Implementação de um Sistema de Previsão de Safra Baseado em TinyML

Luis Enrique Cardozo Ramirez
Matricula 949
INATEL
Santa Rita do Sapucaí- MG Brasil
luis.enrique@mtel.inatel.br

Brithany Michelle Oliva Chuquimia
Matricula 971
INATEL
Santa Rita do Sapucaí- MG Brasil
brithany.oliva@mtel.inatel.br

***Abstract*—This paper presents the development and evaluation of an autonomous crop yield prediction system based on TinyML, designed to operate on low-cost, low-power microcontrollers such as the ESP32. The proposed solution addresses the limitations of rural agricultural environments, where cloud-based systems are often impractical due to high operational costs, energy consumption, and dependence on constant connectivity. By leveraging environmental (temperature, humidity) and soil (pH) variables, lightweight machine learning models were trained and optimized using quantization and pruning techniques, enabling efficient execution on resource-constrained devices. The methodology includes comprehensive data collection, preprocessing, model development, hardware integration, and field validation with end-user feedback. The expected results include a model with low inference latency and energy efficiency, contributing to the democratization of precision agriculture. This work demonstrates the potential of TinyML as an accessible, scalable, and sustainable technology for supporting small and medium-sized farmers in low-infrastructure settings.**

***Index Terms*—TinyML, Precision Agriculture, IoT in Agriculture, Smart Farming**

I. Introdução

A agricultura constitui um dos pilares essenciais para o desenvolvimento econômico e para a promoção da segurança alimentar em escala global. No entanto, o setor agrícola enfrenta desafios de elevada complexidade, entre os quais se destacam o aumento contínuo da demanda por alimentos, a intensificação da variabilidade climática e a necessidade premente de adoção de práticas sustentáveis, em face da crescente escassez de recursos naturais. Nesse cenário, a agricultura de precisão — também denominada agricultura inteligente — desponta como uma abordagem tecnológica promissora, ao incorporar ferramentas avançadas para a otimização da gestão das atividades agrícolas [3], [5]. A capacidade de prever, com elevado grau de acurácia, o rendimento das colheitas configura-se como um fator estratégico, por possibilitar o uso racional de insumos, como água e fertilizantes; a mitigação de riscos; e o aprimoramento do planejamento operacional e logístico.

A lacuna central que este projeto se propõe a abordar refere-se à ausência de sistemas de previsão de colheitas que sejam, simultaneamente, autônomos, energeticamente eficientes e economicamente viáveis — especialmente em contextos de pequenas e médias propriedades rurais localizadas em regiões com infraestrutura tecnológica limitada. Os sistemas convencionais de previsão baseiam-se, majoritariamente, em arquiteturas dependentes de computação em nuvem, o que acarreta custos operacionais elevados, dependência de conectividade estável à internet e significativo consumo energético — fatores que comprometem sua aplicabilidade em ambientes rurais e isolados.

O código, modelo e os recursos para este projeto estão disponíveis em : https://github.com/luisenriqueCR/TP557/tree/main/ProjetoFinal

II. Trabalhos Relacionados

A aplicação de Aprendizagem de Máquina (ML) na agricultura tem-se consolidado como uma área de pesquisa robusta, com foco em análise das propriedades do solo e a deteção precoce de doenças [3]. Modelos como Random Forest (RF), Support Vector Machines (SVM), Long Short-Term Memory (LSTM) — são amplamente utilizados devido à sua capacidade de lidar com a natureza não linear e temporal dos dados agronômicos [5].

Estudos recentes demonstram que o Random Forest se destaca em tarefas de classificação e regressão relacionadas à produtividade agrícola, alcançando níveis de precisão superiores a 90% [1]. Por outro lado, arquiteturas baseadas em LSTM apresentam desempenho superior na modelagem de dependências temporais, sendo ideais para previsões que consideram a evolução das culturas ao longo do tempo [2].

