environmental variables.

2.1.3 Model Evaluation

The model was evaluated on the validation set to assess its generalization capabilities. The primary metrics for evaluation were accuracy, precision, recall, and F1-score. Additionally, we computed a confusion matrix to provide a detailed breakdown of the model's performance.

Table 3: Validation scores for RandomForest Model

Metric	**Value**
Accuracy	0.99
Percision	0.99
Recall	0.99
F1 Score	0.99

Table 4: Test1 scores for RandomForest Model

Metric	Value
Accuracy	0.96
Percision	0.94
Recall	0.95
F1 Score	0.95

Table 5: Test2 scores for RandomForest Model

Metric	Value
Accuracy	0.96
Percision	0.87
Recall	0.98
F1 Score	0.92

2.2 Model 2: Multilayer Perceptron (MLP)

2.2.1 Architecture

The second model explored in this project is a Multilayer Perceptron, a type of feedforward artificial neural network. The MLP model comprises three linear layers, each followed by a ReLU activation function. Dropout layers with a dropout rate of 0.5 are inserted after the first and second ReLU activation functions to prevent overfitting. The final layer of the network employs a sigmoid activation function for the output layer to ensure the final output ranges between 0 and 1, which is suitable for binary classification tasks.

The MLP model has a total of 181 parameters, all of which are trainable. This model structure is intended to capture the non-linear interactions between the input features to predict room occupancy more accurately than linear models.

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
MLP                                      [1]                       --
├─Linear: 1-1                            [10]                      60
├─ReLU: 1-2                              [10]                      --
├─Dropout: 1-3                           [10]                      --
├─Linear: 1-4                            [10]                      110
├─ReLU: 1-5                              [10]                      --
├─Dropout: 1-6                           [10]                      --
├─Linear: 1-7                            [1]                       11
├─Sigmoid: 1-8                           [1]                       --
==========================================================================================
Total params: 181
Trainable params: 181
Non-trainable params: 0
Total mult-adds (M): 0.00
==========================================================================================
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.00
Estimated Total Size (MB): 0.00
==========================================================================================
```

Figure 8: MLP Model Summary

2.2.2 Training Process

The MLP was trained using the Adam optimizer with a learning rate of 0.001 and Binary Cross-Entropy as the loss function. The training process involved 100 epochs with early stopping implemented to prevent overfitting, using a patience parameter of 10 epochs, meaning training would cease if the validation loss did not improve for 10 consecutive epochs.

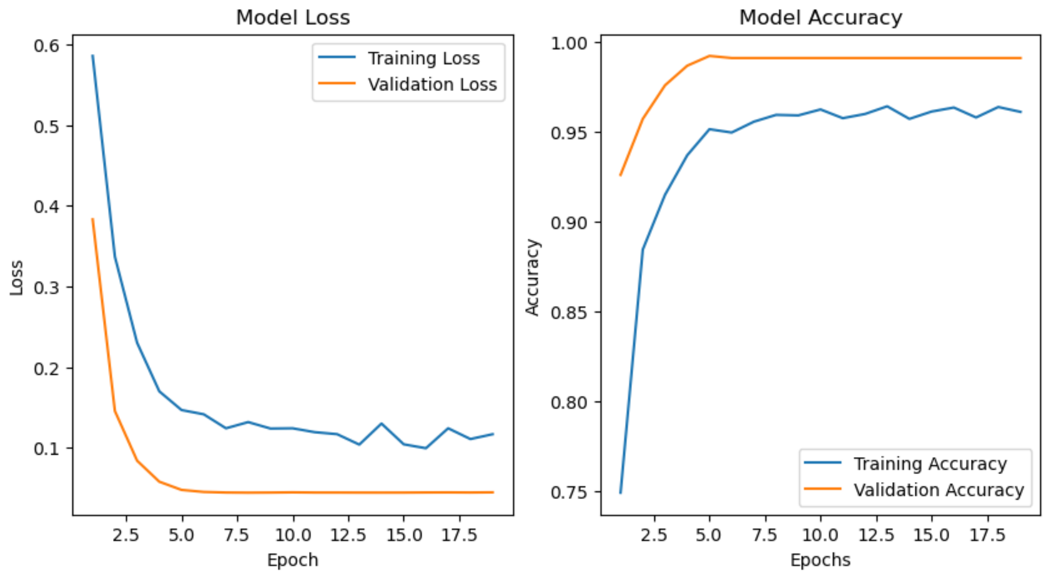


Figure 9: MLP Training Performance

2.2.3 Model Evaluation

Model performance was evaluated using accuracy, precision, recall, and F1 score on both a validation set and two separate test sets to ensure the model's robustness and generalizability.

Table 6: Test1 scores for MLP

Metric	**Value**
Accuracy	0.97
Percision	0.94
Recall	0.99
F1 Score	0.97

Table 7: Test2 scores for MLP

Metric	**Value**
Accuracy	0.99
Percision	0.96
Recall	0.99
F1 Score	0.98

The confusion matrices for both test sets revealed a consistent true positive rate, indicating the model's effectiveness at identifying room occupancy.

2.3 Model 3: Convolutional Neural Network (CNN)

2.3.1 Architecture

Model 3 utilizes a CNN architecture designed for occupancy detection using environmental sensor data. It consists of two 1D convolutional layers (Conv1d) with 32 and 64 output channels, respectively, followed by ReLU activations and a max pooling layer (MaxPool1d) with a kernel size of 2. After flattening, there is a linear layer (Linear) with 64 output units and a ReLU activation, followed by two dropout layers with a rate of 0.5 to prevent overfitting. The final linear layer has a single output unit with a sigmoid activation to output the probability of room occupancy.

The model's parameter summary indicates a total of 14657 trainable parameters, signifying a compact and computationally efficient model suitable for real-time predictions.

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
CNN                                      [1, 1]                    --
├─Conv1d: 1-1                            [1, 32, 5]                128
├─Conv1d: 1-2                            [1, 64, 5]                6,208
├─MaxPool1d: 1-3                         [1, 64, 2]                --
├─Dropout: 1-4                           [1, 128]                  --
├─Linear: 1-5                            [1, 64]                   8,256
├─Dropout: 1-6                           [1, 64]                   --
├─Linear: 1-7                            [1, 1]                    65
==========================================================================================
Total params: 14,657
Trainable params: 14,657
Non-trainable params: 0
Total mult-adds (M): 0.04
==========================================================================================
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.06
Estimated Total Size (MB): 0.06
==========================================================================================
```

