Predição de Casos de Dengue na Cidade de Fortaleza-CE Utilizando Internet das Coisas e Aprendizado de Máquina

Nicodemos Freitas[1], **Emanuel Bezerra Rodrigues**[2]

[1] Programa de Mestrado e Doutorado em Ciência da Computação (MDCC)
Universidade Federal do Ceará (UFC) - Fortaleza/CE - Brasil

[2]Programa de Mestrado e Doutorado em Ciência da Computação (MDCC)
Universidade Federal do Ceará (UFC) - Fortaleza/CE - Brasil

Abstract. *This work presents a contribution in the e-health area and uses, for this, Internet of Things (IoT) tools and application of Machine Learning (ML) to predict dengue cases in weeks in the future. It analyzes weather, population and dengue data to identify outliers, selects variables according to Spearman and Pearson correlation levels, as well as compares Machine Learning models using Mean Absolute Error (MAE) and Coefficient of Determination R^2 as metrics. This work proposes and implements an architecture composed of a weather station simulator and a back-end application that integrates with an IoT platform called dojot that receives dengue case forecasts as notifications.*

Resumo. *Este trabalho apresenta uma contribuição na área de e-health e utiliza, para isso, ferramentas de Internet das Coisas (IoT) e aplicação de aprendizado de máquina para prever casos de dengue em semanas no futuro. Ele faz uma análise de dados meteorológicos, populacionais e de dengue para identificar valores discrepantes, seleção de variáveis a partir de níveis de correlação de Spearman e Pearson, bem como faz comparação de modelos de aprendizado de máquina utilizando como métricas o erro médio absoluto (MAE) e o coeficiente de determinação R^2. Este trabalho propõe e implementa uma arquitetura composta por um simulador de estação meteorológica e uma aplicação back-end que faz a integração com uma plataforma de IoT chamada dojot que recebe como notificação previsões de casos de dengue.*

1. Introdução

Ao longo de 2007 e 2008, o estado do Ceará passou por duas epidemias de arboviroses que resultaram em prejuízos financeiro e humanitário para a população em geral e para os profissionais de saúde. A primeira delas ocorreu em Sobral (CE), em 2007, com 4.434 infectados. A segunda epidemia ocorreu na cidade de Fortaleza (CE), em 2008, com 67.857 infectados [Valter et al. 2020]. Surtos ou epidemias causam superlotação em postos de saúde e hospitais, causam altos custos e complicações para a população. Ações para conter a doença e acabar com a causa epidêmica ou minimizar os efeitos podem ser realizadas se as autoridades souberem com a devida antecedência de informações relacionadas ao problema. Portanto, a necessidade de se adotar iniciativas que abordem o tema em questão se torna relevante, mitigando, assim, prejuízos causados pelas epidemias e possibilitando a diminuição de novos episódios ou sua neutralização.

Este trabalho apresenta uma contribuição na área de e-health, utiliza ferramentas de Internet das Coisas (do inglês Internet of Things (IoT), análise de dados para identificar

valores discrepantes e correlações entre variáveis, e aplicação de aprendizado de máquina para predizer quantitativos de casos de dengue para a 5ª semana à frente. As predições são feitas para a cidade de Fortaleza (CE), estudo de casos deste trabalho. Ele propõe e implementa uma arquitetura composta por um simulador de estação meteorológica e uma aplicação *back-end* que faz a integração com uma plataforma de IoT chamada dojot [CPqD 2021]. A dojot foi desenvolvida pelo Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD) com o objetivo de suportar tecnologias para as cidades inteligentes [Carvalho et al. 2021][Santos et al. 2020], e foi utilizada levando em consideração o objetivo do trabalho em questão.

Este trabalho está organizado em cinco seções. Está seção fez uma introdução ao tema de pesquisa, contextualizando a motivação e os objetivos da pesquisa. Na seção 2, são apresentados os trabalhos relacionados. No seção 3, a proposta é descrita. A seção 4 mostra os resultados alcançados por este trabalho de pesquisa. Por fim, a seção 5 apresenta as conclusões sobre os resultados alcançados.

2. Trabalhos Relacionados

A maioria dos trabalhos citados tem objetivos semelhantes aos do trabalho em questão. [Tavares and Rodrigues 2018] faz uma análise de variáveis meteorológicas e populacionais e suas relações com casos de arboviroses. Já em [Valter et al. 2020] uma análise de correlação é feita utilizando o método de correlação de Pearson sobre as variáveis estudadas e utiliza uma rede neural Perceptron Multi-Camadas (MLP) para predizer casos futuros de dengue até a 15ª semana à frente.

A contribuição de [Varela and Vívian 2016] tem uma abordagem diferente; a intenção não é fazer predição de casos nem correlacionar variáveis. Trata-se do uso de georreferenciamento para visualizar locais com maiores focos de dengue. Um trabalho que se assemelha bastante com [Valter et al. 2020] foi desenvolvido na Colômbia em níveis departamentais [Zhao et al. 2020]. No entanto, [Zhao et al. 2020] fez uma análise de dados e comparou dois modelos de aprendizado de máquina, uma rede neural e um modelo baseado em árvore de decisão chamado floresta aleatória. [Singh et al. 2018] não fez predição de casos de dengue e, em vez disso, utilizou apenas uma árvore de decisão para classificar os sintomas dos usuários em três categorias: não infectado, infectado e gravemente infectado.

Em [Othman and Danuri 2017] foi proposto um *Framework* para alerta precoce de surto de dengue na Malásia. Seu objetivo é prever epidemias e em seguida apresentar as informações aos usuários via aplicação Web ou aplicativo móvel. No entanto, o Framework não foi implementado. A contribuição de [Sareen et al. 2017] foi o uso de georreferenciamento combinado com o pré-diagnóstico de casos de *Zika* utilizando o classificador *Naive Bayes* (NB), em que o objetivo é receber como entrada informações de sintomas e dar como resposta um pré-diagnóstico, se a pessoa está ou não com *Zika*.