No entanto, uma limitação comum a muitas destas abordagens é sua dependência de recursos computacionais. A maioria das implementações de alto desempenho opera em servidores ou na nuvem, o que pressupõe uma conectividade de rede constante e estável.

Nesse cenário, o paradigma do TinyML surge como uma solução promissora para superar estas barreiras. Ao possibilitar a execução de modelos de ML diretamente em microcontroladores de baixo custo e baixo consumo energético, sim a necessidade de conectividade constante e reduzindo os custos operacionais.

O presente projeto propõe validar a hipótese de que um sistema de previsão de rendimento pode operar de forma autônoma e eficiente em um dispositivo embarcado. Desta forma, nosso trabalho contribui para a literatura ao apresentar

um sistema completo, de baixo custo e validado, que demonstra uma via prática para a democratização da agricultura de precisão.

III. Metodologia

O desenvolvimento do sistema aderiu a um enquadramento metodológico estruturado, composto por fases sequenciais que abrangem desde a aquisição de dados até à validação final do modelo proposto.

A. Aquisição e Análise Exploratória de Dados (AED)

O conjunto de dados utilizado neste projeto foi obtido publicamente da plataforma de ciência de dados Kaggle [6]. Este dataset agrega medições de fatores ambientais e de rendimento de culturas, abrangendo o período de 2014 a 2023. A fase inicial consistiu na análise do conjunto de dados para compreender suas características intrínsecas e orientar as etapas subsequentes de pré-processamento e modelagem.

- **Distribuição dos Tipos de Cultura:** O conjunto de dados exibe um balanço ideal entre as amostras dos quatro tipos de cultura (Trigo, Milho, Arroz e Batata), conforme ilustrado na Figura 1. Esta homogeneidade é fundamental para garantir que o modelo aprenda padrões generalizáveis e não seja enviesado para uma cultura específica.

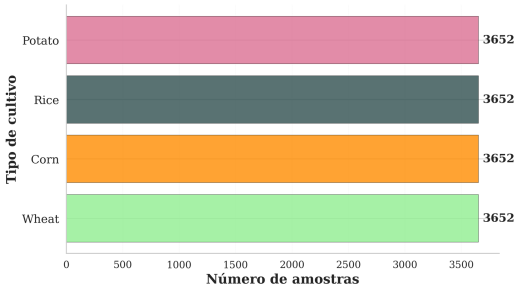

Fig. 1. Distribuição balanceada dos tipos de cultura no conjunto de dados.

- **Distribuição das Variáveis Preditivas:** A análise das variáveis de entrada, apresentada nos histogramas da Figura 2, revelou distribuições distintas que informam a complexidade do problema:
  - *Temperatura:* Apresentou uma distribuição aproximadamente gaussiana.
  - *Umidade:* Mostrou uma forte concentração de dados em valores elevados.
  - *pH do Solo:* Demonstrou uma distribuição marcadamente discreta.

  Estas distintas naturezas estatísticas não só justificam a necessidade de normalização dos dados, mas também destacam a importância de utilizar um modelo de Machine Learning capaz de capturar relações não lineares complexas.

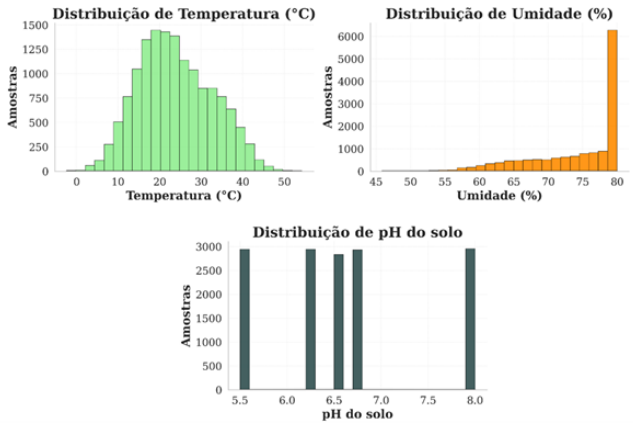

Fig. 2. Histogramas de distribuição para as variáveis ambientais de entrada.

