Deteção de Intrusões em Redes LoRaWAN

Filipe Costa[1], Nuno Cruz[1,2][0000−0001−8570−8670], and José Simão[1,3][0000−0002−6564−593X]

[1] FIT - Future Internet Technologies, ISEL - Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa
[2] LASIGE, Faculdade de Ciências, Universidade de Lisboa
[3] INESC-ID Lisboa

Resumo. Atualmente os dispositivos de IoT são utilizados em diversos tipos de aplicações, desde sistemas domésticos até aplicações para apoiar na gestão de cidades (i.e., *smart cities*). Estes dispositivos são normalmente sensores que recolhem diferentes tipos de dados em condições exigentes, tais como dificuldade de ligação direta à Internet ou a ausência de fontes estáveis de energia. Nos últimos anos, o protocolo LoRaWAN para redes LPWAN (*Low Power Wide Area Network*) destaca-se pela sua arquitetura aberta e distribuída a qual introduz novas ameaças de segurança.
Neste artigo é proposto o uso de algoritmos de aprendizagem automática num Sistema de Deteção de Intrusões (IDS) para redes LoRaWAN. Usando o algoritmo *K-Nearest Neighbors* (KNN), é feita uma análise comportamental da rede para detetar potenciais intrusões, tendo em conta o padrão de pacotes anteriormente observado. Usando o IDS Sucirata, e através da linguagem de programação Lua e Python, foram acrescentadas regras complexas as quais, tendo em conta modelos previamente construídos, indicam a confiança de um pacote ser ou não uma intrusão. Foram utilizados dados do tráfego de sensores localizados na cidade de Lisboa para a criação e aprendizagem do modelo. Os resultados mostram que foi possível analisar os pacotes recebidos, escrever os parâmetros relevantes na base de dados e reconhecer quando é que está a existir uma intrusão ou se é o comportamento esperado pelo dispositivo.

Palavras-chave: IoT · LoRaWAN · Sistema de Deteção de Intrusões · Aprendizagem Automática

1 Introdução

A utilização de dispositivos de IoT e protocolos de comunicação sem fios, aumentou o perímetro de segurança das redes e, consequentemente, introduziu novas vulnerabilidades. É essencial, para além da implementação de mecanismos de segurança no perímetro da rede, a deteção de intrusões o mais rápido possível, distinguindo o tráfego normal de uma intrusão e alertar a infraestrutura quando necessário.

Dos vários protocolos de comunicações existentes em IoT, como o caso de NB-IoT [5], LTE-M [11], SigFox [6], Zigbee [12], optou-se por abordar neste artigo o

protocolo LoRaWAN. Este tem como principais vantagens o seu longo alcance (mais do que 15 km em determinados cenários), baixo consumo nos dispositivos (podendo atingir os 10 anos de bateria), alta capacidade de rede (até 1 milhão de dispositivos) e a utilização de espetro que não necessita de licenciamento. Estas fazem com que seja um protocolo bastante robusto e amplamente utilizado.

Tendo em consideração que o LoRaWAN é um dos protocolos mais empregue para a ligação dos dispositivos IoT com a rede, faz com que seja mais desejável a sua exploração de vulnerabilidades, levando posteriormente a ataques indesejados ao sistema.

Neste contexto, pretende-se que o Sistema de Deteção de Intrusões (IDS) analise apenas o tráfego na rede, detete intrusões, caso sejam identificadas como tal, e, por sua vez, notifique o mais celeremente possível a infraestrutura, não adicionando qualquer latência às comunicações existentes.

Para testar a validade da solução proposta, recorreu-se à linguagem de programação Python para implementar o modelo de aprendizagem supervisionada, usando o algoritmo *K-Nearest Neighbors* (KNN), treinando primeiro com um conjunto de pacotes e verificando depois com um conjunto diferentes por possíveis intrusões. Os resultados mostram que a utilização do modelo para classificar novos pacotes resulta numa alta taxa de verdadeiros positivos.