Figure 10: CNN Model Summary

2.3.2 Training Process

The CNN model was trained using a batch size of 64, over 100 epochs, with an early stopping mechanism implemented to prevent overfitting. The training process utilized the Adam optimizer with a learning rate of 0.001 and the Binary Cross-Entropy Loss function as the criterion for measuring performance.

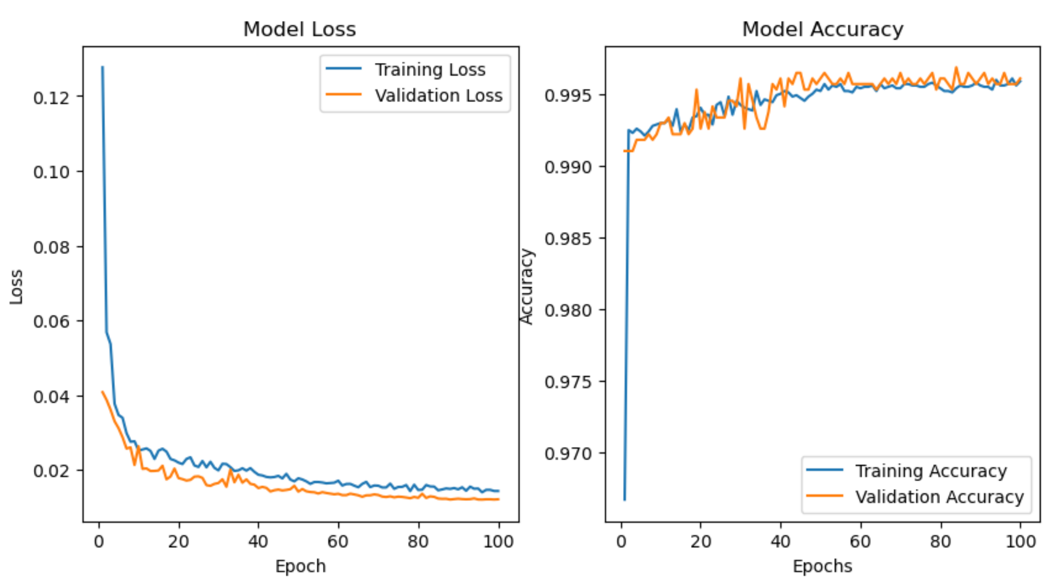


Figure 11: CNN Model Training Performance

2.3.3 Model Evaluation

The training and validation loss curves show a quick convergence, indicating the model's efficiency in learning from the data. The training and validation accuracy reached high levels, staying consistent throughout the later epochs, which demonstrates the model's stability and generalization capabilities. The model was evaluated on two separate test sets, demonstrating high accuracy, precision, recall, and F1 scores:

Table 8: Test1 scores for CNN

Metric	Value
Accuracy	0.93
Percision	0.92
Recall	0.87
F1 Score	0.89

Table 9: Test2 scores for CNN

Metric	Value
Accuracy	0.94
Percision	0.86
Recall	0.84
F1 Score	0.85

The confusion matrix for each test set further confirms the model's predictive accuracy, with a low number of false positives and false negatives. These results validate the CNN model as a robust and reliable tool for occupancy detection.