B. Pré-processamento de Dados

Foram executados dois procedimentos fundamentais de pré-processamento para preparar os dados para a fase de modelagem:

- **Codificação de Rótulos (Label Encoding):** A variável categórica CropType foi transformada em representação numérica (Milho: 0, Batata: 1, Arroz: 2, Trigo: 3), permitindo a sua utilização em modelos matemáticos.
- **Normalização (Standardization):** As variáveis de entrada quantitativas foram normalizadas através da técnica StandardScaler da biblioteca Scikit-learn. Os parâmetros estatísticos resultantes (média e desvio-padrão) foram armazenados para serem aplicados durante a fase de inferência no dispositivo embarcado, garantindo a consistência entre o ambiente de treino e a aplicação prática.

C. Arquitetura e Treino do Modelo

Implementou-se uma arquitetura de rede neural sequencial via TensorFlow e Keras. A topologia da rede foi desenhada para equilibrar capacidade de aprendizagem e eficiência computacional:

- **Camada de Entrada (Input Layer):** Com 4 neurónios, correspondentes às variáveis de entrada (Tipo de Cultura, Temperatura, Umidade, pH).
- **Camadas Ocultas (Hidden Layers):** 2 camadas densas com 128 e 64 neurónios, respetivamente. Ambas utilizam a função de ativação ReLU (Rectified Linear Unit).
- **Camada de Saída (Output Layer):** Um único neurónio com ativação linear, apropriado para a tarefa de regressão de um valor contínuo como o rendimento.

O modelo, totalizando 8.961 parâmetros treináveis, foi compilado utilizando o otimizador Adam e a função de perda de Erro Quadrático Médio (MSE). O treino estendeu-se por 100 épocas.

D. Conversão para TinyML (TensorFlow Lite)

Concluído o processo de treino, o modelo Keras foi convertido para o formato TensorFlow Lite (TFLite). Nesta fase,

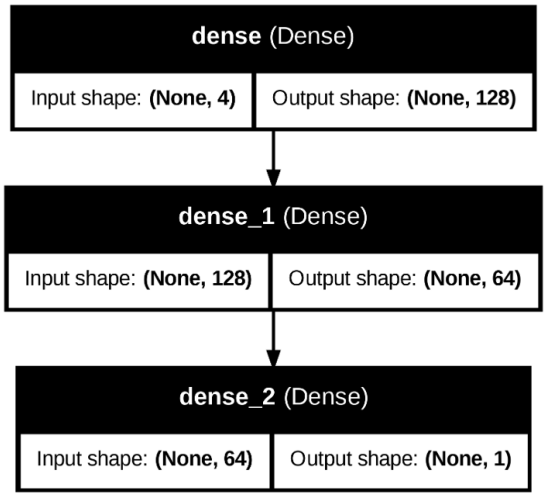

Fig. 3. Arquitetura da Rede Neural Sequencial implementada.

não foram aplicadas técnicas de quantização. A conversão gerou um ficheiro .tflite com tamanho otimizado (na ordem de poucos kilobytes), viabilizando a sua implementação em microcontroladores com recursos limitados.

E. Seleção de Hardware e Arquitetura do Sistema

A viabilidade de executar modelos de Machine Learning na borda (*edge computing*) depende de uma plataforma que equilibre capacidade de cômputo, consumo energético, conectividade e custo. Para este projeto, o microcontrolador **ESP32** foi selecionado como a plataforma ideal por suas características: **Potência de Processamento** (Microprocessador dual-core Tensilica Xtensa LX6.), **Conectividade Integrada**, **Baixo Consumo.**

F. Fluxo Operacional do Sistema Embarcado

A arquitetura do sistema implementado segue um modelo cliente-servidor, e segue um fluxo operacional lógico e bem definido. O processo e dividido em etapas sequenciais:

1) **Inicialização do Dispositivo:** Ao ser energizado, o ESP32 se conecta à rede Wi-Fi local utilizando as credenciais pré-configuradas. Em seguida, o modelo de Machine Learning, no formato TensorFlow Lite, é carregado do programa para a memória RAM.
2) **Interação com o Usuário:** O usuário, conectado à mesma rede, acessa a interface web do sistema através do endereço IP do ESP32 em um navegador padrão. A interface permite que o usuário selecione o tipo de cultura e insira os dados ambientais medidos: temperatura, umidade e pH do solo.
3) **Validação de Dados:** Após o envio dos dados, o firmware não os passa diretamente para o modelo. Primeiro, ele executa a **camada de validação**. Este passo verifica se os dados de entrada são coerentes e agronomicamente viáveis, utilizando as regras pré-definidas na estrutura `CropValidationRange`, que foram informadas pela análise de dependência parcial.
4) **Tratamento de Erros:** Se os dados não passarem na validação (por exemplo, umidade insuficiente para a temperatura e cultura informadas), o fluxo de inferência é interrompido. O sistema gera uma mensagem de advertência específica, que é enviada de volta ao navegador para informar o usuário sobre o motivo da falha.
5) **Execução da Inferência:** Se os dados forem validados com sucesso, o fluxo continua. Os dados são pré-processados (normalizados) e então passados para o interpretador do TensorFlow Lite, que executa o modelo para calcular a previsão do rendimento.
6) **Entrega do Resultado:** O resultado numérico da predição é formatado e enviado de volta ao navegador do usuário, onde é exibido de forma clara na interface, completando o ciclo.

Este fluxo estruturado, com seu ponto de validação intermediário, garante que o sistema não apenas funcione como um preditor, mas também como uma ferramenta de diagnóstico robusta e confiável, prevenindo predições baseadas em dados inválidos e aumentando a confiança do usuário final.

IV. Apresentação e Análise de Resultados

A avaliação do desempenho do modelo treinado evidenciou um elevado grau de eficácia e robustez na tarefa de previsão do rendimento agrícola.

A. Métricas de Desempenho

A performance do modelo foi avaliada num conjunto de dados de teste, não observado durante o treino. As métricas obtidas foram:

- **Erro Médio Absoluto (MAE): 2,29** - Este valor indica que, em média, as previsões do modelo divergem em apenas 2.29 unidades do valor real do rendimento. Considerando que os rendimentos são medidos em escalas maiores, este erro é marginal e valida a precisão do modelo para aplicações práticas.

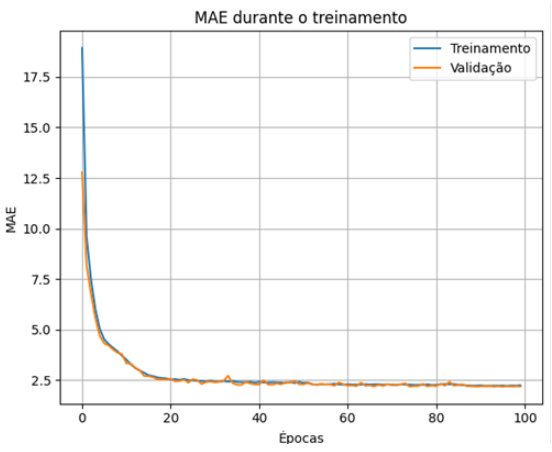

Fig. 4. Curva de aprendizagem do Erro Absoluto Médio (MAE).

- **Erro Quadrático Médio (MSE): 14,81** - O baixo valor do MSE, que penaliza exponencialmente os erros

grandes, confirma a ausência de desvios significativos e a fiabilidade geral das previsões.