As contribuições do trabalho em questão se destacam das dos demais por agregarem dados de diferentes fontes a uma plataforma de IoT, dados estes que vêm de dispositivos físicos de IoT como dados meteorológicos e de *DataSets* locais que fornecem dados de agravos de casos de dengue. Outro importante diferencial deste trabalho é fazer uma análise dos dados e comparações de modelos de *Machine Learning* (ML), inclusive com modelos treinados e testados na literatura. Finalmente, foi feita a integração de

uma aplicação *back-end* com uma plataforma de IoT gratuita para armazenar, processar e notificar predições de casos de dengue para a 5ª semana à frente. Um resumo sobre a diferença entre os trabalhos é apresentado na tabela 1.

Tabela 1. Comparações de Trabalhos

Trabalhos Relacionados	Local de Estudo	Período	Faz análise de dados	Doença Estudada	Utiliza modelo de ML	Faz comparação de modelos de ML	Utiliza IoT como complementação
TAVARES; RODRIGUES, 2018)	Fortaleza-CE	2011-2017	✓	Dengue, Zika e Chikungunya	✗	✗	✓
VALTER et al., 2020	Fortaleza-CE	2007-2019	✓	Dengue	✓	✗	✗
ZHAO et al., 2020	Colômbia	2014-2018	✗	Dengue	✓	✓	✗
SAREEN et al., 2017	Amritsar, Índia	2016	✓	Zika	✗	✗	✗
OTHMAN; DANURI, 2017	Malásia	Não informado	✗	Dengue	✗	✗	✗
SINGHet al., 2018	Índia	2010	✓	Dengue	✓	✗	✓
VARELA; VÍVIAN, 2016	Distrito Federal	2014-2016	✓	Dengue, Zika e Chikungunya	✗	✗	✗
Este Trabalho	Fortaleza-CE	2007-2020	✓	Dengue	✓	✓	✓

3. Proposta

Este trabalho propõe uma integração entre uma estação meteorológica com sensores embarcados, um *Middleware* de IoT e uma aplicação *back-end*, com o objetivo de predizer casos de dengue que podem ocorrer no futuro. Um dispositivo físico é utilizado como estação meteorológica para enviar dados a dojot. A dojot [Xavier 2020] é utilizada para armazenar os dados coletados da estação meteorológica, disponibilizar esses dados de forma escalável, desencadear eventos e disparar notificações de previsões de casos de dengue. A aplicação é utilizada para tratar dados consumidos da dojot e fazer predição de casos de dengue para a 5ª semana à frente.

Todos os módulos que fazem parte da arquitetura possuem um objetivo específico que se complementa com os demais. A composição da aplicação *back-end* inclui tecnologias como a linguagem de programação *python* 3.8 para comunicação MQTT e HTTP, módulos *Python* como Pandas e *Numpy* para tratamento de dados e *Keras, XGboost, TensorFlow* e *Sklearn* para Aprendizado de Máquina. A aplicação em questão foi desenvolvida e dividida em duas partes que são executadas em diferentes dispositivos de hardware de forma independente. São eles: uma *Raspberry Pi* e um notebook. A comunicação com a dojot ocorre de duas formas: via MQTT, que se trata de uma comunicação sem retorno da *Raspberry Pi* para a dojot; e via comunicação HTTP, que é bidirecional e pode ser feita da aplicação, executando no notebook via módulo integrador descrito a seguir, ou ainda pode ser feita da dojot via módulo responsável pela execução do fluxo. A ilustração da arquitetura do sistema proposto é mostrada na figura 1.

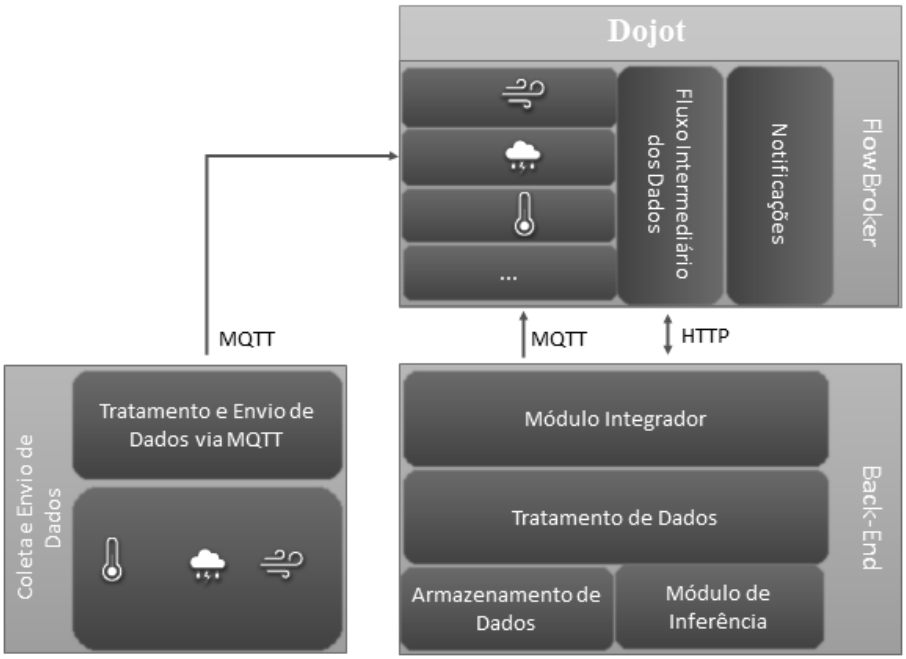


Figura 1. Arquitetura Geral da Proposta

3.1. Coleta e Envio de Dados

A coleta e o envio de dados são compostos por dois módulos em dois dispositivos físicos diferentes. Os módulos Sensores Virtuais e Tratamento e envio de dados são implementados em uma *Raspberry Pi* e se comunicam através do protocolo MQTT com a plataforma dojot, a qual possui o *broker* MQTT. A *Raspberry Pi* emula o funcionamento de uma estação meteorológica que coleta, agrega e envia dados. A escolha da *Raspberry Pi* se deve ao seu propósito de ser um minicomputador e, ao mesmo tempo, ter a possibilidade de agregar sensores por meio de seus pinos. Como neste trabalho é preciso fazer tratamento de dados antes de serem enviadas informações para a camada superior, foi preferível utilizar a *Raspberry Pi* pelo fato de ela já suportar a linguagem de programação Python, contando com todos os módulos de tratamento de dados, de comunicação MQTT, além de ela permitir armazenamento de grandes arquivos.

