

Sensor de Detecção de Incêndios Florestais com LoRa e NodeRED

Igor L.R. Azevedo * Luiz Augusto F.D. Costa **

* 16/0124981, Engenharia Elétrica, UnB, igorlima1740@gmail.com

** 18/0035291, Engenharia Elétrica, UnB, luiz.fidalgo@hotmail.com

Abstract: Assist in the logistics of fighting forest fires and improve the detection and notification system of responsible entities.

Resumo: auxiliar na logística do combate a incêndios florestais e melhorar o sistema de detecção e notificação das entidades responsáveis por combater incêndios.

Internet das coisas já é implementado em diversas tarefas na indústria e no dia a dia das pessoas. Esse projeto consiste em aplicar internet das coisas, mais precisamente LoRa e NodeRED, para detecção de incêndios. Assim, quatro circuitos foram montados sendo que 3 deles funcionam como medidores de incêndio e um como Gateway, o qual está conectado em uma rede WI-FI, para notificação de entidades responsáveis pelo combate ao incêndio.

Keywords: Fire; Forest; IoT; LoRa; NodeRED.

Palavras-chaves: Incêndio; Floresta; IoT; LoRa; NodeRED.

1. INTRODUÇÃO

Com o aumento da população, mudanças climáticas e uma falta de políticas públicas de combate à incêndios florestais é esperado que ocorra, cada vez mais, incêndios em unidades de conservação (UC). Sendo unidade de conservação, um espaço territorial o qual possui seus recursos ambientais, incluindo águas jurisdicionais, legalmente instituído pelo poder público. *Gov.Federal (2000)*

Incêndios florestais são uma preocupação devido a sua capacidade de danificar o solo e comprometer seriamente seu ciclo de funcionamento em um ecossistema. Na maioria das unidades de conservação, não existe um planejamento adequado visando à prevenção e ao combate a incêndios *Bontempo et al. (2011)*.

Há diversos fatores que influenciam na propagação dos incêndios como material combustível (quantidade, umidade, tipo e arranjo), condições climáticas, topografia e o tipo de floresta. Dessa forma, devido a fatores incertos e altamente variáveis, torna-se difícil a elaboração de políticas públicas e abordagens para a prevenção e combate de incêndios florestais.

Além disso, unidades de conservação geralmente estão localizadas em áreas afastadas dos centros urbanos dificultando uma detecção de incêndios rápida e precisa. Consequentemente, é comum que as autoridades responsáveis por apagar os incêndios sejam notificadas quando o mesmo já tomou grandes proporções.

Por conseguinte, o conhecimento do perfil dos incêndios florestais nas unidades de conservação e da **logística** associada ao seu combate é fundamental para se traçarem estratégias eficientes para o planejamento de ações de prevenção e combate a incêndios. *Gumercindo Souza Lima1 (2017)*

O Brasil, nos últimos anos, com a falta de recursos e uma clara falta de prioridade por parte do governo federal vem sofrendo grandes perdas em suas unidades de conservação devido as queimadas. Com a falta de uma política pública consolidada ao combate de incêndios florestais, é provável que incêndios no país sejam cada vez mais comuns ameaçando assim a fauna e flora do Brasil.

Diante do exposto, e visando em como auxiliar na logística do combate a incêndios florestais e melhorar o sistema de detecção e notificação das entidades responsáveis por combater incêndios, o objetivo desse trabalho foi desenvolver um sistema capaz de mapear unidades de conservação e assim detectar a ocorrência de incêndios.

2. TRABALHOS RELACIONADOS

Nessa seção, será apresentada outras referências de projetos que realizaram a detecção de incêndios através de Hardware e internet das coisas.

Em *Divya et al. (2019)*, foi utilizado um sensor de fogo, um sensor de fumaça e um sensor de temperatura para detecção do incêndio. Para a comunicação foi utilizado WI-FI. A ideia do projeto consiste em destacar a "feature" wi-

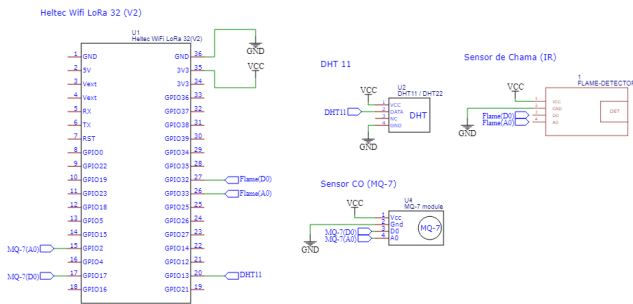


Figura 1. Esquemático do sensor.

reless sensor networks (WSN) como uma possível solução para a detecção precoce de qualquer ameaça de incêndio.

