

# ALGORITMO INTELIGENTE PARA HORTA AUTOMATIZADA

Rodrigo Machado Pedreira<sup>1</sup>; Alessandra Dutra Coelho<sup>2</sup>; Wânderson de Oliveira Assis<sup>2</sup>; Fernando de Almeida Martins<sup>2,3</sup>; Rogério Cassares Pires<sup>3</sup>

<sup>1</sup> Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);

<sup>2</sup> Professor do Instituto Mauá de Tecnologia (IMT);

<sup>3</sup> Engenheiro do Instituto Mauá de Tecnologia (IMT).

**Resumo.** *O trabalho propõe o desenvolvimento de um algoritmo baseado em rede neural que permita a implementação de um sistema de irrigação automatizado e inteligente. Técnicas de machine learning foram empregadas com o objetivo obter-se a previsão do comportamento esperado para o cultivo. O objeto de estudo é uma pequena horta equipada com sensores que monitoram variáveis atmosféricas (temperatura, umidade, pluviometria) e medem a umidade do solo em níveis de profundidade distintos em diferentes pontos. Usando essas informações deseja-se otimizar a irrigação definindo a duração de regadura a fim de atingir a umidade do solo desejada. Espera-se com isso que o plantio tenha melhor qualidade e seja mais produtivo, além de minimizar o consumo desnecessário de água. A transmissão de dados envolve os protocolos LoRa, LoRaWAN e MQTT, além das plataformas Node-RED e ThingSpeak.*

## Introdução

Este projeto vem sendo desenvolvido em anos anteriores por alunos de iniciação científica do IMT (Instituto Mauá de Tecnologia), assistidos pela equipe do centro de pesquisas da instituição. A proposta, desde o início, contemplava a implementação de um algoritmo baseado em rede neural artificial para controlar um sistema de irrigação. Os trabalhos anteriores construíram a base para que isto seja possível.

Foi construída uma horta automatizada, Figura 1, a qual consegue coletar informações por meio de sensores e controlar, de forma remota, válvulas de operação que alimentam os aspersores. Há uma estação meteorológica que monitora variáveis atmosféricas (temperatura, umidade, pluviometria), um sensor que avalia a vazão de água e três trios de sensores que medem a umidade do solo, cada trio instalado em uma posição estratégica e seus sensores dispostos em três níveis de profundidade (10, 30 e 70 cm). Os dados captados são gerenciados por um microcontrolador equipado com um módulo LoRa que os transmite via LoRaWAN para um servidor MQTT. Inversamente, também é possível transmitir dados para o microcontrolador o que possibilita o acionamento a distância de três válvulas solenoides que regulam o fluxo de água para três aspersores.

Um dos primeiros trabalhos relacionados à horta (DIAS e COELHO, 2019)<sup>1</sup> propôs um pequeno sistema composto por um microcontrolador ESP32, um sensor de temperatura e umidade DTH11 e um sensor de umidade de solo comercial. Os dados eram transmitidos via WiFi e MQTT para um servidor Ubidots que possuía dashboards para visualizar os dados.

No mesmo ano, inclusive pelo mesmo autor, o projeto foi melhorado (DIAS e COELHO, 2019)<sup>2</sup>. Neste usou-se o microcontrolador PIC24 que é comparativamente mais barato que o ESP32. O WiFi foi substituído pela tecnologia LoRa, que tem um alcance muito maior, na ordem de quilômetros, e consome muito menos energia podendo entrar em *deep sleep*. Foi incluída a ferramenta Node-RED, que atua como um intermediário no envio dos dados, ainda via protocolo MQTT, e o Ubidots foi trocado pelo ThingSpeak, que produz gráficos que são enviados à plataforma Smart Campus do IMT, Figura 3. Essa nova ferramenta permitiu a

---

<sup>1</sup> Referência Dias e Coelho, Inovação no agronegócio utilizando IoT, 2019.

<sup>2</sup> Referência Dias e Coelho. Monitoramento de sensores de umidade do solo com internet das coisas (IoT) aplicado na agricultura de precisão, 2019.

introdução de outro destinatário que era encarregado de enviar dados para uma aplicação de celular para visualização.

Paralelamente havia outro trabalho (BOYADJIAN *et al.*, 2019) responsável por instalar os sensores, aspersores e uma válvula de vazão com comunicação *Bluetooth* que pode ser programada para ativar em ciclos ou horários predefinidos. As informações armazenadas no ThingSpeak podem ser tratadas utilizando código MatLab, que está diretamente integrado na plataforma ThingSpeak. Por intermédio deste processo, seria possível desenvolver uma rede neural para a irrigação automática da horta.