A Secção 2 apresenta alguns conceitos base e apresenta trabalhos relacionados nesta área. Na Secção 3 é discutida a arquitectura do sistema e na Secção 4 as escolhas feitas para no seu desenvolvimento. A Secção 5 analisa o desempenho do modelo e a Secção 6 apresenta as principais conclusões e o trabalho futuro.

2 Conceitos e trabalho relacionado

Esta secção apresenta uma descrição dos principais conceitos relacionados com o artigo, o contexto e o enquadramento tecnológico em que está inserido bem como algum trabalho relacionado.

2.1 Sistemas de Deteção de Intrusões - IDS

Um IDS é um dispositivo ou software que está em constante monitorização da rede ou sistema, analisando assim todos os pacotes que lhe chegam à procura de qualquer intrusão ou violação das politicas de segurança associadas [2, 7]. O IDS é passivo nas comunicações do sistema, ou seja, não adiciona qualquer latência com a sua análise dos pacotes. No entanto não tem a capacidade de intervir diretamente no sistema de forma a prevenir ataques.

O dispositivo capaz de realizar a função de prevenção é o Sistema de Prevenção de Intrusões (IPS). Este é inserido nas comunicações entre uma determinada origem e destino. Ao estar fisicamente presente no percurso das comunicações, vai adicionar mais latência a estas, no entanto vai ser capaz de terminá-las. Este necessita de um maior cuidado na sua análise do tráfego porque podem vir a existir muitos falsos positivos e comunicações legitimas poderão vir a ser interpretadas como intrusões.

Existem duas técnicas principais utilizadas para a deteção de intrusões:

– IDS baseado em assinatura (SIDS)
– IDS baseado em anomalia (AIDS)

A primeira é a forma mais simples e é realizada baseando-se em assinaturas conhecidas dos ataques. Caso um pacote tenha uma parte do seu conteúdo da mesma forma que a assinatura, é gerado um alerta. De forma a ser percetível o conceito de assinatura num IDS, a Figura 1) apresenta um exemplo de uma assinatura simples:

```
alert icmp $HOME_NET any -> $HOME_NET any (msg:"PING ICMP"; itype:8;
    classtype:not-suspicious; sid:1; rev:1;)
```

Fig. 1. Estrutura de uma assinatura apenas com o contexto do pacote ICMP

Na regra apresentada, um pacote ICMP que for enviado com o endereço de origem e destino presente na lista "$HOME-NET", com qualquer porto de origem e destino, e com o icmp-type = 8 (Echo) (ou seja, quando se realiza um ping para um dispositivo), o IDS vai gerar um alerta para o sistema com a mensagem: "PING ICMP". Nos capítulos posteriores irá ser realizada uma análise mais pormenorizada das assinaturas. A utilização de SIDS faz com que seja bastante difícil a deteção de ataques conhecidos como "zero-day", ou seja ataques ainda não conhecidos pela base de dados de assinaturas.

A segunda técnica (AIDS) é baseada não só em assinaturas como na análise comportamental dos dispositivos no sistema, ou seja, guarda numa base de dados o comportamento ao longo do tempo e caso este se altere é gerado um alerta. Desta maneira é possível detetar novos ataques, mesmo que estes não estejam ainda identificados com uma assinatura. Consequentemente, podem vir a ser gerados muitos falsos positivos se o IDS não estiver bem dimensionado [7]. É nesta técnica que se encontram os algoritmos de aprendizagem automática os quais decidem sobre novos pacotes com base no treino realizado anteriormente, conseguindo assim detetar anomalias que ainda não tinham sido explicitamente identificadas.

2.2 LoRaWAN

LoRaWAN é um protocolo concebido para redes de longa distância. Permite que dispositivos de baixa potência comuniquem com aplicações na Internet [9], especifica o formato dos pacotes e define a forma como a rede tem de processar os pacotes. São identificadas três classes de dispositivos, cada uma apropriada para o tipo de dados a enviar.