Os sensores virtuais fazem o papel de dispositivos físicos, pois eles simulam a origem dos dados na borda da rede. Eles foram implementados na forma de *scripts* escritos na linguagem de programação *Python* 3.8. Os sensores podem enviar quaisquer tipos de dados adquiridos de diferentes fontes. Nesse módulo, a informação enviada para a camada superior ainda não foi tratada e, portanto, está no mesmo formato em que foi adquirida. Os sensores virtuais implementados neste trabalho em questão são: sensor de precipitação de chuva, velocidade do vento e umidade relativa do ar.

O principal objetivo do módulo de tratamento e envio de dados é adquirir dados de fontes heterogêneas, tratar e enviar essas informações via comunicação MQTT como se os dados estivessem vindo de uma única fonte, abstraindo o tipo de comunicação e o formato da informação recebida pelos sensores virtuais. Os dados armazenados na estação meteorológica e aplicação *back-end* encontram-se em formato CSV, e a dojot recebe esses dados no formato *Json*. De forma contraria, o módulo de inferência discutido na seção

3.3, recebe as informações para fazer predições no formato CSV, portanto, fica justificado a necessidade de módulos para tratar dados na estação meteorológica simulada e no *Back-End*. Todas as informações enviadas via protocolo MQTT estão associadas a um tópico, que é gerado assim que os dispositivos virtuais são criados na dojot, de tal forma que para cada dispositivo virtual há um tópico diferente seguindo o modelo *publish/subscribe*.

3.2. Dojot

Torna-se primordial o papel da dojot nessa arquitetura, pois ela armazena dados de diferentes fontes, associando-os a um único domínio, tornando os dados disponíveis através de consulta, além de ter funcionalidades como fácil integração com outras plataformas ou aplicações por meio do módulo *Flowbroker*. Há ainda outros recursos que poderiam ser facilmente associados, como a programação de eventos, que pode desencadear requisições HTTP da plataforma para qualquer destino de interesse.

Na arquitetura implementada, as informações enviadas à plataforma dojot têm duas origens: i) dados de um *dataset* com informações de casos de dengue e dados populacionais. Os dados de dengue e populacionais são organizados por data e são enviados pela aplicação *back-end* via comunicação MQTT. As duas formas de envio existem devido à natureza dos dados e à temporalidade com que eles são coletados na realidade – os dados meteorológicos, por exemplo, são coletados por hora. O envio dos dados populacionais é desencadeado por uma requisição HTTP disparada da dojot e corresponde à mesma data da dos dados meteorológicos; a segunda forma de envio é utilizando uma *Raspberry Pi*, que faz o papel de uma estação meteorológica que envia dados como precipitação de chuva, velocidade do vento e umidade do ar, os quais são enviados via protocolo MQTT. Os dados enviados da estação meteorológica e da aplicação *back-end* para a dojot são dados reais separados para teste na fase de treinamento dos modelos de aprendizado de máquina.

A plataforma dojot foi escolhida na contribuição deste trabalho devido a sua fácil instalação, à documentação fácil de se explorar e suprir todas as necessidades impostas por este trabalho, com exceção da falta de um módulo de aprendizado de máquina. Há três módulos da dojot que são utilizados na presente proposta. O primeiro deles é o IoT agent, em que são criados os dispositivos virtuais, responsáveis pelo interfaceamento direto entre dispositivos físicos ou virtuais que estão enviando as informações na borda da rede via protocolo MQTT. O outro módulo é o Flowbroker, onde é montado todo o fluxo dos dados na dojot, desde a recepção da informação por dispositivos virtuais, passando pelo desencadeamento de requisições HTTP para a aplicação *back-end*, até a recepção da resposta da predição, e, depois, há o encaminhamento desta para o módulo de notificação. O terceiro módulo é o módulo de notificação, responsável por receber eventos de notificação, finalizando o ciclo dos dados. O fluxo dos dados na dojot será mais bem descrito na seção 4.3.

3.3. *Back-End*

A integração de aplicações externas com a plataforma dojot pode ser feita via API com auxílio do módulo *Flowbroker*. O módulo Flowbroker permite a criação de fluxos de comunicação com aplicações externas e desencadeamento de eventos programados. Neste trabalho, o *Flowbroker* é utilizado para monitorar os dados recebidos, disparar eventos

para a aplicação *back-end* e lançar notificações. Um dos eventos disparados pelo *Flow-broker* se trata de uma requisição HTTP, feita sempre que a dojot recebe o equivalente a uma semana de dados coletados. O efeito dessa requisição no *back-end* é solicitar para a dojot via API as instâncias de dados que equivalem a uma semana de informações recebidas – nesse caso, os últimos sete dias. Quando a aplicação recebe esses dados, ela, mais uma vez, faz o tratamento destes e os utiliza no modelo de aprendizado de máquina, do inglês Machine Learning, que será discutido adiante. Assim que o modelo de ML recebe os dados como entrada é feita uma predição da quantidade de casos de dengue para a quinta semana à frente. Esse resultado é retornado como resposta da requisição feita da dojot para a aplicação *back-end* e apresentado como notificação na dojot, sendo possível ser disponibilizado para qualquer outra aplicação interessada. A aplicação *back-end* que se executa em um notebook é composta por quatro módulos, sendo um deles responsável apenas pelo armazenamento de dados em *dataSets* locais, enquanto o restante dos módulos tem funções mais ativas e frequentemente troca informações entre si.

O módulo integrador é responsável por toda a comunicação HTTP entre a aplicação *back-end* e qualquer plataforma ou aplicação externa. É por meio dele que a aplicação *back-end* entende as requisições da dojot e encaminha essas requisições aos demais módulos responsáveis por executá-las. O módulo de tratamento de dados se comunica com todos os módulos na aplicação *back-end*, uma vez que as informações sempre precisam de tratamento ao chegarem à aplicação *back-end* e antes de enviarem informações à dojot. O módulo de tratamento de dados ainda pode ser adaptado para extrair, transformar e carregar dados de diferentes fontes no módulo de armazenamento. O módulo de predição é responsável por fazer predição de casos de dengue para a quinta semana à frente. Ele conta com modelos de aprendizado de máquina treinados e testados, aptos a fazerem predições.