No ano seguinte, um novo trabalho (KUKU *et al.*, 2020) contribuiu, principalmente, para a parte eletrônica da horta e com a vedação do sistema mecânico responsável pela irrigação. Desta forma o sistema está protegido contra adversidades do ambiente e é possível instalá-lo em ambientes externos, veja Figura 2. O trabalho mais recente (CONTIERI *et al.*, 2021) consolidou as informações dos trabalhos anteriores e fez uma excelente análise do sistema completo da horta automatizada. Além disso acrescentou um sensor de vazão da água no sistema existente e conduziu testes para avaliar e documentar o desempenho do sistema.

Em complemento aos trabalhos anteriores este trabalho propõe o desenvolvimento de um algoritmo baseado em rede neural que permita a implementação de um sistema de irrigação automatizado e inteligente. Técnicas de *machine learning* serão empregadas com o objetivo de filtrar dados incoerentes e obter-se a predição do comportamento esperado para o cultivo.

## Material e Métodos

O projeto faz uso de equipamentos que já foram aprimorados, testados e posicionados pela equipe do Centro de Pesquisa (CP-CEUN-IMT) e por alunos de iniciações científicas anteriores (BOYADJIAN *et al.*, 2019), (KUKU *et al.*, 2020):

- a. Módulo LoRa RN2903 – Módulo transceptor certificado pela ANATEL que utiliza o padrão de comunicação LoRaWAN (*Long Range Wide Area Network*) e opera em 915 MHz. Inclui um microcontrolador e EEPROM. Seus principais atrativos são: baixo consumo energético, longo alcance, alta capacidade de rede e facilidade de uso;
- b. Microcontrolador PIC 24FJ128GA306 – Microcontrolador com modos de operação de baixo consumo avançados. Responsável por captar os sinais gerados pelos sensores e encaminhá-los para o módulo LoRa, e por controlar os atuadores de acordo com os dados recebidos pelo módulo LoRa;
- c. Válvula de vazão solenoide 12VDC – Válvula normalmente fechada e de 2 vias que controla a passagem de água para os aspersores;
- d. Sensor de fluxo de água YF-S201b – Uma pequena válvula equipada com um rotor em formato de catavento com um ímã acoplado. Essa estrutura trabalha em conjunto com um sensor de efeito Hall que envia um sinal PWM, desta forma é possível determinar a vazão de água;
- e. Sensores de umidade desenvolvidos em iniciação científica anterior para monitorar a umidade do solo;
- f. Estação Meteorológica – Coleta informações como: umidade do ar, temperatura atmosférica e pluviometria;
- g. Aspersores Mini Rotor 12SA da Rain Bird.

Além disso foram utilizados os seguintes softwares e tecnologias:

- a. ThingSpeak – É uma plataforma de serviço analítico online para IoT (*Internet of things*) que possibilita agregar, analisar e visualizar dados em tempo real;
- b. Node-RED – É uma ferramenta visual de programação baseada em fluxo, de código aberto e originalmente desenvolvida pela IBM. Foi escrita em JavaScript e fornece um

editor baseado em navegador web que permite, de maneira simples, criar fluxos usando nós e escrever funções em JavaScript. Como executa na plataforma Node.js é necessário tê-la instalada;

- c. LoRa – LoRa (*Long Range*) é uma tecnologia LPWAN (*Low-power wide-area networks*) proprietária de modulação de rádio usada para transmissão de pequenos pacotes de dados à longas distâncias, 5 km em áreas urbanas a até 15 km em áreas rurais (LUCAS e SANTOS, 2018), com um baixo custo e consumo energético. No modelo OSI LoRa é a camada física;
- d. LoRaWAN – LoRaWAN (*Long Range Wide Area Network*) ou LoRaMAC é um padrão de comunicação LPWA (*Low Power Wide Area*) mantido pela LoRa Alliance, baseado na tecnologia LoRa, para gerenciar conexões de rede e definir como os dispositivos usam o *hardware* LoRa. Ele implementa detalhes de funcionamento, qualidade do serviço, segurança e ajustes de potência com o intuito de otimizar o consumo da bateria dos módulos (dispositivos). No modelo OSI LoRaWAN é a camada MAC;
- e. MQTT – MQTT (*Message Queuing Telemetry Transport*) é um protocolo de comunicação para IoT e um padrão aberto OASIS (Organização para o Avanço de Padrões em Informação Estruturada). É simples e leve, desenvolvido para dispositivos com recursos limitados e redes com pouca banda, alta latência e instáveis (características usuais de dispositivos IoT). De acordo com o protocolo de comunicação tradicional MQTT é a camada de aplicação. A comunicação pode ocorrer de maneira assíncrona e é realizada através do protocolo TCP/IP, embora não seja restrita a este.

