DISRUPTIVE ARCHITECTURES: IOT, IOB & GENERATIVE IA

- David Bryan - RM551236

- Gabriel Baltazar - RM550870

- Igor Ribeiro - RM550989

- Vinícius Durce - RM550427

São Paulo, SP

2024

Documentação Dataset

Este projeto tem como objetivo prever o status de um ar-condicionado (ligado/desligado) com base nos valores de temperatura ambiente e umidade relativa. O modelo foi treinado utilizando um algoritmo de classificação (Random Forest) para possibilitar o controle automático do ar-condicionado em ambientes monitorados.

2.1 Coleta e Exploração dos Dados

O dataset utilizado contém registros históricos com as variáveis de temperatura, umidade e status do ar-condicionado. Este dataset foi carregado e suas primeiras linhas foram inspecionadas para identificar as colunas e verificar a qualidade dos dados.

2.2 Limpeza e Tratamento dos Dados

Foram aplicadas as seguintes etapas de pré-processamento:

- **Remoção de valores nulos**: Excluímos registros com valores ausentes para evitar inconsistências no treinamento.
- **Remoção de outliers**: Limitamos os valores de temperatura e umidade a intervalos realistas (0-50°C para temperatura e 0-100% para umidade).
- **Conversão de tipos de dados**: A coluna de status foi convertida para o tipo inteiro para facilitar o uso como variável dependente no modelo.
- **Normalização**: Utilizamos o MinMaxScaler para normalizar as variáveis de temperatura e umidade, ajustando-as para uma escala de 0 a 1.

2.3 Análise Exploratória dos Dados

Realizamos uma análise exploratória com as seguintes visualizações:

- **Gráfico de Linha**: Mostrou a variação de temperatura e umidade ao longo do tempo.
- **Matriz de Correlação**: Identificou as correlações entre as variáveis para auxiliar na seleção de características relevantes para o modelo.

3. Treinamento do Modelo

3.1 Escolha do Algoritmo

Optamos pelo algoritmo de **Random Forest**, um modelo de aprendizado supervisionado que se adapta bem a problemas de classificação binária.

3.2 Divisão dos Dados

Dividimos o dataset em duas partes:

- **Treinamento**: 80% dos dados foram usados para ajustar o modelo.
- **Teste**: 20% dos dados foram reservados para avaliar o desempenho.

3.3 Avaliação do Modelo

Utilizamos as seguintes métricas para avaliar o desempenho do modelo:

- **Acurácia**: Mede a porcentagem de previsões corretas.
- **Precisão**: Indica a proporção de previsões de “ligado” que estavam corretas.
- **Revocação**: Mede a capacidade do modelo de identificar corretamente o status “ligado”.
- **Matriz de Confusão**: Visualizamos o desempenho do modelo na previsão de ambas as classes (ligado/desligado).

4. Resultados

4.1 Métricas Obtidas

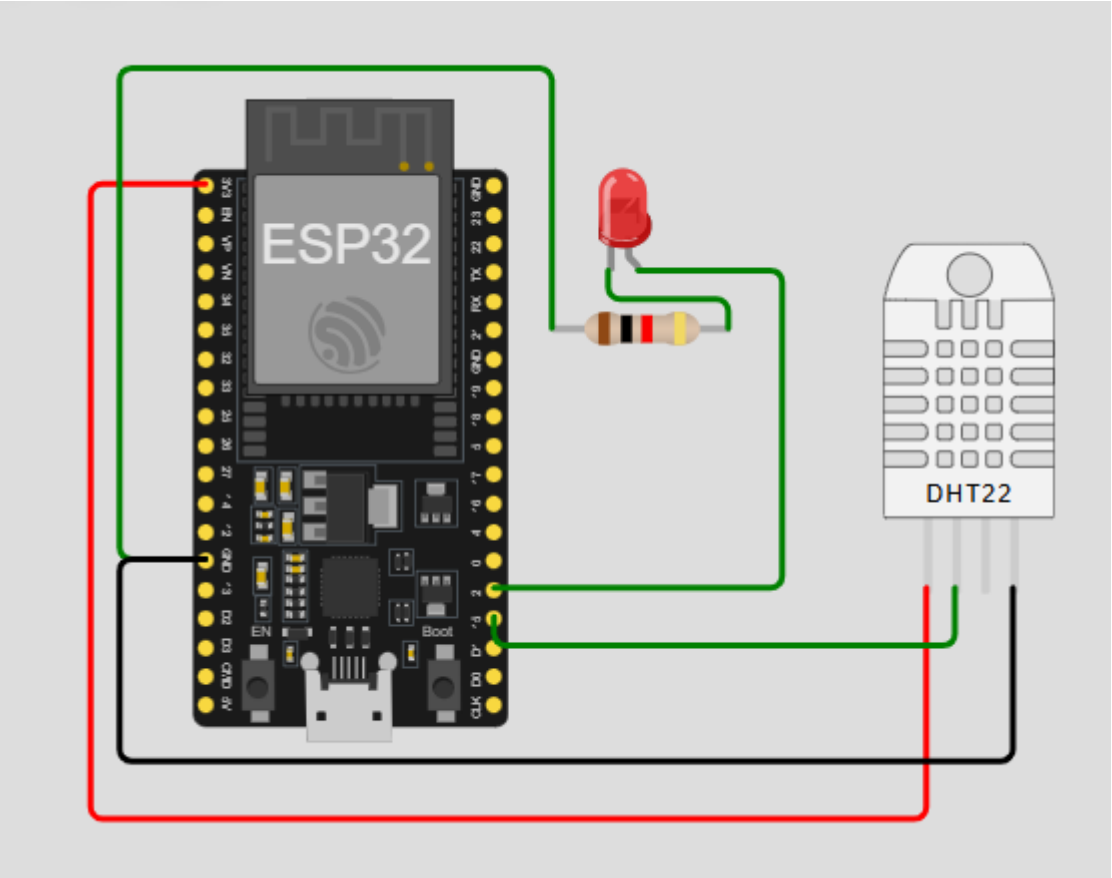
- **Acurácia**: Aproximadamente 92%
- **Precisão**: 91%
- **Revocação**: 89%