4. Resultados

Nesta seção serão apresentados os resultados obtidos com o funcionamento da arquitetura proposta no que diz respeito à coleta de dados e à predição de casos de dengue usando diferentes modelos de ML. A subseção 4.1 faz uma análise dos dados utilizando métodos estatísticos. Na subseção 4.2, os modelos de aprendizado de máquina são avaliados utilizando as métricas MAE e Coeficiente de determinação R^2 para o alvo, com a quinta semana de casos de dengue à frente. Finalmente, os resultados da integração entre a aplicação *back-end* e a plataforma dojot, o fluxo de toda comunicação e o fechamento do ciclo da comunicação através de notificações são apresentados no subseção 4.3.

4.1. Análise dos Dados

Este trabalho utiliza dados de três fontes distintas para a análise. São elas: i) IBGE [IBGE 2021], ii) SINAN [SINAN 2021] e iii) INMET [INMET 2021]. Inicialmente foram selecionados 12 variáveis que segundo trabalhos da literatura [Tavares and Rodrigues 2018] [Valter et al. 2020] teriam uma certa influência na quantidade de casos de arboviroses. A análise a seguir tem dois objetivos complementares: i) identificar valores discrepantes nos dados e, se necessário, removê-los; ii) selecionar variáveis com níveis de correlação significantes com o alvo. A primeira etapa pode impactar na segunda e, posteriormente, na predição de casos de dengue. A tabela 2 mostra os dados utilizados e suas abreviações utilizadas nesta seção.

Tabela 2. Dicionário das variáveis

Descrição	Abreviações
Densidade demográfica	Dens_dem
Acumulado de casos de dengue	Acumulado
População	Populacao
Média móvel simples da velocidade do vento	Vent _vel _7
Média móvel simples da temperatura média	Temp_med_7
Média móvel simples da temperatura miníma	Temp _min_7
Média móvel simples dos Suscetíveis	Suscepti
Média móvel simples do índice de reprodutibilidade	Reproduti
Média móvel simples da precipitação de chuva	Prec _7
Média móvel simples da umidade relativa do ar	Umidade _7
Infectados dengue	Dengue
Infectados dengue acumulado 21 dias	Deng _21
Infectados de dengue média móvel	Deng _7

O gráfico de Box plot ou teste de caixa [Silva et al. 2017] foi utilizado para identificar valores fora da curva. O teste consiste em identificar valores que ultrapassem o limite inferior ou superior do conjunto de dados estudado. Dentro do limite inferior se encontram valores que representam 25% dos menores valores, e estão dentro do limite superior 25% dos maiores valores. Estes são, portanto, os limiares aceitáveis. A figura 2 mostra o gráfico de box plot, no qual se encontram todos os atributos ou variáveis estudadas. Algumas nomenclaturas no gráfico foram abreviadas para não interferirem nas demais. O termo sma em alguns atributos se refere à média móvel simples; o atributo Acumulado se trata do acumulado de casos de dengue desde 2007.

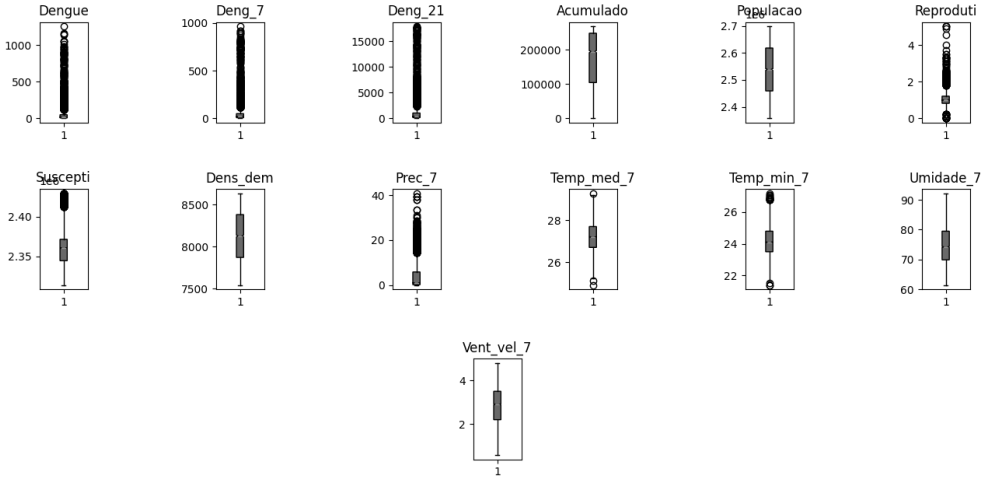
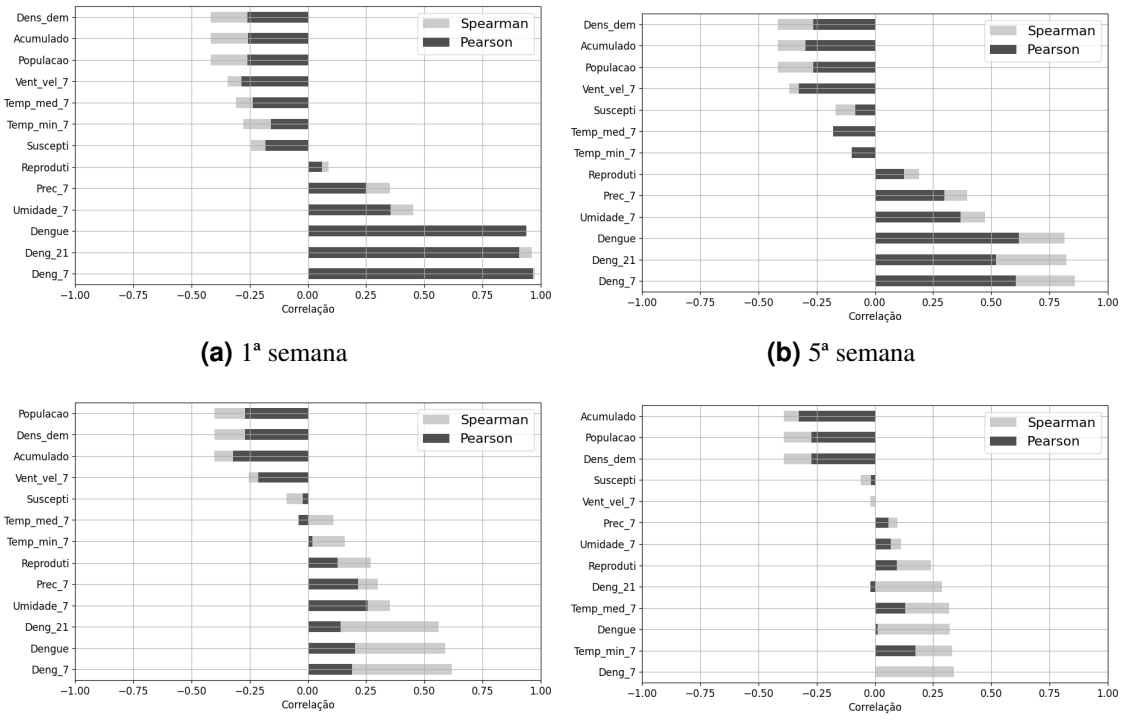