Em *Neumann et al. (2018)*, foi utilizado sensores de temperatura e de umidade para detecção de variação abrupta em seus valores devido a um incêndio. Para a comunicação, foi utilizado dispositivos com bluetooth com um alcance de 100 metros, os quais se comunicam com *smartphones* de pessoas próximas. E em seguida, esses *smartphones* enviam os dados para o "Smart Forest Server" e assim entidades responsáveis são notificadas em caso de incêndio.

Em *Vega-Rodríguez et al. (2019)*, foi utilizado um sensor de CO_2 um motor para a detecção da velocidade dos ventos e um sensor *DHT11* para a detecção de temperatura. O protocolo utilizado para comunicação foi o *LoRa* e foi utilizado também a estrutura de Gateway para comunicação com um servidor. O servidor se comunica com uma aplicação web que disponibiliza todos os dados dos sensores.

3. SENSOR DE DETECÇÃO DE INCÊNDIOS FLORESTAIS

De modo a permitir a detecção de incêndios florestais, desenvolveu-se um dispositivo de baixo custo, baixo consumo de energia, com comunicação sem fio de longo alcance e que conta com sensores específicos para analisar as condições ambientais locais.

3.1 Hardware

O sistema é composto por dois hardwares: sensores e gateway. Ambos utilizam o mesmo modelo de módulo responsável pela comunicação e processamento das informações, o Heltec WiFi LoRa 32 (V2), uma placa de desenvolvimento que integra um SoC ESP32 e um chip SX1278 LoRa.

O primeiro hardware possui, além do módulo Heltec, três sensores responsáveis pela coleta de dados do ambiente, eles são: sensor de temperatura e umidade DHT11, sensor de chamas KY-026 e sensor de monóxido de carbono MQ-7. O esquemático deste hardware é indicado na Figura 1 e uma foto de sua implementação é indicada na Figura 2.

3.2 Custo de implementação do hardware

Além disso, no que diz respeito a seu baixo custo, o valor, no Brasil, de cada componente utilizado pode ser identificado na Tabela 1.

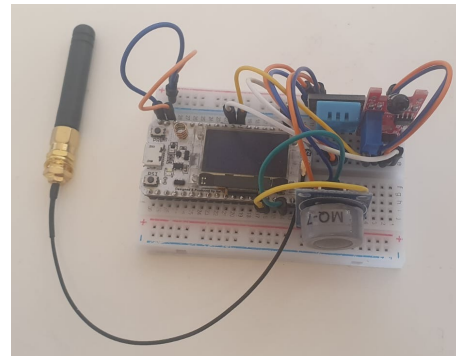


Figura 2. Hardware do sensor.

Item	Valor
Heltec WiFi LoRa 32 (V2)	R\$ 180,00
DHT11	R\$ 14,00
KY-026	R\$ 11,25
MQ-7	R\$ 20,55

Tabela 1. Valor dos componentes

Componente	Corrente modo	Corrente modo
	ativo (mA)	sleep (uA)
Heltec WiFi LoRa 32 (V2)	160	800
ESP32	120	800
SX1278	10.8	0.2
DHT11	0.3	60
KY-026	7	-
MQ-7	130	-

Tabela 2. Corrente consumida pelos componentes

Sendo assim, o valor total do gateway seria na faixa de R\$ 180,00, enquanto os sensores custam em média R\$ 225,80. Neste caso não está sendo considerada a alimentação dos dispositivos, mas apenas seus componentes de processamento, comunicação e comando.

Entretanto, é possível a substituição da placa de desenvolvimento Heltec por módulos independentes que realizem a comunicação WiFi e LoRa. Sendo assim, é possível adquirir, no Brasil, um chip ESP32 por cerca de R\$ 36,00 e um módulo SX1278 LoRa por aproximadamente R\$ 40,00. Logo, o valor do gateway poderia ser reduzido para próximo de R\$ 76,00, enquanto os sensores ficariam na faixa de R\$ 121,80.