O trabalho pode ser descrito em etapas. Basicamente os dados são registrados, transferidos, armazenados, tratados e finalmente utilizados para o treinamento da rede neural que controla o sistema de irrigação. A todo momento é possível visualizar os dados em um dashboard no site do Smart Campus Mauá. Durante a coleta são implementados diferentes roteiros de aguçagem.

A Figura 1 mostra o local de trabalho, uma horta de 6m x 6m, enquanto na Figura 2 temos a “central de controle” da horta.

Figura 1– Horta sensorizada



Fonte: Fernando Martins, 2022.

Figura 2– Circuito de controle da horta



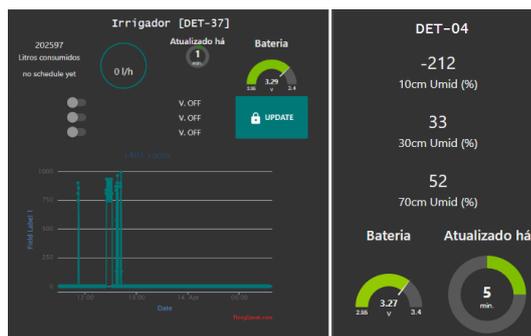
Fonte: (KUKKE *et al.*, 2020).

A primeira etapa consiste em coletar dados utilizando o equipamento previamente instalado na horta (BOYADJIAN *et al.*, 2019), (KUKU *et al.*, 2020). São de interesse: umidade do solo em três níveis de profundidade, temperatura e umidade do ar, pluviometria e vazão de água na irrigação.

Como o intuito é prever e controlar a umidade do solo, escolheu-se medir esses fatores pois a temperatura e umidade do ar afetam a evaporação da água na superfície do terreno, assim como a quantidade de chuva, a vazão e duração da irrigação influenciam diretamente na umidade do solo. As informações atmosféricas são adquiridas pela estação meteorológica, enquanto os dados do solo, assim como os da irrigação, são gerenciados pelo microcontrolador PIC 24FJ128GA306.

Para a transmissão dos sinais foi utilizado o protocolo LoRa (*Long Range*), o qual é uma excelente escolha para dispositivos IoT. Ele permite comunicações síncronas e assíncronas, a longas distancias, robustas à interferência e ruído, com baixo custo de implementação e consumo energético. Em conjunto, foi utilizado o protocolo LoRaWAN, que é a tecnologia que gerencia a transmissão dos sinais. Para trabalhar com os dados localmente escolheu-se usar o programa Node-Red. Esta ferramenta se destaca pela sua facilidade de uso ao usar programação visual, o que torna o desenvolvimento mais rápido e dinâmico. Os dados extraídos dos sensores são enviados para o ThingSpeak (via MQTT), local onde serão armazenados, e para nodes que criam dashboards, que por sua vez ficam disponíveis no site Smart Campus, veja Figura 3.

Figura 3– Dashboard no site do Smart Campus



O Node-Red também está responsável por controlar os irrigadores nos momentos configurados. A lógica consiste em verificar periodicamente o tempo atual e comparar com os horários de ativação e desativação predeterminados, então o sinal é formatado e codificado corretamente para ser enviado via MQTT para o *Network Server*, que, por sua vez, fica encarregado de enviar o sinal para os irrigadores via LoRaWAN.

Na fase de manipulação e tratamento de dados, estes já estão armazenados e disponíveis no ThingSpeak. A plataforma, além de incluir gráficos padrões com um certo grau de customização, permite a criação de gráficos personalizados criados com o MatLab, que é integrado no site, veja Figura 4. O próximo passo é tratar os dados e criar a rede neural que irá descrever matematicamente o comportamento do sistema a fim de gerar um modelo que prediz a umidade do solo. Para isso escolheu-se utilizar a linguagem de programação Python, a qual é uma linguagem interpretada de alto nível com um extenso acervo de bibliotecas, excelente para lidar com um grande volume de dados e com inteligência artificial.

A primeira tarefa é selecionar o período de dados desejados (ex. de setembro a novembro). Os dados recuperados do ThingSpeak estão no formato CSV. Para realizar a manipulação destes utilizou-se a biblioteca Pandas. Em seguida é necessário identificar valores incoerentes (ex. umidade negativa, veja Figura 4), esse processo de identificação será tratado em mais detalhes na seção abaixo.