Figura 2. Teste de box plot

A identificação dos atributos com maiores quantidades de valores discrepantes é evidenciada com a visualização. Todos os pontos sobre a barra delimitadora superior ou abaixo da barra inferior são pontos fora da curva. Entendem-se por pontos fora da curva os valores muito distantes da média dos dados. Variáveis com valores bem distribuídos têm comportamento semelhante ao gráfico que representa a população ou velocidade do

vento, em que os dados estão próximos à mediana, representada por um pequeno corte ao meio, na parte mais volumosa do gráfico. Uma forma de aumentar a precisão dos modelos de aprendizado de máquina já bastante convencional é retirar os pontos fora da curva dos dados, entretanto, há um motivo evidente para as variáveis estudadas neste trabalho apresentarem esses valores. Após a retirada dos valores discrepantes dos dados e de testar os modelos de ML, o desempenho dos modelos teve perdas significantes de desempenho, levando em consideração as métricas utilizadas. Esse fator ocorre porque os valores discrepantes são resultado de surtos de dengue ocorridos na cidade de Fortaleza e, portanto, não serão descartados. Há um evidente relacionamento entre as variáveis estudadas e o alvo, os surtos de dengue são resultado de picos nos valores das variáveis estudadas. Isso fica mais evidente no gráfico de correlação de variáveis.

A figura 3 mostra a relação entre os dados de entrada e quatro alvos: casos de dengue para 1ª, 5ª, 10ª e 15ª semanas à frente, consecutivamente. Pode-se observar que, conforme a quantidade de semanas avança, os métodos de correlação diminuem consideravelmente a capacidade de relacionar as variáveis com os alvos. Uma explicação empírica para esse fato está nas variáveis de entrada que representam o estado do ambiente em certo momento; os dias e semanas seguintes irão sofrer outros efeitos e algumas vezes esses efeitos mudarão drasticamente o estado anterior, aumentando, assim, a imprevisibilidade. No estudo de correlações, os valores variam de -1 a 1 e, quanto mais próximo de 1 ou -1, mais forte é a relação entre variável e alvo. Os valores com tendência a -1 indicam que os dados têm relação inversamente proporcionais. Abreviações foram utilizadas nas figuras porque o nome completo das variáveis ultrapassa a limitação de tamanho permitida pelo módulo *python matplotlib* utilizado para gerar as figuras.



(a) 1ª semana **(b)** 5ª semana

(c) 10ª semana **(d)** 15ª semana

Figura 3. Correlação para semanas no futuro

Observa-se, nas figuras, que o método de correlação de *Pearson* tem menos capacidade de correlacionar os dados e tem uma maior perda da correlação com o passar das semanas. Para variáveis ou atributos para os quais o coeficiente de correlação de Spearman mostrou maiores níveis de relacionamento com o alvo, assume-se que essas variáveis não se relacionam com o alvo de forma linear, ou têm um relacionamento monótono [Lira and Neto 2006]. Portanto, neste trabalho, serão considerados os dados que tiveram correlação entre -0.30 a -1 e 0.30 a 1, levando em consideração o coeficiente de correlação de Spearman. Com isso, as variáveis ou atributos descartados foram: índice de reprodutibilidade; temperatura média; temperatura mínima; e variáveis ou atributos suscetíveis.

Neste trabalho, foram treinados e testados cinco modelos de aprendizado de máquina com a mesma metodologia, visando obter o melhor desempenho. Também foi feita a replicação de um modelo treinado e testado no trabalho de [Valter et al. 2020]. Dessa forma, os modelos avaliados foram: i) *Support Vector Regression*; (SVR); ii) *K-Nearest Neighbors*; (KNN); iii) *Extreme Gradient Boosting* (XGBoost); iv) *Long Short-Term Memory* (LSTM); v) *Multilayer Perceptron* MLP/VALTER [Valter et al. 2020]; e vi) *Multilayer Perceptron* MLP/NICODEMOS.

As redes neurais MLP e LSTM foram implementadas com nove camadas, com um total de 711 neurônios. Em [Valter et al. 2020] foram implementadas três camadas - na primeira, contendo 45 neurônios; na segunda, com 45 neurônios; e, na terceira camada, apenas um neurônio. A tabela 3 mostra os hiperparâmetros utilizados em cada modelo.

Tabela 3. Relação de hiperparâmetro por modelo de ML

MODELO	VARIÁVEL	VALOR
SVR	kernel	rbf
	degree	4
	C	1.2
KNN	n-neighbors	6
	metric	euclidean
XGBoost	max-depth	7
	seed	10
	learning-rate	0.2
LSTM MLP/NICODEMOS MLP/VALTER	activation	Relu
	optimizer	Adam
	metrics	mean-absolute-error
	epochs	250
	batch-size	10
	kernel-initializer	he-uniform

4.2. Comparação de modelos de ML

Todos os modelos utilizados nesta seção foram submetidos à mesma base de dados para treino e teste, sendo 80% para treino, 20% para teste. As métricas utilizadas para avaliação de desempenho dos modelos foram Coeficiente de Determinação R^2 e Erro Médio Absoluto, métricas muito utilizadas quando se trabalha com regressão [Vafaei et al. 2018]. Os testes foram feitos levando em consideração apenas previsões de casos de dengue para 1ª, 2ª, 3ª, 4ª e 5ª semanas à frente. Foi possível observar uma tendência: quanto mais semanas

à frente, maior é o erro médio absoluto e menor o coeficiente de determinação R^2; consequentemente, pior é a previsão de todos os modelos. A quantidade de semanas à frente escolhidas também levou em consideração os resultados obtidos no estudo de correlação das variáveis e de alvo, como mostrado na seção 4.1. A seguir, serão apresentados os resultados dos testes de comparação de modelos e uma breve discussão dos resultados.