2.4 Model 4: Long Short-Term Memory Network (LSTM)

2.4.1 Architecture

The architecture of Model 4 is based on a Long Short-Term Memory Network (LSTM), which is particularly suited for time-series prediction due to its ability to remember long-term dependencies. The LSTM model was configured with the following parameters:

```
Model Summary:
lstm.weight_ih_l0.shape = torch.Size([128, 5]) -> 640 parameters
lstm.weight_hh_l0.shape = torch.Size([128, 32]) -> 4096 parameters
lstm.bias_ih_l0.shape = torch.Size([128]) -> 128 parameters
lstm.bias_hh_l0.shape = torch.Size([128]) -> 128 parameters
lstm.weight_ih_l1.shape = torch.Size([128, 32]) -> 4096 parameters
lstm.weight_hh_l1.shape = torch.Size([128, 32]) -> 4096 parameters
lstm.bias_ih_l1.shape = torch.Size([128]) -> 128 parameters
lstm.bias_hh_l1.shape = torch.Size([128]) -> 128 parameters
linear.weight.shape = torch.Size([1, 32]) -> 32 parameters
linear.bias.shape = torch.Size([1]) -> 1 parameters
Total Parameters: 13473
```

Figure 12: LSTM Model Summary

2.4.2 Training Process

The LSTM model was trained using the following setup:Loss function: Binary Cross-Entropy Loss, Optimizer: Adam with a learning rate of 0.001, Batch size: 64, Number of epochs: 100, Early stopping: Implemented to prevent overfitting and to stop training when the validation loss ceases to decrease

The training process involved feeding batches of the preprocessed data through the LSTM network and backpropagating the error to update the model's weights.

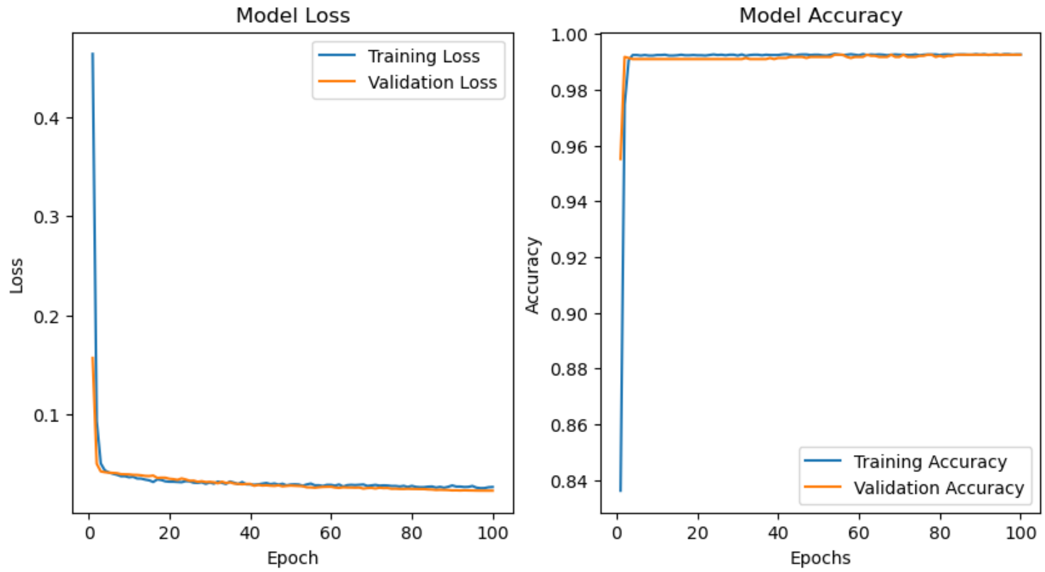


Figure 13: LSTM Training Performance

2.4.3 Model Evaluation

The model was evaluated on a separate validation set not seen during the training phase to ensure an unbiased assessment of its performance.

Model 4 exhibited promising results on both the validation and test sets, achieving high scores across all the evaluation metrics. The model's ability to capture temporal dependencies in the data led to accurate occupancy predictions, as reflected in the precision and recall values.

On the test sets, the LSTM model's performance was characterized by:

- High accuracy, indicating a small proportion of total misclassifications
- High precision and recall, suggesting few false positives and false negatives
- A high F1 score, demonstrating a balance between precision and recall

The confusion matrix provided further insights into the model's performance, with a high true positive rate indicating accurate occupancy detection.