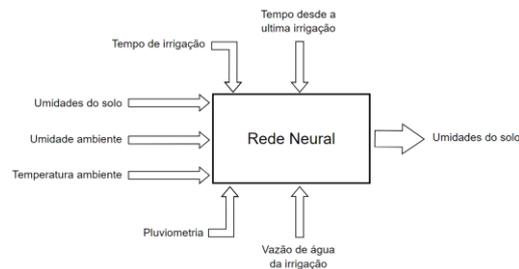
Figura 4– Gráficos padrões do ThingSpeak com exemplos de valores de umidade negativos



Depois os valores precisam ser sincronizados. Por exemplo a entrada de número 100 referente à vazão dos irrigadores é diferente do seu par referente à umidade do solo pois os sensores enviam dados com intervalos de tempo diferentes (ex. a cada 1 min) ou um envio foi perdido (i.e., sinal foi enviado, mas não foi recebido pelo *gateway*). Para isso os dados referentes a cada hora foram agrupados.

Para as umidades, temperatura e vazão foi feita uma média, para pluviometria foi calculado o total relativo (i.e., valor final menos inicial), para o tempo de irrigação uma somatória (efetivamente o total de minutos nesta hora). Por fim um valor auxiliar da quantidade de horas desde a última irrigação foi introduzido. A Figura 5 ilustra o comportamento esperado da rede.

Figura 5– Esquema do funcionamento da rede neural



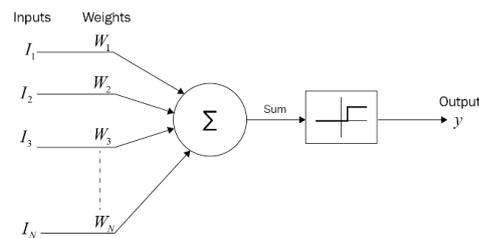
### a) Criação da rede neural

Redes neurais são modelos computacionais inspirados no cérebro humano capazes de realizar aprendizagem de máquina (*machine learning*) e reconhecer padrões. O cérebro possui neurônios que recebem estímulos e transmitem um sinal de resposta para outra célula por meio de conexões sinápticas. Analogamente as redes neurais são constituídas por unidades de processamento simples, batizadas de neurônios artificiais, que processam os sinais recebidos e transmitem um sinal de saída para as unidades conectadas.

A rede neural mais simples é o *perceptron*, Figura 6, inventado em 1958 por Frank Rosenblatt e inspirado no primeiro modelo matemático de um neurônio artificial, desenvolvido por McCulloch and Pitts em 1943. O *perceptron* é um classificador linear, ou binário, i.e., a partir de uma entrada prediz se esta pertence a uma classe ou outra. Essa predição é baseada em uma função matemática aplicada sobre o produto escalar entre um vetor de entrada e um vetor de pesos.

Existem diversos tipos de rede neurais. Este trabalho utilizou um MLP (*Multilayer perceptron*) ou *perceptron* multicamada, que é uma classe de redes neurais FNN (*feedforward neural network*). Como o nome sugere, essa rede tem vários *perceptrons* distribuídos em mais de uma camada de processamento. Em redes do tipo *feedforward*, ou pró-alimentada, os sinais de saída sempre fluem em um único sentido e não há ciclos, pois, caso houvesse, seria classificada como rede recorrente (RNN). Redes neurais podem ser divididas em 3 tipos de camadas: a camada de entrada, a(s) camada(s) oculta(s) e a camada de saída. A Figura 7 ilustra esse fluxo unidirecional e os três tipos de camadas, mas não é uma MLP. Em uma MLP a saída de um node, outra maneira de se referir aos neurônios artificiais, está necessariamente conectada com todos os nodes da próxima camada.

Figura 6– Modelo de um perceptron



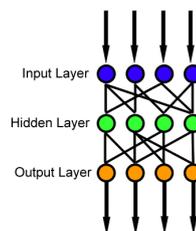
Fonte: <https://www.oreilly.com><sup>3</sup>. Acesso em: 11/11/2022.

A camada de entrada só fornece os dados de entrada e não realiza nenhum processamento. O número de nodes nesta camada é igual ao número de fenômenos de entrada (ex. umidade do solo, temperatura) e tipicamente um para o *bias* (uma constante).