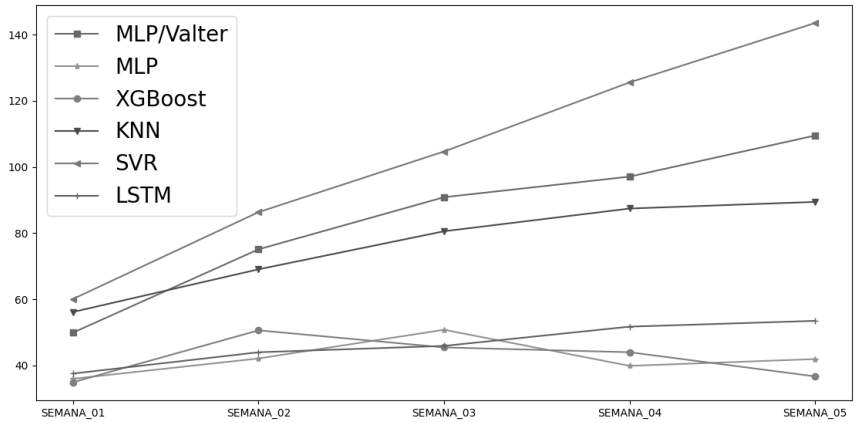


Figura 4. Desempenho dos modelos de ML - MAE

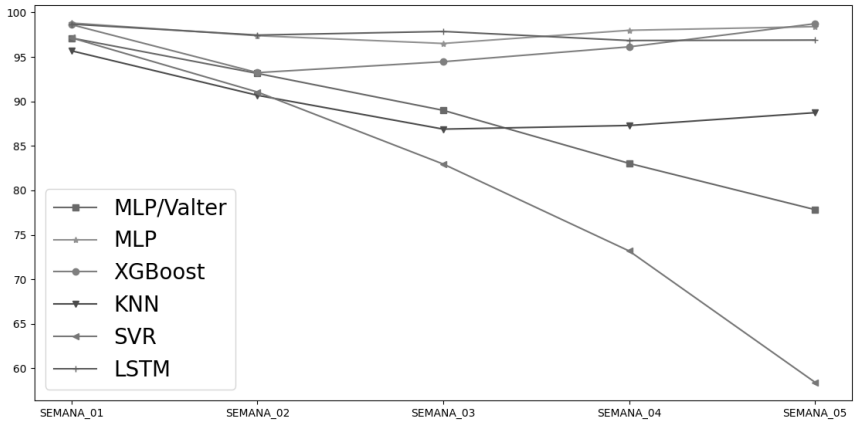


Figura 5. Desempenho dos modelos de ML - R^2

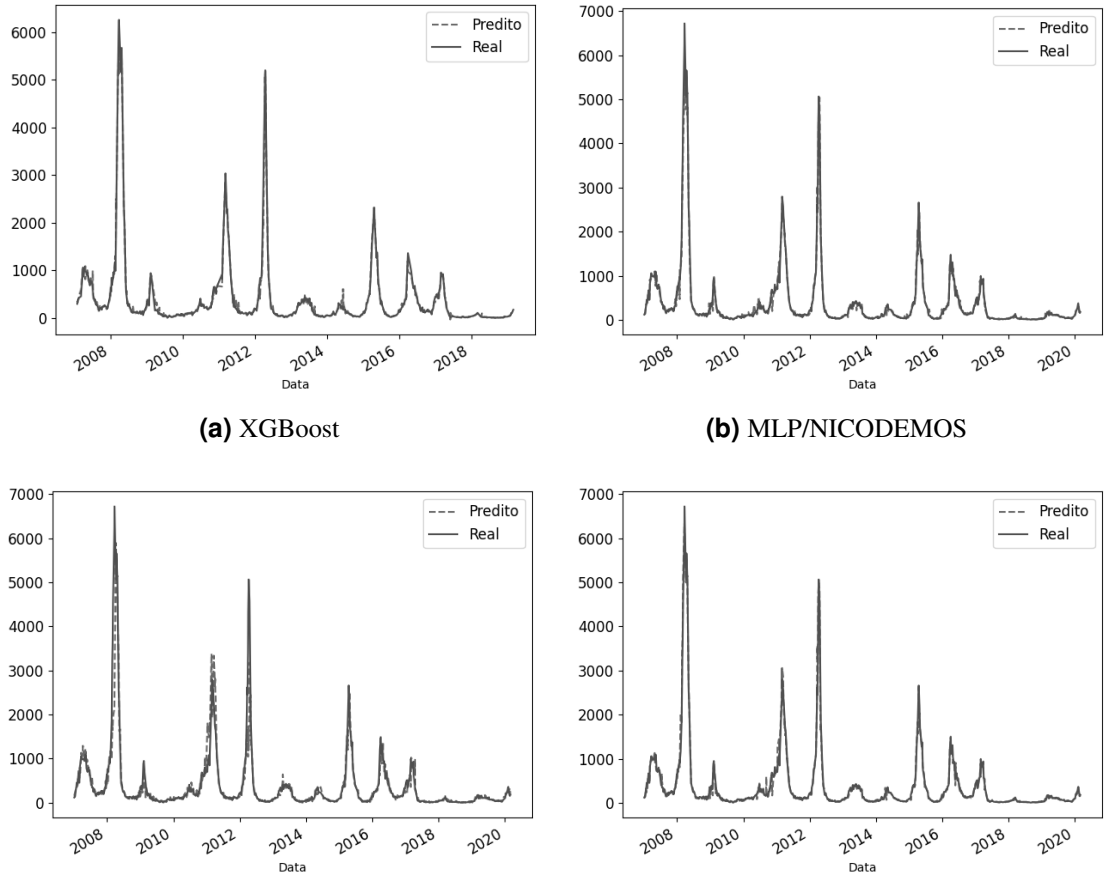
De acordo com a figura 4, o modelo XGBoost teve o menor erro, errando até 36 casos, em média, e levando em consideração o alvo, sendo a quinta semana de casos de dengue no futuro. Isso torna o XGBoost um bom modelo, dada a sua capacidade de predição e simplicidade de implementação, ganhando até de métodos mais sofisticados, como o MLP e o LSTM. Para o objetivo deste trabalho de prever epidemias, um erro de 36 casos de dengue pode ser considerado baixo.