A arquitetura LoRaWAN tem na sua camada física o protocolo LoRa, o qual opera em bandas de frequência não licenciadas, o que faz da sua utilização, em termos monetários, algo mais em conta em comparação com protocolos que tiram proveito de bandas licenciadas. No caso da União Europeia é possível utilizar as frequências 868 e 433 MHz, sendo a primeira a mais utilizada. A Figura 2) ilustra as camadas presentes no protocolo:

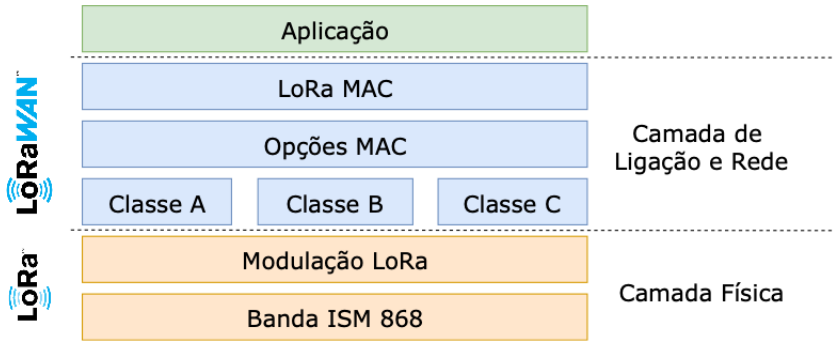

Fig. 2. Camadas do LoRaWAN

Arquitetura A rede do LoRaWAN utiliza uma topologia em estrela, apresentada na Figura 3), onde os dispositivos IoT, também conhecidos como equipamentos finais, End-Devices (ED), estão conectados através de uma ou várias gateways LoRaWAN a uma rede IP. Os dispositivos podem enviar dados para qualquer uma das gateways da mesma rede e esta vai tratar de encaminhar a informação para o servidor correto.

As gateways LoRaWAN são transparentes para os EDs, que são responsáveis por realizar o encaminhamento dos pacotes (processo de passagem de LoRaWAN para IP), e estão interligadas com o servidor de rede, Network Server (NS).

O NS está, por sua vez, ligado ao servidor aplicacional, Application Server (AS), encaminhando as mensagens dos EDs para a aplicação correta. No interior do NS encontra-se, por fim, um Join Server (JS) e este é responsável pela autenticação dos EDs na rede.

2.3 Trabalho Relacionado

Trabalhos anteriores desenvolvidos, incluindo [10] e [8], revelam a importância da introdução de IDS para a deteção de intrusões nos dispositivos IoT. Verificou-se que, nestes tipos de ambientes, os atacantes podem tentar obter acesso a dispositivos na qual não tem permissão. Os IDS tradicionais utilizados em redes não estão dimensionados para a complexidade do IoT. Os últimos desenvolvimentos indicam que é necessário incorporar mecanismos de inteligência artificial, de forma a ter perceção do comportamento do sistema, detetar novas formas de ataque e recorrer a IDS colaborativos.

Prabhakaran et al. propuseram a utilização de um IDS Suricata [1] para detetar ataques de Denial of Service (DoS) em redes com o protocolo IPv6 Low power Wireless Personal Area Networks (6LoWPAN) e de uma ferramenta de penetração (Metasploit [**?**]) para injetar tráfego. No entanto, estes focam-se apenas na especificação de assinaturas estáticas para a deteção deste tipo de ataques.

Uma semelhante aproximação foi realizada no artigo [4], indicando que o protocolo LoRaWAN demonstrou ser suscetível a replay, jamming, wormhole e

flipping attacks. Os ataques de jamming são abordados em detalhe, com o intuito de negar aos dispositivos de IoT o acesso ao serviço. Neste caso, recorrem a algoritmos conhecidos como o Kullback Leibler Divergence e o Hamming Distance para analisar o tráfego e conseguirem uma elevada precisão na deteção.