As camadas ocultas ficam entre a camada de entrada e a de saída, e realizam processamento de dados. Costuma-se utilizar a mesma função para todas as camadas ocultas, todavia, independente das funções escolhidas, estas devem ser não-lineares, caso contrário as camadas ocultas seriam redundantes. Não é obrigatório que se tenha camadas ocultas, porém, sem ao menos uma, só é possível classificar dados linearmente separáveis, como mostra a Figura 8. A presença de várias camadas ocultas é uma das características do *deep learning* que visa reconhecer padrões complexos. Na MLP a função de ativação mais comum atualmente é a ReLU (*Rectified Linear Unit*), antigamente o padrão era Sigmoid ou Tangente hiperbólica.

A camada de saída é a última e apresenta uma ou mais saídas, no geral uma. Ela também realiza processamento e pode usar uma função de ativação diferente da camada oculta. Por exemplo, nenhuma das funções mencionadas no parágrafo anterior são empregadas nessa camada. Cada conexão entre os nodes armazena um peso, o qual é um número real que simboliza a influência do dado que passa por esta conexão no sistema. Os pesos podem ser ajustados e é exatamente esse ajuste que confere a propriedade de aprendizagem das redes neurais. Em uma MLP realiza-se a aprendizagem supervisionada. Nesse tipo de aprendizagem tem-se um par de valores de entrada e valores de saída desejados, chamados de dados de treinamento, assim é possível calcular o erro entre a saída real e a saída desejada e ajustar os pesos. O algoritmo mais comum para treinamento das redes neurais *feedforward* (FNN) é chamado *backpropagation*. É importante notar que é possível “treinar demais” uma rede neural, fenômeno chamado *overtraining* ou *overfitting*. Neste caso a rede perde a capacidade de generalizar e reconhecer padrões e reconhece somente dados muito próximos dos de treinamento. A Figura 9 mostra curvas da ativação sigmoide em nodes com pesos diferentes, exceto o peso aplicado ao *bias*. Redes neurais tipicamente têm uma entrada “extra”, chamada *bias*, associado a cada camada de processamento. É simplesmente uma constante conectada aos nodes da camada e seu impacto é análogo ao do termo livre de uma função.

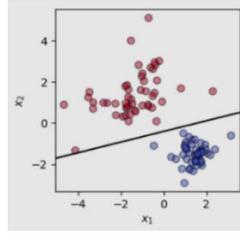
Figura 7 – Esquema de uma rede neural *feedforward*



Fonte: [https://en.wikipedia.org/wiki/File:Feed\\_forward\\_neural\\_net.gif](https://en.wikipedia.org/wiki/File:Feed_forward_neural_net.gif) Acesso em: 16/11/2022.

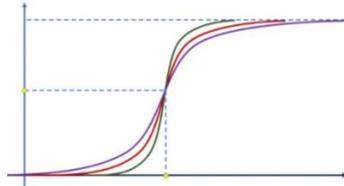
<sup>3</sup> <https://www.oreilly.com/library/view/artificial-intelligence-by/9781788990547/97eeab76-9e0e-4f41-87dc-03a65c3efec3.xhtml>

Figura 8– Linha da separação linear de dados



Fonte: <https://towardsdatascience.com> <sup>4</sup>. Acesso em: 16/11/2022.

Figura 9– Impacto dos pesos em uma sigmoide



Fonte: <https://iaexpert.academy/2020/09/28/importancia-do-bias-nas-redes-neurais> Acesso em: 16/11/2022.

## Resultados e Discussão

### Dados

Infelizmente não foi possível coletar uma boa quantidade de dados. Os sensores não estavam operacionais no início do trabalho e só foram consertados na metade do terceiro trimestre. Desta forma a coleta começou tardiamente, atrasando a implementação do plano de irrigação. Neste plano personalizado a irrigação foi dividida em intervalos de tempo menores durante o decorrer do dia, assim obtém-se mais dados sobre o impacto que a irrigação tem na umidade do solo. Todavia, notou-se que a horta estava alagando, por conta disso o plano foi suspenso e passou-se a irrigar duas vezes ao dia como era feito anteriormente.

Além disso, a maior parte dos dados apresenta alguma irregularidade e é necessário atenção para utilizá-los. Por exemplo, durante um período de 6 dias, um dos sensores do solo apontou 100% de umidade nos níveis de 10 e 30cm de profundidade. Essa ocorrência compromete os dados, pois a influência da irrigação não pode ser observada. Por outro lado, valores como estes podem ser resultantes de chuva, que é um fator de interesse à rede neural proposta. De fato, choveu em grande parte desse período, o que justifica esse padrão.

Este mesmo sensor, após alguns dias, começou a apontar constantemente valores percentuais de umidade negativos. Neste caso não há alternativa a não ser descartar os resultados e tal comportamento põe em dúvida os dados coletados anteriormente. Os dados dos outros dois sensores do solo também apresentaram um comportamento semelhante, porém de maneira mais exacerbada, ao ponto que não foram utilizados.