O coeficiente de determinação R^2 mostra, em porcentagem, o quanto o modelo se ajusta aos dados. Em outras palavras, o R^2 explica o quanto o modelo se ajusta a variabilidade dos dados. Devido à forma como as métricas são calculadas, o MAE sofre menos com valores discrepantes se comparado ao coeficiente R^2. Isso ocorre porque no cálculo do R^2, as diferenças entre valores preditos e valores reais são elevadas ao

quadrado. Dependendo do propósito para o qual o modelo for utilizado, pode-se levar em consideração na escolha do modelo uma métrica ou outra. Conforme o propósito deste trabalho, levaram-se em consideração ambas as métricas para complementar a escolha do modelo.

Algo que merece ser observado é que o modelo utilizado no trabalho de [Valter et al. 2020] e replicado no trabalho em questão para comparação, teve desempenho pior que quatro modelos em pelo menos quatro alvos. Durante a fase de treino do modelo MLP/VALTER foi preservada todas as configurações de hiperparâmetros, tal qual se encontra em [Valter et al. 2020]. A única mudança se deu na separação dos dados, que seguiu a mesma divisão do trabalho proposto. Todo estudo de correlações feito sobre os atributos foi aplicado apenas no trabalho em questão.

O baixo desempenho do modelo da literatura frente aos demais pode estar relacionado a vários fatores. É provável que a causa do baixo desempenho do modelo da literatura esteja relacionada à quantidade de camadas ocultas e à quantidade de neurônios em cada camada. As figuras 6 e 7, retratam a precisão de cada modelo de ML diante dos dados de teste. Entre os dados utilizados para teste, o maior valor real registrado foi de 6.724 casos de dengue e a previsão correspondente a esse valor foi feita para a data 24/03/2008, cinco semanas antes.



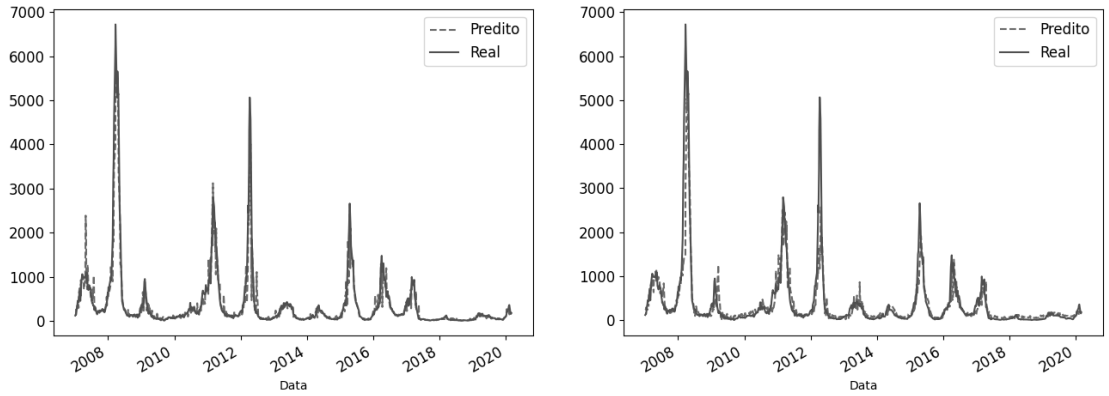
(a) XGBoost

(b) MLP/NICODEMOS

(c) MLP/VALTER

(d) LSTM

Figura 6. Desempenho dos modelos por ano - Dados de Teste



(a) KNN **(b)** SVR

Figura 7. Desempenho dos modelos por ano - Dados de Teste

No eixo X das figuras estão os períodos em anos de casos de dengue estudados, sendo um total de 964 amostras distribuídas entre 2007 e 2020. O eixo Y representa a quantidade de casos de dengue. Observa-se que todos os modelos de uma forma geral tiveram bons resultados. O modelo KNN em particular não obteve um bom desempenho logo em 2007. Visualmente, o KNN teve desempenho semelhante ao dos demais modelos nos demais anos. Todos os modelos conseguiram prever de forma considerável o maior surto em 2008, quando o alvo foi de 6.724 casos de dengue. Isso mostra que os modelos foram bem treinados, testados e, consequentemente, cumprem com o objetivo deste trabalho, que é prever epidemias. A análise visual torna-se interessante por possibilitar a consulta do histórico das predições durante os anos.

4.3. Ciclo da Notificação de Casos

A figura 8 é resultado de todo o ciclo dos dados, do envio dos dados da estação meteorológica, consumo dos dados pela aplicação *back-end* e submissão da predição como notificação para a plataforma dojot. Essa notificação pode ser disponibilizada para outras aplicações interessadas na informação.

Para que os dados brutos cheguem a ser uma informação útil, há um desencadeamento de ações na plataforma dojot, mais especificamente pelo módulo FlowBroker. No FlowBroker, há algumas condições criadas para que os dados de fontes distintas se refiram a mesma data. A primeira condição imposta no fluxo é desencadeada quando a plataforma recebe dados meteorológicos equivalentes a um dia. Como os dados meteorológicos são enviados por hora, então, 24 recebimentos equivalem a um dia. A segunda condição no fluxo se refere aos dados populacionais e de dengue, sendo estes enviados uma vez ao dia. Assim, sete envios equivalem a uma semana. Dessa forma, só se tem dados para fazer uma predição após a segunda condição ser atendida. Há duas requisições HTTP neste fluxo: a primeira delas, indica a aplicação *Back-End*, que já pode enviar os dados populacionais e de dengue referentes à mesma data dos dados meteorológicos para a plataforma dojot. A segunda requisição indica para o *Back-End* que já tem dados para fazer uma predição, então, a aplicação *Back-End* requisita à dojot esses dados, faz a predição e envia o resultado para a dojot exibir como notificação.