Por fim, o artigo [3] pretende expor a possibilidade da utilização de algoritmos de inteligência artificial para a análise comportamental dos dispositivos em LoRaWAN. Foram utilizados dois métodos de análise, o primeiro com o algoritmo k-means para criar grupos de dispositivos que partilham o mesmo comportamento e o segundo método, baseado em Decision Tree (DT) e Long Short Term Memory (LSTM) para prever o comportamento.

3 Arquitetura

Conforme foi mencionado no inicio do artigo, o objetivo foca-se na deteção de intrusões ao nível dos dispositivos de IoT que utilizam o protocolo de LoRaWAN para comunicação. Esta deteção é realizada com recurso à análise comportamental através de modelos de Machine Learning. A Figura 3) ilustra os dois cenários possíveis:

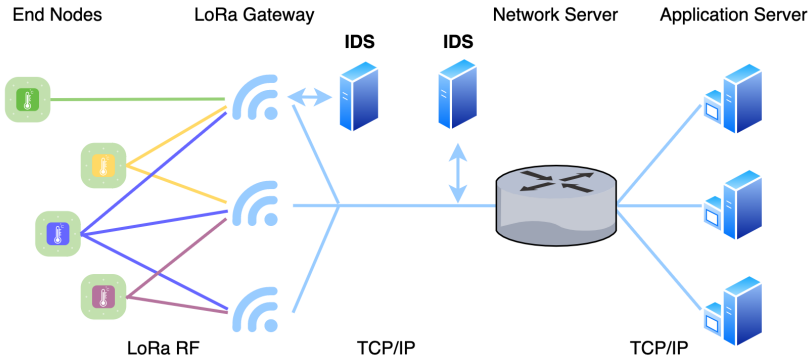

Fig. 3. Arquitetura Geral

- A instalação do IDS é realizada em cada gateway.
- A instalação do IDS é apenas realizada junto do NS, recebendo assim os pacotes de todas as gateways.

No cenário em que o IDS está implementado no NS existe o problema de poderem existir mensagens iguais com tempos diferentes de receção o que não é contemplado neste artigo.

Os dados dos sensores são enviados para a gateway com recurso ao protocolo LoRaWAN. Esta ao reencaminhar para o Network Server recorre ao formato

packet-forwarder da Semtech para passar para um pacote IP o conteudo do dispositivo. É precisamente após a gateway que o IDS vai analisar os pacotes que passam a rede.

3.1 Análise dos dados

De forma a ser possível a integração de modelos de Machine Learning com a arquitetura indicada, é necessário recorrer a amostras de dados de dispositivos reais. Deste modo o modelo a criar é o mais realista possível.

Posto isto, recorreu-se a um conjunto de amostras de dados de uma gateway LoRaWAN, que pertence ao ISEL, instalada numa das Torres das Amoreiras e ligada à rede TTN (The Things Network).

Esta recebe mensalmente uma média de 500 000 mensagens de milhares de dispositivos. Realizou-se um pré-processamento do conjunto de dados, com recurso á biblioteca "ProfileReport" do "Pandas_profiling" [13] de forma a gerar um relatório dos dados. O programa foi executado no Colab da Google de forma a ser possível a colaboração entre todos.

A amostra analisada foi relativa ao mês de Junho de 2020, mês este de plena pandemia Covid-19 em Portugal. Do relatório foi possível concluir que existiram 509 042 mensagens enviadas de 2876 dispositivos para o NS. Dos vários parâmetros utilizados nas mensagens, selecionaram-se os mais relevantes de forma a ser possível distinguir os vários dispositivos. As Figuras 4) e 5) mostram o desenvolvimento da relação sinal-ruído e da potência do sinal da amostra total.