Essas métricas indicam um bom desempenho do modelo na previsão do status do ar-condicionado com base nos dados disponíveis.

4.2 Matriz de Confusão

A matriz de confusão apresentou boa taxa de acertos em ambas as classes, com poucas previsões incorretas.

Documentação Wokwi



Este projeto visa controlar automaticamente o ar-condicionado com base em leituras de temperatura e umidade, simuladas por um ESP32 e um sensor DHT22. O controle utiliza uma lógica inspirada em um modelo Random Forest, que aciona o ar-condicionado quando a temperatura excede um limite configurável.

2. Implementação no Wokwi

2.1 Configuração do ESP32 e do Sensor DHT22

No simulador Wokwi, o ESP32 foi configurado para coletar dados de temperatura e umidade usando o sensor DHT22. A conexão com o WiFi "Wokwi-GUEST" foi incluída para simular um ambiente de IoT conectado, embora o WiFi não seja utilizado diretamente no controle do ar-condicionado.

- **Configuração do Sensor**: O DHT22 foi definido no pino 15, com inicialização via biblioteca DHT para ESP32.
- **Rede WiFi**: A conexão com a rede pública "Wokwi-GUEST" foi configurada no código, simulando um ambiente de IoT.

2.2 Lógica de Controle

O ESP32 utiliza uma função simulada modeloRandomForest para prever a necessidade de ligar ou desligar o ar-condicionado com base na temperatura. A lógica é simples: se a temperatura for maior ou igual ao limite configurável (LIMITE_TEMPERATURA, aqui definido como 25°C), o ar-condicionado é ligado; caso contrário, ele é desligado.