Quanto ao sensor de vazão, passado um tempo, por volta de um mês, o envio de dados cessou abruptamente. Além disso, assim como os sensores do solo, as medições do sensor de vazão apresentaram algumas peculiaridades interessantes. A Tabela 1 abaixo resume o comportamento dos dados. Os valores medidos têm 8 casas decimais, porém, na análise da tabela, utilizaram-se 7 casas, a fim de livrar-se de algumas dízimas.

Os registros de interesse são principalmente os 2099 reais positivos não nulos. Zeros também tem significado e são importantes para o treinamento da rede neural, porém não é necessária uma análise tão detalhada. Ao examiná-los detectam-se comportamentos anormais como a repetição de um mesmo número e vários múltiplos deste. A Tabela 2 abaixo mostra uma síntese da análise realizada, com destaque para dois números.

<sup>4</sup> <https://towardsdatascience.com/rosenblatts-perceptron-the-very-first-neural-network-37a3ec09038a>

Tabela 1 – Dados de vazão (L/h) coletados

	Total	$\infty$	Zero	$\mathbb{R}^*$	$\mathbb{R}^*_-$	$\mathbb{R}^*_+$	Valores distintos
Ocorrências	92641	3	90436	2202	103	2099	135

Tabela 2 – Demonstrativo da análise dos 2099 valores de vazão (L/h)

		60	63,1578947	Ambos	Restante	Total
Valor exato	Quantidade	0	315	-	1148	2099
	Parcela do total	0%	15%	-	54,7%	100%
Múltiplos exatos > 1	Quantidade	579	596	539	-	-
	Parcela do total	27,6%	28,4%	25,7%	-	-

É uma anomalia valores se repetirem tantas vezes quando se tem 7 casas decimais de precisão, como acontece com 63,1578947 e outros valores. Dentre estes valores reais positivos não nulos existem somente 110 valores distintos e apenas 58 ocorrem uma única vez. Mais que isso, há diversos múltiplos exatos do número 63,1578947 (sem contabilizar ele mesmo), e outrossim para o 60, como mostra a Tabela 2.

Outro problema é algumas medições serem muito baixas, menores que 70 L/h. A vazão medida é a combinação do fluxo de três aspersores, 70 L/h já seria pouco para um aspersor. Por esse motivo valores inferiores a 70 não foram utilizados. Removendo estes pequenos valores dos 2099 sobram 1699 (81%). Quanto aos números destacados na Tabela 2, coincidência não é motivo suficiente para descartá-los. Dessa forma, decidiu-se mantê-los, afinal, embora improvável, um possível motivo para este fenômeno é a resolução do sensor ou o processo de (de)codificação dos dados.

Se a rede neural criada com estes dados provar, por meio de testes reais de irrigação da horta com base na predição, ser ineficaz uma possível solução é descartar os valores múltiplos exatos. Infelizmente não há tempo suficiente para testar a rede criada.

### Rede neural

Primeiramente, foi acertado que apenas dados das três últimas semanas de setembro e da primeira de outubro, um total de um mês, seriam utilizados. Este período engloba o início da coleta de dados e o momento em que o sensor de vazão parou de enviar medições. Durante este intervalo, foram registradas 1684 entradas de irrigação reais e maiores que 70 L/h. Os dados foram agrupados por horas, resultando em 707 registros. A Figura 10 mostra um resumo dos dados que foram utilizados.

Para criação da rede neural decidiu-se utilizar a biblioteca scikit-learn, que possui inúmeras ferramentas para machine learning e é amplamente usada.

Para o treinamento da rede utilizou-se 70% dos dados, os 30% restantes foram usados para teste. Além disso, escolheu-se utilizar 2 camadas ocultas, devido à complexidade dos dados. O número ideal de neurônios na camada oculta é tópico de acirradas discussões e não se chegou a um consenso ainda. Algumas recomendações generalizadas são: média da camada de entrada e a de saída; 2/3 da camada de entrada somados à camada de saída; a raiz quadrada da multiplicação entre a entrada e a saída. A rede projetada tem 9 entradas e 3 saídas, mas se as “regras” fossem seguidas ter-se-ia: 6, 9, 5, respectivamente. Utilizou-se o algoritmo MLPRegressor, a função de ativação Relu, o algoritmo otimizador de pesos L-BFGS, que é recomendado para data sets menores, e por fim, um algoritmo de normalização de dados nos dados de treinamento, o StandardScaler.