Em acréscimo, caso este sistema fosse produzido em maior escala, opções de importação de componentes ainda aumentariam a viabilidade econômica da tecnologia.

3.3 Alimentação do circuito

De modo a permitir que o sensor seja utilizado de forma independente e remota, realizou-se um estudo de seu consumo energético e de possibilidades de fornecimento de energia.

A corrente de operação de cada componente do circuito encontra-se apresentada na Tabela 2.

É válido ressaltar alguns comentários com relação ao consumo destes componentes:

- o ESP32 consome a corrente de 120 mA quando está realizando a transmissão ou recebimento de algum dado através de WiFi. Caso o módulo esteja sendo usado apenas para processamento e aferição dos dados dos sensores, como é o caso do hardware do sensor, seu consumo pode ser considerado como em modo sleep, ou seja, 800uA.
- O módulo MQ-7 precisa aquecer uma membrana interna para realizar a aferição da concentração de CO, um processo que leva cerca de 60 segundos. Dessa forma, seu consumo durante este período é bastante elevado. Entretanto, ele pode ser desligado enquanto não estiver sendo utilizado.

Sendo assim, para o cálculo do consumo do sensor devem ser considerados seus dois estados: quando este está esperando um intervalo de aferição e quando está realizando a aferição.

Enquanto espera o intervalo de aferição, todos os seus componentes, exceto o módulo LoRa, estão em modo sleep. Logo, sua corrente será aproximadamente 12 mA.

Já enquanto está realizando a aferição, todos os componentes, exceto o ESP32, passam para o modo ativo, consumindo uma corrente de aproximadamente 150 mA.

Considerando que os sensores são aferidos a cada cinco minutos, e que esta aferição e o envio da mensagem pela rede LoRa dure cerca de 60 segundos (pior caso, considerando o aquecimento da membrana do módulo MQ-7), a corrente média do circuito pode ser calculada a partir da Equação 1.

$$I_{media} \simeq \frac{60s}{5min} 150mA + \frac{5min - 60s}{5min} 12mA = 39,6mA \quad (1)$$

Dessa forma, utilizando uma bateria de Li-Ion modelo 18650 de 3,7 V e 2.200 mAh, o tempo de operação do sensor seria dado pela Equação 2.

$$t_{Operacao} \simeq \frac{P_{bateria}}{P_{media}} = \frac{3.7V * 2.200mAh}{3.3V * 39,6mA} = 62,3h \quad (2)$$

Assim, a bateria seria capaz de alimentar o sensor durante cerca de 62,3 h, ou cerca de 2,6 dias.

Então, uma alternativa de carregamento da bateria seria o uso de painéis fotovoltaicos. Para dimensioná-lo, utilizaram-se os dados da plataforma Solar Electricity Handbook ¹ e obtiveram-se os valores de irradiância apresentados na Figura 3.

Considerando o mês com menor irradiância em Brasília, dezembro, temos uma média de 4,91 kWh/m²/dia. Sendo assim, a potência média pode ser calculada a partir da Equação 3.

$$Pot_{media} = \frac{4910Wh/m^2}{24h} = 204,6W/m^2 \quad (3)$$

Então, considerando uma eficiência de 10% dos painéis fotovoltaicos, a potência elétrica média será aproximada-