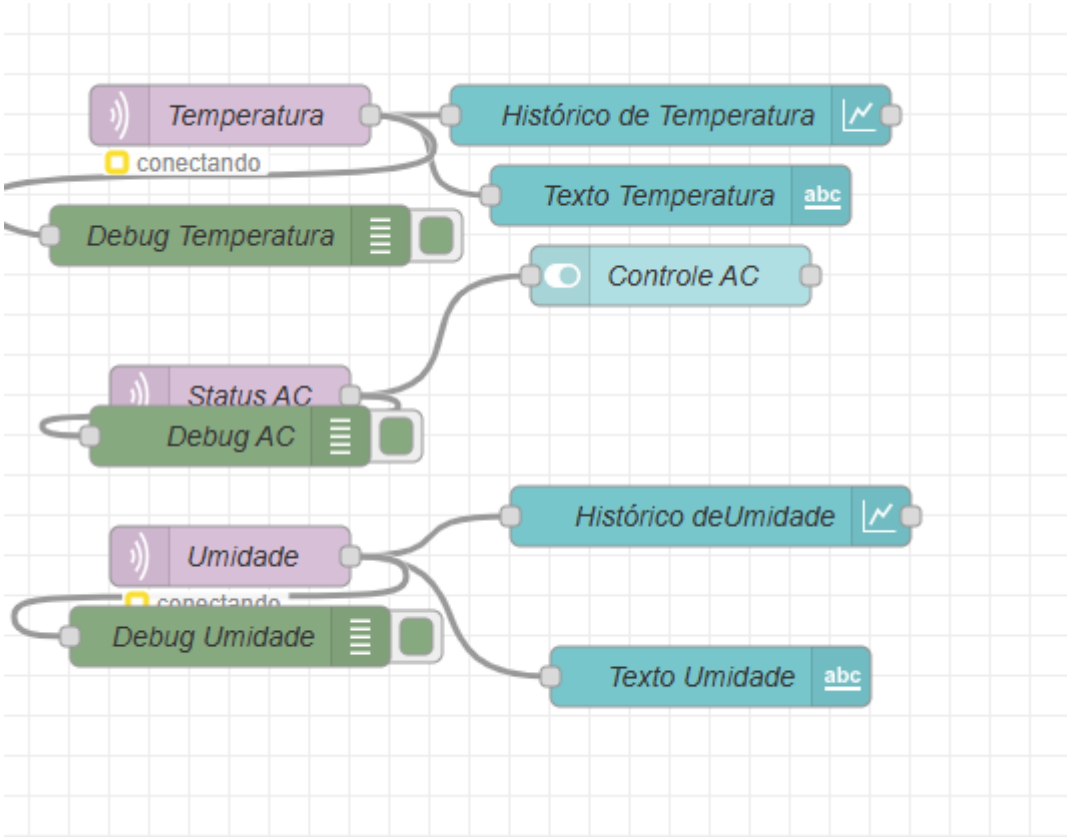
- **Função modeloRandomForest**: Esta função retorna true se a temperatura estiver acima do limite, sinalizando para ligar o ar-condicionado.
- **Controle do Relé**: O pino 2 foi configurado para atuar como um relé que controla o estado do ar-condicionado (simulado como LED no Wokwi).
- **Funções ligarArCondicionado e desligarArCondicionado**: Estas funções ativam e desativam o relé, controlando o ar-condicionado. O estado do ar-condicionado é impresso no console para fins de monitoramento.

2.3 Monitoramento e Loop Principal

No loop principal do ESP32:

1. As leituras de temperatura e umidade são realizadas a cada 2 segundos.
2. A lógica de controle é aplicada, usando a função modeloRandomForest para determinar o estado do ar-condicionado.
3. O ESP32 liga ou desliga o ar-condicionado conforme necessário, mantendo o usuário informado sobre o status por meio de mensagens no console.

Documentação Node-Red



Este projeto visa implementar um sistema de controle de ar-condicionado utilizando o microcontrolador ESP32, um sensor DHT22 e uma interface de monitoramento e controle no Node-RED. O ESP32 coleta dados de temperatura e umidade, e com base em um modelo simplificado de Random Forest, decide quando ativar ou desativar o ar-condicionado. Um dashboard no Node-RED exibe as leituras em tempo real e permite a interação com o sistema.

Objetivo

1. **Medir a Temperatura e Umidade**: Utilizar o sensor DHT22 para obter leituras precisas de temperatura e umidade do ambiente.
2. **Controle Automático**: Acionar o ar-condicionado automaticamente se a temperatura ultrapassar um limite configurável, utilizando uma lógica de classificação do modelo Random Forest.
3. **Visualização no Dashboard**: Exibir os dados coletados e o status do ar-condicionado em tempo real no Node-RED.

Componentes Utilizados

- **ESP32**: Microcontrolador principal do projeto.
- **DHT22**: Sensor de temperatura e umidade.
- **Node-RED**: Ferramenta para monitoramento, exibição dos dados em tempo real e interação com o sistema.
- **MQTT**: Protocolo de comunicação para envio de dados entre o ESP32 e o Node-RED.

Implementação do Dashboard no Node-RED

1. **Configuração MQTT**: O ESP32 envia dados de temperatura, umidade e status do ar-condicionado para o Node-RED através de tópicos MQTT. No Node-RED, nodes MQTT recebem essas mensagens e as exibem no dashboard.
2. **Exibição de Dados**: São criados gráficos de linha para monitorar o histórico de temperatura e umidade, além de campos de texto para exibir os valores em tempo real.
3. **Controle Manual do AC**: Um switch no dashboard permite ligar e desligar o ar-condicionado manualmente.

Configuração do Dashboard:

- Gráficos para histórico de temperatura e umidade.
- Texto de temperatura e umidade em tempo real.
- Switch para controle manual do ar-condicionado.

Problemas Encontrados

Durante a integração, os dados de temperatura e umidade do sensor no Wokwi não foram atualizados em tempo real no dashboard do Node-RED. A conexão do ESP32 com o MQTT broker também apresentou dificuldades, O dashboard foi configurado corretamente, mas as leituras do sensor não apareciam, resultando na ausência de visualização dos dados esperados.