Figura 10– Tabela gerada pelo Pandas dos dados utilizados na MLP

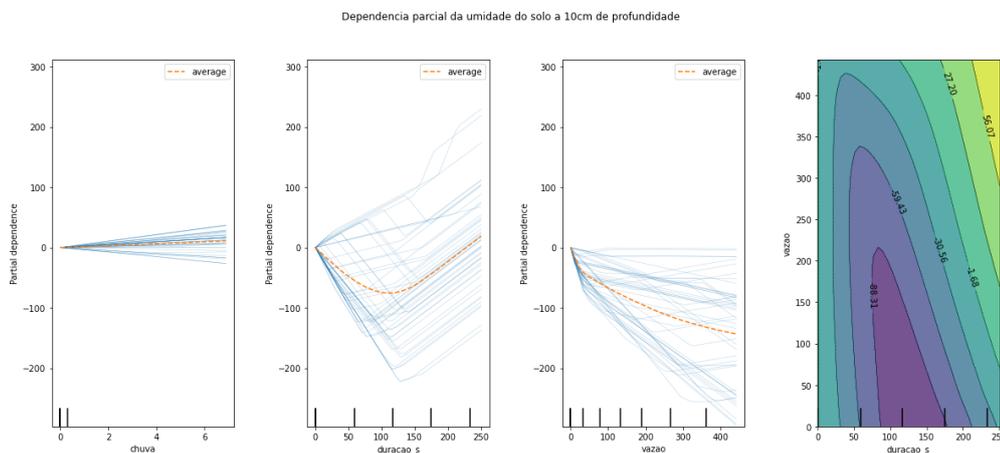
cdate	umid10	umid30	umid70	temp	umid	vazao	chuva	horas_desde	duracao_s
2022-09-07 10:00:00	56.625000	82.000000	64.125000	14.751563	92.609375	0.000000	0.3	0	0.0
2022-09-07 11:00:00	58.500000	83.375000	65.125000	14.740323	89.483871	224.535198	0.3	0	643.0
2022-09-07 12:00:00	59.428571	77.428571	63.000000	15.639344	88.885246	0.000000	0.0	1	0.0
2022-09-07 13:00:00	57.625000	76.500000	61.625000	17.073333	86.800000	0.000000	0.0	2	0.0
2022-09-07 14:00:00	61.875000	74.250000	61.250000	18.201515	79.863636	0.000000	0.0	3	0.0
...	...	...	...	...	...	...	...	...	...
2022-10-06 16:00:00	70.714286	63.571429	34.714286	30.573529	37.602941	0.000000	0.0	8	0.0
2022-10-06 17:00:00	69.000000	63.444444	34.888889	29.489394	38.681818	0.000000	0.0	9	0.0
2022-10-06 18:00:00	68.142857	63.142857	34.714286	27.641538	43.030769	0.000000	0.0	10	0.0
2022-10-06 19:00:00	69.333333	63.777778	35.111111	25.666667	57.590909	130.116435	0.0	0	117.0
2022-10-06 20:00:00	91.250000	67.625000	42.375000	20.698462	86.707692	0.000000	0.3	0	0.0

707 rows x 9 columns

Inicialmente foram utilizados 9 neurônios na primeira camada oculta e 5 na segunda. À primeira vista parecia um ótimo resultado com coeficiente de determinação 0,99 (quanto maior melhor, máximo é 1) e estatísticas de erro baixíssimas, porém um resultado tão “perfeito” para um sistema tão grande é causa para suspeita. De fato, analisando as dependências parciais e condições de expectativa individuais de alguns parâmetros, ficou claro que o modelo certamente não poderia estar correto.