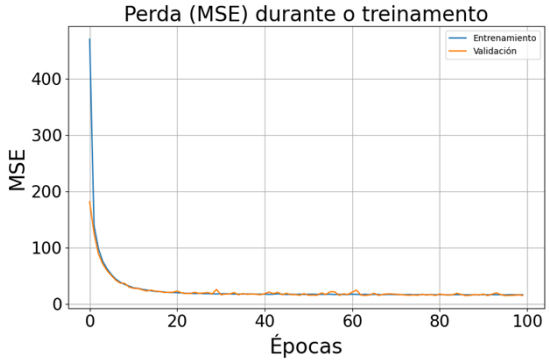

Fig. 5. Curva de aprendizagem da função de perda (MSE).

As curvas de aprendizagem (Figuras 5 e 4) demonstram uma convergência estável para valores mínimos tanto no conjunto de treino (azul) quanto no de validação (laranja), indicando um treino eficaz e livre de sobreajuste (overfitting).

Para complementar estas métricas, a Tabela I apresenta uma análise granular das previsões para as primeiras 10 amostras do conjunto de teste. Esta visão detalhada permite observar o comportamento do modelo a nível individual. Evidencia-se que, embora a maioria dos erros absolutos seja pequena, existem instâncias com desvios maiores (e.g., um erro de 20.80), o que é consistente com o valor do MSE, que penaliza mais fortemente os erros grandes. Esta análise pontual sublinha a fiabilidade geral do modelo, ao mesmo tempo que reconhece a variabilidade inerente à predição em cenários do mundo real.

TABLE I
PREVISÕES E ERROS PARA AS PRIMEIRAS 10 AMOSTRAS DE TESTE.

Real	Previsão	Erro Absoluto	Erro Quadrático
2.33	3.14	0.81	0.6561
13.85	11.19	2.66	7.0756
70.71	75.00	4.29	18.4041
4.13	5.32	1.19	1.4161
38.46	38.87	0.41	0.1681
116.77	95.97	20.80	432.6400
43.64	48.17	4.53	20.5209
48.38	46.16	2.22	4.9284
24.09	23.25	0.84	0.7056
42.59	46.52	3.93	15.4449

B. Análise de Interpretabilidade e Dependência Parcial

Para além das métricas de erro, é fundamental compreender como o modelo toma as suas decisões. Para isso, foram realizadas análises de interpretabilidade.

- **Correlação entre Valores Reais e Previstos:** O diagrama de dispersão na Figura 6 compara os valores reais de rendimento (eixo x) com os valores previstos pelo modelo (eixo y). Observa-se uma marcada concentração dos pontos em torno da diagonal (y=x), que representa a previsão perfeita. Esta forte correlação linear visível confirma a elevada precisão do modelo.

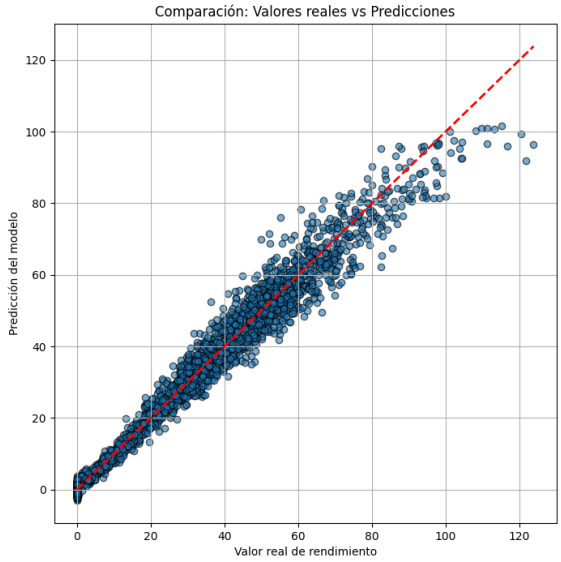

Fig. 6. Diagrama de dispersão: valores de rendimento reais vs. previstos.