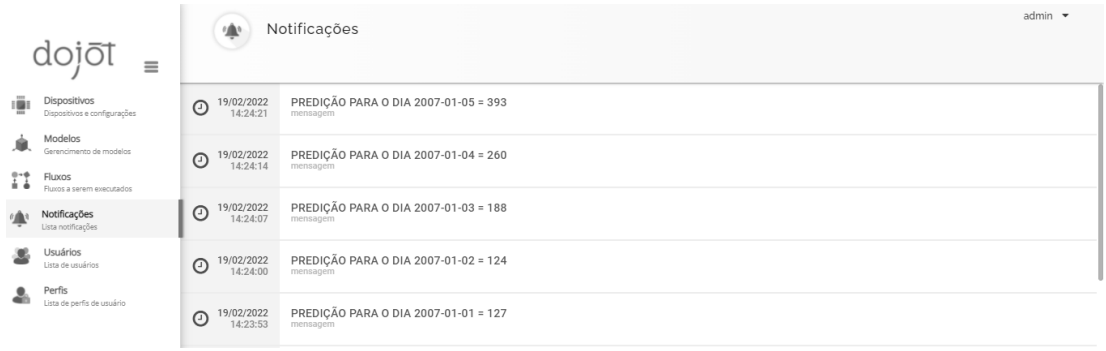


Figura 8. Notificações da dojot

5. Conclusões

Este trabalho apresentou um estudo na área de e-health utilizando Internet das Coisas e aprendizado de máquina. Sua principal contribuição foi a integração entre uma aplicação back-end e uma plataforma de IoT. Durante a realização deste trabalho foi feito um estudo sobre os dados meteorológicos, populacionais e de dengue. Foi feita uma análise de correlação entre variáveis estudadas e casos de dengue para 1ª, 5ª, 10ª e 15ª semanas à frente. Variáveis com níveis significativos de correlações com o alvo foram selecionadas levando em consideração o Coeficiente de correlação de Spearman.

Uma comparação com modelos de aprendizado de máquina foi realizada com o objetivo de selecionar o melhor modelo na fase de teste. Foi possível predizer casos de dengue para a 5ª semana no futuro, com MAE de 36 e coeficiente de determinação R2 de 98%. A plataforma dojot teve uma participação muito significativa no objetivo final deste trabalho. A dojot foi responsável pelo armazenamento, pela disponibilização dos dados e pela notificação de casos de dengue preditos. Para o envio dos dados, uma estação meteorológica foi simulada utilizando uma *Raspberry Pi* e os dados enviados via protocolo MQTT para a plataforma dojot. Os dados meteorológicos, juntamente com dados de agravo de casos de dengue, constituem o conjunto de informações utilizado pelo modelo de *ML* selecionado.

Referências

Carvalho, H., Sá, J., and Farias, F. (2021). Implantação de uma Arquitetura de Software para Monitoramento de Dados Ambientais em um Cenário de Smart Campus. pages 167–170.

CPqD (2021 (acessado em 01 de Fevereiro, 2021)). *dojot documentation*.

IBGE (2021 (acessado em 18 de Março, 2021)). *Projeção da População*.

INMET (2021 (acessado em 19 de Março, 2021)). *MANUAL DE USO DA API ESTAÇÕES E DADOS METEOROLÓGICOS*.

Lira, S. A. and Neto, A. C. (2006). Coeficientes de correlação para variáveis ordinais e dicotômicas derivados do coeficiente linear de pearson. *Ciencia y Engenharia/ Science and Engineering Journal*, 15(October):45–53.

Othman, M. K. and Danuri, M. S. N. M. (2017). Proposed conceptual framework of Dengue Active Surveillance System (DASS) in Malaysia. *ICICTM 2016 - Proceedings*

of the 1st International Conference on Information and Communication Technology, (May):90–96.

Santos, S. C., Firmino, R. M., Mattos, D. M., and Medeiros, D. S. (2020). An IoT Rainfall Monitoring Application based on Wireless Communication Technologies. *2020 4th Conference on Cloud and Internet of Things, CIoT 2020*, pages 53–56.

Sareen, S., Sood, S. K., and Gupta, S. K. (2017). Secure internet of things-based cloud framework to control zika virus outbreak. *International Journal of Technology Assessment in Health Care*, 33(1):11–18.

Silva, M. R., de Moura, F. P., and Jardim, C. H. (2017). O diagrama de caixa (Box Plot) aplicado à análise da distribuição temporal das chuvas em Januária, Belo Horizonte e Sete Lagoas, Minas Gerais-Brasil. *Revista Brasileira de Geografia Física*, 10:023–040.

SINAN (2021 (acessado em 19 de Março, 2021)). *Sinan Dengue/Chikungunya*.

Singh, S., Bansal, A., Sandhu, R., and Sidhu, J. (2018). Fog computing and IoT based healthcare support service for dengue fever. *International Journal of Pervasive Computing and Communications*, 14(2):197–207.

Tavares, P. D. S. and Rodrigues, E. B. (2018). IoT-Based Architecture for Data Analytics of Arboviruses in Smart Cities. *Proceedings - IEEE Symposium on Computers and Communications*, 2018-June(Mdcc):952–957.

Vafaei, S., Soosani, J., Adeli, K., Fadaei, H., Naghavi, H., Pham, T. D., and Bui, D. T. (2018). Improving accuracy estimation of Forest Aboveground Biomass based on incorporation of ALOS-2 PALSAR-2 and Sentinel-2A imagery and machine learning: A case study of the Hyrcanian forest area (Iran). *Remote Sensing*, 10(2).

Valter, R., Oliveira, M., Silva, F. G. S., and Andrade, D. (2020). Intelligent Epidemiological Surveillance in the Brazilian Semiarid.

Varela, V. and Vívian (2016). Rastreamento endêmico da dengue, zika e chikungunya via Android e sistema de informação geográfica (SIG). page 55.

Xavier, R. F. (2020). Desenvolvimento de uma Aplicação de Monitoramento Utilizando a Plataforma Iot Dojot.

Zhao, N., Charland, K., Carabali, M., Nsoesie, E. O., Maheu-Giroux, M., Rees, E., Yuan, M., Balaguera, C. G., Ramirez, G. J., and Zinszer, K. (2020). Machine learning and dengue forecasting: Comparing random forests and artificial neural networks for predicting dengue burden at national and sub-national scales in Colombia. *PLoS Neglected Tropical Diseases*, 14(9):1–16.