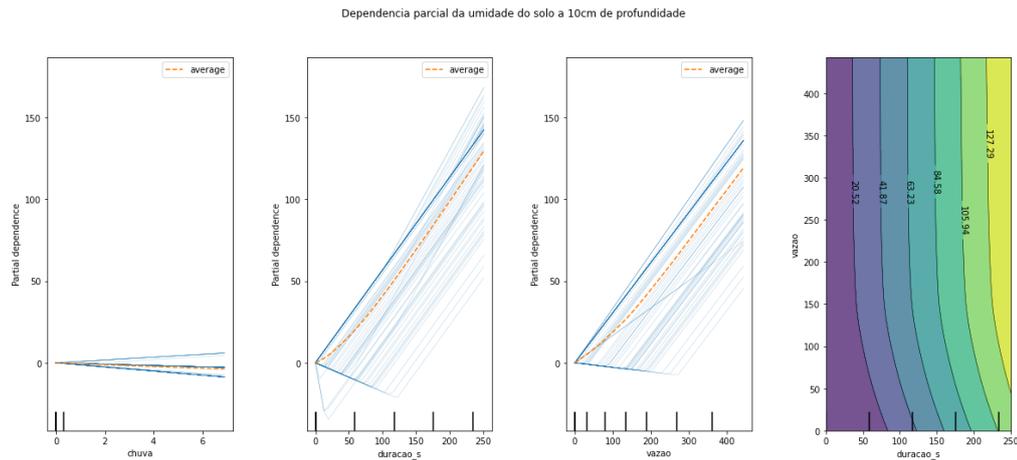
A Figura 11 mostra alguns gráficos, cujas curvas foram centralizadas, que mapeiam a influência de um critério na umidade do solo a 10cm de profundidade. Resultados negativos indicam que a variável avaliada tem o efeito de reduzir a umidade do solo. É de se esperar que a duração da irrigação e a vazão exerçam uma forte influência no aumento da umidade, porém o modelo indica exatamente o contrário. Da esquerda para a direita têm-se: quantidade de chuva, duração da irrigação, vazão, e duração e vazão juntos no mesmo gráfico.

Figura 11– Gráficos incoerentes da dependência parcial centralizada da umidade do solo à 10cm



Decidiu-se reajustar alguns parâmetros do MLP visando melhores resultados e concluiu-se que a combinação de 7 neurônios nas duas camadas forneceu um resultado aparentemente satisfatório, a Figura 12 mostra os mesmos gráficos da Figura 11, porém referentes à nova rede neural. Lamentavelmente não foi possível pôr a prova a MLP por questões de tempo.

Figura 12– Gráficos coerentes da dependência parcial centralizada da umidade do solo à 10cm



## Conclusões

Não é possível assegurar a viabilidade da implementação de uma MLP com estes parâmetros escolhidos, uma vez que, não há tempo suficiente para utilizar a rede neural desenvolvida na prática e escutar sua eficácia. Além disso, como não se obteve uma quantidade de dados grande o suficiente, somado ao fato de que os disponíveis não estavam com uma boa qualidade, a rede treinada não é confiável.

Apesar da conclusão acima outros trabalhos na área demonstram o imenso potencial do IoT e inteligência artificial na agricultura. Produtos de melhor qualidade, menos água desperdiçada, mais eficiência na colheita, para citar alguns benefícios.

O sistema todo, desde o registro dos sensores até o acionamento remoto da irrigação, se mostrou mais uma vez funcional. Para trabalhos futuros é recomendado focar na coleta de dados e acompanhar de perto as medidas tomadas pelos sensores. Estima-se que um período de 4 meses de coleta seja um bom ponto de partida para começar o treinamento da rede.

É uma boa ideia, também, investigar outros tipos de rede neural, em especial as que tem “memória” (RNN), tendo em vista que os acontecimentos das horas anteriores, principalmente a antecedente, exercem um impacto na umidade do solo atual. No modelo deste trabalho a única variável que propagava informações do passado era a que media a quantidade de horas desde a última irrigação, uma abordagem um tanto quanto rudimentar.

## Referências Bibliográficas

BOYADJIAN, B. P.; ASSIS, W. O.; COELHO, A. D.; MARTINS, F. A.; PIRES, R. C. **Sistema inteligente para gestão de recursos hídricos na agricultura de precisão utilizando rede neural artificial**. 11º Seminário Mauá de Iniciação Científica. São Caetano do Sul. 2019.

CONTIERI, E.; COELHO, A. D.; ASSIS, W. O.; PIRES, R. C. **Estudo da automação de sistema de irrigação utilizando redes neurais artificiais embarcada em aplicação de internet das coisas**. Escola de Engenharia Mauá. São Caetano do Sul. 2021.

DIAS, B. G. L.; COELHO, A. D. **Inovação no agronegócio utilizando IoT**. Escola de Engenharia Mauá. São Caetano do Sul. 2019.

DIAS, B. G. L.; COELHO, A. D. **Monitoramento de sensores de umidade do solo com internet das coisas (IoT) aplicado na agricultura de precisão**. Escola de Engenharia Mauá. São Caetano do Sul. 2019.

KUKE, A. S.; VETTORAZZO, L. G.; ASSIS, W. O.; COELHO, A. D.; MARTINS, F. A. **Controle de sistema de irrigação automático utilizando internet das coisas e conceitos de redes neurais artificiais**. Escola de Engenharia Mauá. São Caetano do Sul. 2020.

LUCAS, B. V.; SANTOS, T. B. **Luz de emergência Smart**. Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá. São Caetano do Sul. 2018.