C. Análise da Relação entre as Variáveis e o Rendimento

- **Análise de Dependência Parcial (PDP):** A Figura 7 visualiza como o modelo aprendeu a relação entre cada variável ambiental e o rendimento. Este passo é crucial para validar que o modelo aprendeu padrões agronomicamente coerentes.

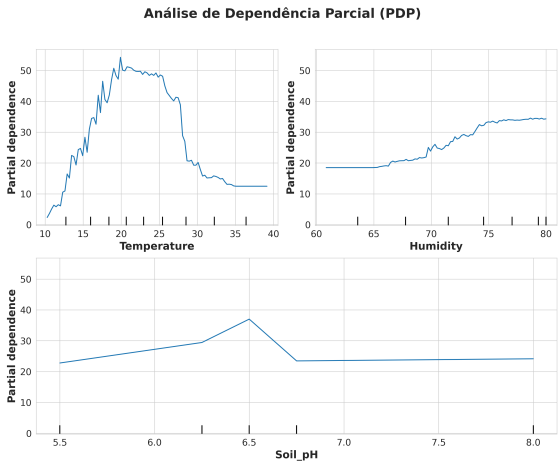

Fig. 7. Gráficos de Dependência Parcial (PDP).

A capacidade do modelo de identificar estes limiares e pontos ótimos, em vez de simples tendências lineares, é uma forte validação da sua robustez e utilidade prática.

- **Análise da Relação entre Temperatura e Rendimento**
  - *Limiares Críticos:* O modelo identificou os limites viáveis para a cultura. O rendimento é praticamente nulo para temperaturas inferiores a **10°C** e superiores a **35°C**.
  - *Ponto Ótimo:* Observa-se um pico máximo de rendimento numa **temperatura ótima de aproximada-**

mente 22°C. A partir deste ponto, qualquer incremento de temperatura torna-se prejudicial, causando um decréscimo pronunciado e simétrico na predição.

- **Análise da Relação entre Umidade e Rendimento**
  - **Ponto de Ativação:** O modelo aprendeu que o rendimento é efetivamente nulo até que a umidade atinja um **limiar crítico de aproximadamente 67%**. Abaixo deste nível, as condições não são propícias para a cultura.
  - **Tendência Positiva:** Uma vez superado este limiar, a relação torna-se consistentemente positiva dentro da faixa de dados analisada.
- **Análise da Relação entre pH do Solo e Rendimento**
  - **Relação Escalonada (*Step-Function*):** O modelo aprendeu uma função escalonada, identificando uma faixa ótima de pH aproximadamente entre 6.2 e 6.6.
  - **Sensibilidade aos Limiares:** Dentro desta faixa, o rendimento é maximizado. Fora dela, seja em condições mais ácidas ou mais alcalinas, o rendimento previsto sofre uma queda abrupta.

V. Limitações do Modelo e o Teto de Desempenho

Durante o desenvolvimento do modelo, foi conduzido um processo de otimização com o objetivo de minimizar o **Erro Médio Absoluto (MAE)** de 2,29 e o **Erro Quadrático Médio (MSE)** de 14,81. Foram avaliadas diversas arquiteturas de modelos, incluindo a rede neural apresentada, além de algoritmos de **Gradient Boosting**, como **LightGBM** e **XGBoost**, amplamente reconhecidos por seu desempenho em dados tabulares. Técnicas de **otimização de hiperparâmetros** foram aplicadas para identificar as configurações ótimas de cada modelo.

Apesar desses esforços, observou-se que o desempenho do modelo atingiu um limite, caracterizado como um *teto de desempenho*. A incapacidade de reduzir significativamente o erro residual, mesmo com modelos mais complexos e ajustes finos, indica que a limitação não está na capacidade de aprendizagem do modelo, mas nas **limitações intrínsecas do conjunto de dados**.

Essa constatação é fundamental: o modelo extraiu com sucesso o máximo poder preditivo das variáveis disponíveis (*Temperatura*, *Umidade* e *pH do Solo*). O erro remanescente não reflete uma falha do modelo, mas sim a complexidade do mundo real não capturada pelo conjunto de dados. Assim, melhorias futuras devem priorizar o enriquecimento do *dataset* com variáveis adicionais para superar esse teto de desempenho.