¹ <http://www.solarelectricityhandbook.com/solar-irradiance.html>

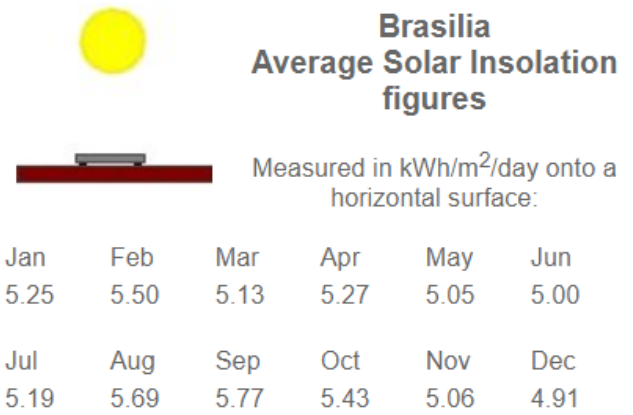


Figura 3. Irradiância média mensal em Brasília.

```
//Lê dados dos sensores
//Sensor CO
int COsensorAnalog = co.readAnalog();
int COsensorDigital = co.readDigital();
//Sensor chama
int firesensorAnalog = flame.readAnalog();
int firesensorDigital = flame.readDigital();
//Sensor temperatura e umidade
float h = dht.readHumidity();
float t = dht.readTemperature(); // Read temperature as Celsius (the default)
float f = dht.readTemperature(true); // Read temperature as Fahrenheit (isFahrenheit = true)
//Heat index (Sensação Térmica)
float hif = dht.computeHeatIndex(f, h); // Compute heat index in Fahrenheit (the default)
float hic = dht.computeHeatIndex(t, h, false); // Compute heat index in Celsius (isFahrenheit = false)
```

Figura 4. Funções de leitura dos sensores.

mente 20,5 W/m². Logo, sabendo que o sensor é alimentado por uma tensão de 3.3V e que sua corrente média é de 39,6 mA, pode-se calcular a área do painel fotovoltaico a ser utilizado a partir da Equação 4.

$$Area \simeq \frac{3.3V * 39,6mA}{20,5W/m^2} = 63,75cm^2 \quad (4)$$

Dessa forma, o sensor pode ser alimentado utilizando uma bateria recarregável de Li-Ion modelo 18650 e um painel fotovoltaico com mais de 63,75 cm², tendo uma autonomia de até 2,6 dias sem luz solar, como dias chuvosos ou nublados. Vale ressaltar que esta é apenas uma estimativa, sendo ainda necessário projetar os componentes do circuito de alimentação dos sensores para garantir sua eficiência e segurança.

3.4 Software

Cada hardware do sistema possui um software próprio, ambos desenvolvidos em linguagem C utilizando a plataforma Arduino IDE. O software dos sensores é responsável por fazer a leitura de seus periféricos, apresentá-la em uma tela OLED e enviá-la através da rede LoRa Mesh em um intervalo de tempo definido, no caso do projeto, a cada 2 segundos. Já o software do gateway é responsável por receber as mensagens enviadas através da rede LoRa e encaminhá-las a um broker MQTT.

A leitura dos dados dos sensores é registrada e armazenada a partir das funções apresentadas na Figura 4.

Em seguida, os dados são formatados e enviados através da rede LoRa Mesh, conforme indicado na Figura 5.

No caso do gateway, seu software é iniciado com o estabelecimento de sua conexão com uma rede WiFi e com o broker MQTT, como indicado na Figura 6.

```
String mensagem = lora.formatMessage(h, t, f, hic, hif, COsensorAnalog, COsensorDigital, firesensorAnalog, firesensorDigital);
oled.displayReadings(t, h, hic, COsensorAnalog, firesensorAnalog);
msgCount = lora.sendMessage(msgCount, mensagem, destination);
```

Figura 5. Funções de envio dos dados dos sensores.

```
void setup()
{
  Heltec.begin(true, true, true, true, BAND);
  wifiSensor.setup_wifi(ssid, password);
  wifiSensor.setup_mqtt(mqtt_server);
}
```

Figura 6. Configuração da conexão com a rede WiFi e com o broker MQTT.

```
void loop()
{
  wifiSensor.loop_connect();
  lora.onReceive(LoRa.parsePacket());
}
```

Figura 7. Rotina de verificação de conectividade e encaminhamento de mensagens do gateway.

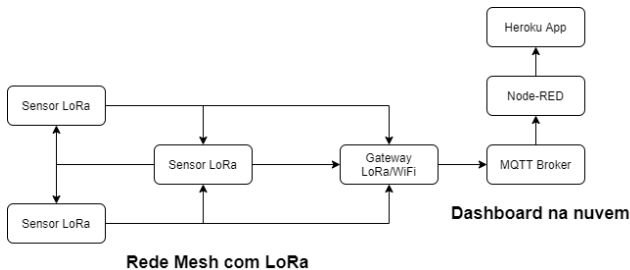


Figura 8. Protocolos de comunicação.

Então, aplica-se um loop responsável pela verificação da conexão do módulo com a rede WiFi, evitando perdas de conectividade, e pelo envio das mensagens para o broker MQTT caso estas sejam recebidas através do módulo LoRa, como indicado na Figura 7.

3.5 Diagrama de comunicação

O diagrama de comunicação do sistema encontra-se indicado na Figura 8.