Table 10: Test1 scores for LSTM

Metric	Value
Accuracy	0.97
Percision	0.94
Recall	0.99
F1 Score	0.96

Table 11: Test2 scores for LSTM

Metric	Value
Accuracy	0.95
Percision	0.82
Recall	0.99
F1 Score	0.90

2.5 Model 5: Transformer

2.5.1 Architecture

Model 5 employs the Transformer architecture, renowned for its success in natural language processing tasks, adapted here for the purpose of occupancy detection. The model architecture consists of an encoder and decoder, with each component composed of multiple layers that perform complex transformations. The encoder includes three TransformerEncoderLayers, while the decoder contains six TransformerDecoderLayers. Both the encoder and decoder utilize MultiheadAttention mechanisms to capture dependencies without regard to their distance in the input data.

The model's architecture leverages a high number of parameters, totaling 34,686,977, to capture complex patterns and relationships within the environmental data that are indicative of room occupancy.

2.5.2 Training Process

The Transformer was trained using a robust routine that ensures the model effectively learns from the temporal and contextual relationships within the environmental data. The training was executed for a considerable number of epochs with careful monitoring for convergence, employing techniques such as learning rate scheduling and early stopping to optimize performance and mitigate overfitting.

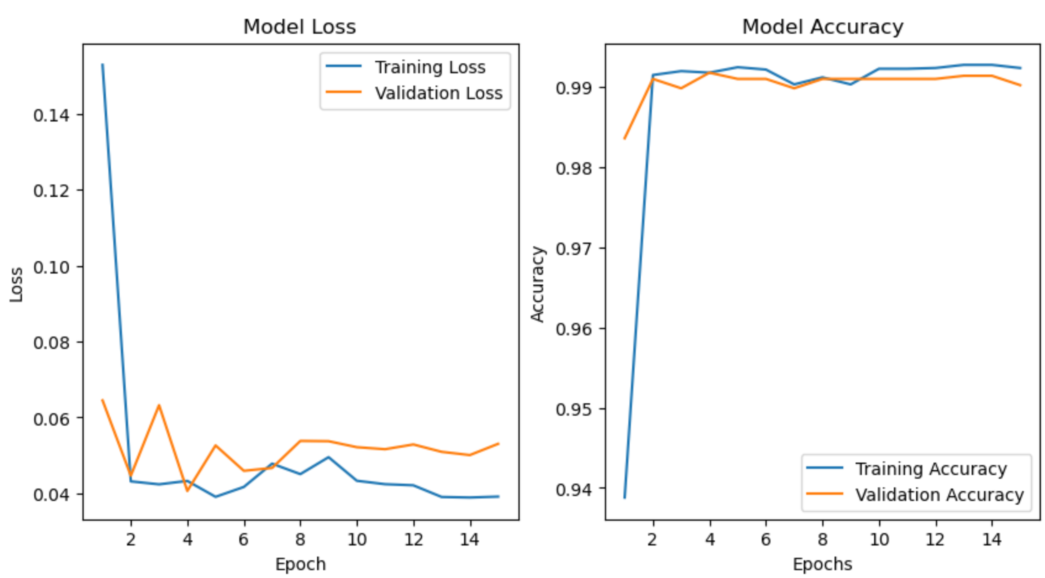


Figure 14: Transformer Training Performance

2.5.3 Model Evaluation

The Transformer model yielded impressive results on both test sets, demonstrating high accuracy, precision, recall, and F1 scores, which are indicative of the model's strong predictive performance:

Table 12: Test1 scores for Transformer model

Metric	Value
Accuracy	0.95
Percision	0.94
Recall	0.93
F1 Score	0.93

Table 13: Test2 scores for Transformer model

Metric	Value
Accuracy	0.97
Percision	0.92
Recall	0.93
F1 Score	0.92

The confusion matrices from both test sets corroborate the model's efficiency, with a remarkably low incidence of false positives and negatives, solidifying the Transformer's suitability for real-world occupancy detection applications.

2.6 Discussion

The RandomForest Classifier shows promise as a reliable occupancy detection model, though its performance varies across different test sets, likely due to unrecognized occupancy patterns or environmental conditions. Enhancements through feature engineering and hyperparameter tuning are suggested. The MLP model, while initially outperforming RandomForest in binary classification, indicates possible overfitting, as seen in its lower performance on a second test set. Future improvements could include hyperparameter adjustments, regularization, and expanded training data to improve generalization. The CNN model, with its simple yet effective architecture, efficiently classifies occupancy with minimal computational resources. The LSTM model excels in predicting occupancy from time-series data, achieving a good balance of precision and recall, important for energy management and user comfort. Lastly, the Transformer model's success in predicting occupancy highlights its potential in intelligent building management, contributing to energy conservation and HVAC automation, with further exploration needed on model improvements and practical applications.