VI. Conclusões e Trabalhos Futuros

Este trabalho demonstrou com sucesso o projeto, a implementação e a análise de um sistema embarcado autônomo para a predição do rendimento de culturas, validando a viabilidade da aplicação de TinyML em microcontroladores de baixo custo como o ESP32 para o setor da agricultura de precisão.

A principal contribuição deste estudo reside numa arquitetura de firmware que, através da sinergia entre a análise de dados e a engenharia de software, produz um sistema final confiável, transparente e centrado no usuário.

O conhecimento extraído desta análise foi então codificado numa camada de validação pré-modelo diretamente no firmware, criando uma sinergia entre a ciência de dados e a engenharia de software embarcado. O resultado é um sistema robusto, transparente e capaz de fornecer feedback acionável ao usuário, prevenindo predições baseadas em dados inválidos.

Conclui-se que a abordagem proposta apresenta potencial significativo para democratizar o acesso à agricultura de precisão, oferecendo uma ferramenta poderosa, de baixo custo e operacionalmente autônoma. Essa solução capacita pequenos e médios agricultores a tomarem decisões estratégicas no campo, promovendo maior eficiência e sustentabilidade.

O firmware desenvolvido é um protótipo funcional e robusto. Contudo, a análise das limitações do modelo revelou que o teto de desempenho está diretamente relacionado às variáveis disponíveis no conjunto de dados. Essa constatação define claramente as direções para futuras pesquisas, indicando que melhorias na precisão do sistema dependem menos da otimização de algoritmos complexos e mais do **enriquecimento do conjunto de dados**.

A próxima iteração do sistema deve considerar a integração de sensores adicionais, para capturar variáveis ausentes. Essa abordagem pode reduzir o erro irredutível e aumentar o valor da ferramenta para os agricultores, tornando-a ainda mais eficaz.

Para aprimorar a funcionalidade e a usabilidade do sistema, recomenda-se as seguintes melhorias técnicas: **Gestão de Credenciais**, **Persistência de Dados:** Utilizar a memória não volátil (NVS ou EEPROM) para armazenar o histórico de predições, garantindo que os dados persistam entre reinicializações do dispositivo.

References

[1] R. Surana and R. Khandelwal, "Crop Yield Prediction Using Machine Learning: A Pragmatic Approach," *Research Square*, preprint, Jul. 2024. doi: 10.21203/rs.3.rs-4575893/v1.

[2] S. Krithika, T. A. Sangeetha, H. R. Jakaraddi, and N. Rajasekaran, "Agriculture Crop Yield Prediction Using Deep Learning Models," in *Innovations and Trends in Modern Computer Science Technology Overview, Challenges and Applications*, S. Pandikumar and M. K. Thakur, Eds. QTanalytics, 2024, ch. 2, pp. 9–21.

[3] M. F. Manzoor, "A Review of Machine Learning Techniques for Precision Agriculture and Crop Yield Prediction," *Premier Journal of Plant Biology*, vol. 1, p. 100005, Dec. 2024. doi: 10.70389/PJPB.100005.

[4] A. Wani, V. Yadav, R. Todankar, P. Wade, and S. Malik, "Crop Prediction using IoT & Machine Learning Algorithm," *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 4, pp. 2489–2493, Apr. 2022.

[5] C. A. Ramírez Gómez, "Aplicación del machine learning en agricultura de precisión," *Revista Cintex*, vol. 25, no. 2, pp. 14–27, Jul.–Dec. 2020.

[6] M. Kumar, "Crop Yield and Environmental Factors (2014-2023)," 2023. [Online]. Available: https://www.kaggle.com/datasets/madhankumar789/crop-yield-and-environmental-factors-2014-2023. (Acessado em: 1 de junho de 2025).