A comunicação entre os sensores e entre sensores e gateway é realizada a partir de um protocolo próprio que implementa uma rede Mesh usando a tecnologia LoRa.

Em seguida, as mensagens recebidas pelo gateway são interpretadas e publicadas em um broker MQTT, no caso do projeto, utilizou-se o broker EMQX.

Os tópicos publicados pelo gateway são visualizados a partir de um fluxo desenvolvido na plataforma Node-RED, a qual está hospedada na plataforma em nuvem Heroku, que permite que os dados sejam acessados remotamente.

4. VISUALIZAÇÃO E USO DOS DADOS

4.1 Dashboard com dados em tempo real e dados históricos

Com os dados obtidos advindo dos sensores espalhados pela floresta, torna-se necessário enviar os mesmos para

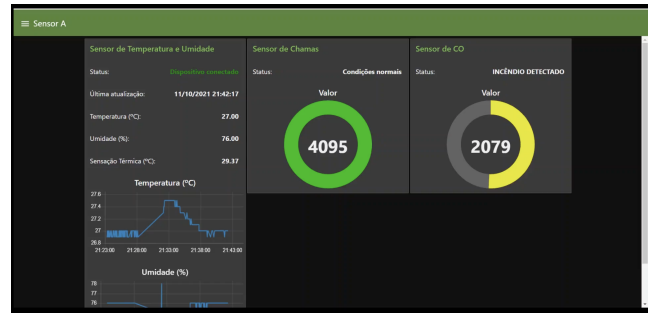


Figura 9. Visualização das informações dos sensores.

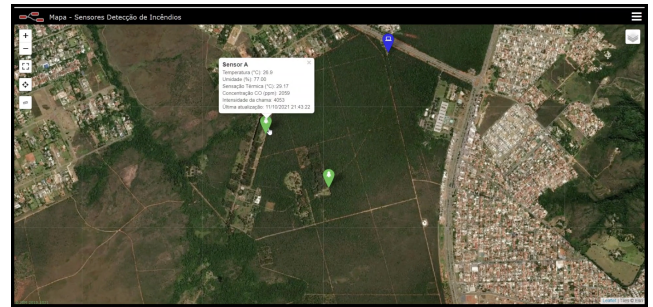


Figura 10. Mapa com a posição de cada sensor.

uma aplicação web. Assim, os valores dos sensores podem ser monitorados e em caso de variação abrupta - incêndio - as entidades responsáveis serem notificadas. Para o desenvolvimento de tal aplicação, foi utilizada a ferramenta Node-RED².

A figura 9 apresenta a tela da aplicação web de um dos sensores espalhados pela floresta. Nessa figura, é possível visualizar a temperatura, a umidade, a sensação térmica, o valor do sensor de CO e o valor do sensor de chamas. É exibido também o histórico de temperatura e umidade. Além disso, é indicado se há ou não a ocorrência de incêndio.

É importante destacar que o dashboard apresentado na figura 9 é para um denominado sensor A. Tais informações podem ser visualizadas para qualquer sensor.

4.2 Representação geográfica

Para uma eficiente atuação no combate a incêndios florestais mais uma informação é necessária, a localização. Sabendo se há ou não um incêndio, é fundamental enviar as geo coordenadas do sensor para as entidades responsáveis.

Dessa forma, com o objetivo de localizar cada sensor geograficamente foi adicionado à aplicação web, um mapa no qual é possível visualizar a localização de cada sensor e suas respectivas aferições. Tal funcionalidade da aplicação web pode ser visualizada na figura 10.

5. VALIDAÇÃO EXPERIMENTAL

De modo a avaliar a efetividade do sistema para detecção de incêndios florestais, realizaram-se testes em ambiente controlado para observar os dados coletados pelos sensores e o alcance da comunicação.

² <https://sensor-incendios-florestais.herokuapp.com/ui>



Figura 11. Teste de alcance da comunicação.

5.1 Teste de alcance

Realizou-se um teste de alcance dos módulos LoRa, trabalhando em 915MHz. Em um ambiente residencial, foi possível manter a comunicação a uma distância de cerca de 700m, como indicado na Figura 11.

5.2 Testes dos dados coletados

Com relação aos dados coletados pelos sensores, realizaram-se testes qualitativos de suas medições.

Inicialmente realizou-se um teste com o sensor DHT11. Apesar deste apresentar certa instabilidade para a leitura de seus dados, disponibilizando a informação "nan" em seus campos em alguns momentos, e de aferir valores incoerentes com a realidade (como uma umidade superior a 80% em Brasília durante o período de seca), este apresentava um comportamento coerente quando aproximado de superfícies mais quentes e úmidas, ao elevar seus valores de temperatura e umidade, e vice-versa.

No caso do sensor KY-026, seus resultados também foram condizentes com a realidade, ao variar seus valores analógicos dependendo da intensidade de luz infravermelho que o atingia. Entretanto, é importante ressaltar que este módulo trabalha de forma inversa, ou seja, quando submetido à uma intensidade alta de infravermelho, seu valor analógico irá reduzir.

Por fim, o sensor MQ-07 também correspondeu ao esperado, ao elevar seus valores analógicos quando aproximado de regiões com maior concentração de monóxido de carbono (CO).

Entretanto, não foram realizadas análises quantitativas relacionadas à precisão dos sensores utilizados, o que pode ser aplicado em trabalhos futuros.

5.3 Testes do Broker MQTT

Inicialmente utilizou-se o broker MQTT Mosquitto para publicação e visualização dos dados enviados pelos sensores. Entretanto, este apresentou instabilidade para conexão e leitura dos dados em diversos momentos.

Dessa forma, optou-se por buscar outros brokers, como, por exemplo o EMQX, o qual apresentou bons resultados de estabilidade.

5.4 Testes da plataforma Heroku

A plataforma Heroku oferece gratuitamente o serviço de hospedagem da aplicação do Node-RED, além de permitir

o uso de todas as funcionalidades comuns da ferramenta. Entretanto, por se tratar de uma versão gratuita, ela não garante o funcionamento contínuo da aplicação e, com uma periodicidade de cerca de 1 a 2 dias, a aplicação é reiniciada, o que gera a perda do histórico de dados.

Sendo assim, em trabalhos futuros, deseja-se relacionar um banco de dados à aplicação do Node-RED de forma a armazenar os dados obtidos dos sensores para manutenção dos gráficos históricos da aplicação.

6. CONCLUSÃO

A partir do projeto, foi possível desenvolver um sistema de detecção de incêndios florestais com baixo custo de implementação e manutenção e com alto potencial de escalabilidade. Isto se dá pelo uso de módulos e sensores simples e de fácil acesso, além da aplicação de protocolos de comunicação de longo alcance utilizando redes Mesh.

Deseja-se, em trabalhos futuros, realizar testes quantitativos relacionados à precisão e confiabilidade das medições dos sensores, além de testes do alcance do sensor em diferentes ambientes. E, como forma de armazenar os dados coletados, relacionar a plataforma Node-RED com um banco de dados remoto.

Também, é de interesse dos pesquisadores consultar outros profissionais da área de proteção ambiental e florestal, de modo a definir estratégias concretas de uso e teste destes sensores em situações reais, além de complementar este estudo com o planejamento de formas de combate à incêndios que possam ser detectados.

REFERÊNCIAS

- Bontempo, G., Lima, G., Ribeiro, G., Doula, S.M., and Jacovine, L. (2011). Registro de ocorrência de incêndio (roi): evolução, desafios e recomendações. 0.
- Divya, A., Kavithanjali, T., and Dharshini, P. (2019). Iot enabled forest fire detection and early warning system. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 1–5. doi:10.1109/ICSCAN.2019.8878808.
- Gov.Federal (2000). Sistema nacional de unidades de conservação da natureza. URL http://www.planalto.gov.br/ccivil_03/leis/19985.htm.
- Gumercindo Souza Lima1, Fillipe Tamiozzo Pereira Torres1, A.d.G.C.G.d.A.F.M.R.d.S.J. (2017). avaliação da eficiência de combate aos incêndios florestais em unidades de conservação brasileiras. 0.
- Neumann, G.B., Almeida, V.P.d., and Endler, M. (2018). Smart forests: fire detection service. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, 01276–01279. doi:10.1109/ISCC.2018.8538719.
- Vega-Rodríguez, R., Sendra, S., Lloret, J., Romero-Díaz, P., and Garcia-Navas, J.L. (2019). Low cost lora based network for forest fire detection. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 177–184. doi:10.1109/IOTSMS48152.2019.8939193.