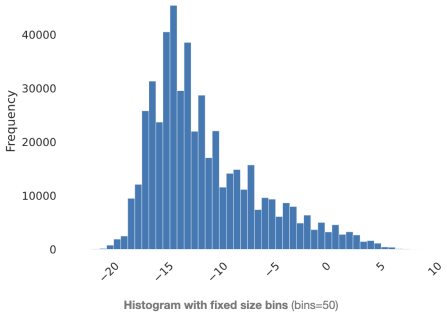

Fig. 4. Valores de LSNR da amostra

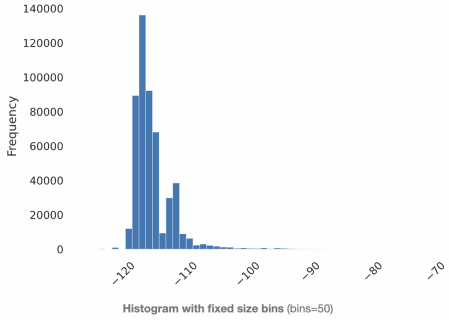

Fig. 5. Valores de RSSI da amostra

É possível concluir que a relação sinal-ruído é mais predominante nos -15 dB e relativamente ao RSSI, este destaca-se nos -118 dBm, dando a entender que os dispositivos se encontram todos numa posição geográfica fixa.

Pela Figura 6) verifica-se que a opção de SF = 12 e BW = 125 é a mais predominante. A Figura 7) indica que maioritariamente o tamanho do payload é de 70 Bytes. A biblioteca utilizada analisa também a correlação dos parâmetros

Value	Count	Frequency (%)
SF12BW125	410088	80.5%
SF7BW125	69173	13.6%
SF10BW125	11085	2.2%
SF8BW125	7491	1.5%
SF11BW125	7344	1.4%
SF9BW125	4046	0.8%

Fig. 6. SF e BW da amostra

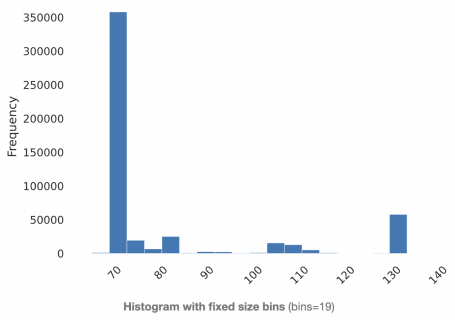

Fig. 7. Comprimento em Bytes do payload

utilizados de forma a ser possível perceber se são úteis para o modelo de aprendizagem. A Figura 8) mostra um tipo de correlação denominada por Phik que permite analisar parâmetros categóricos, ordinais relativamente à sua correlação.

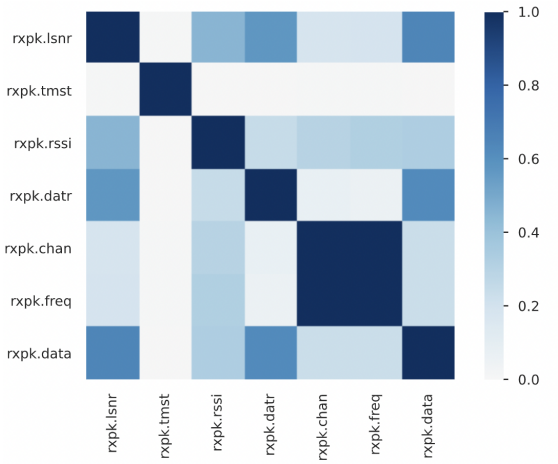

Fig. 8. Correlação Phik

Verifica-se que a correlação é máxima quando se relaciona um determinado parâmetro com ele próprio, como já seria de esperar. É de notar que a correlação também é máxima entre o "chan" (channel) e a "freq" (frequency) utilizada. O parâmetro "data" (tamanho do payload em Bytes) tem uma elevada correlação com o "lsnr", o "datr" e com "rssi", tendo em conta o funcionamento do LoRaWAN faz sentido. Por fim, conclui-se que o parâmetro "tmst" não tem qualquer correlação, com outro a não ser com o próprio.

Da amostra de dados total, optou-se testar a arquitetura apenas para um dispositivo e apenas para um dia de dados. O dispositivo escolhido tem como

DevAddr = 0000BF53 que de acordo com o relatório anterior, os valores mais frequentes são:

- **datr** = SF12BW125
- **lsnr** = [-14 a -11] dB
- **rssi** = -118 dBm
- **data** = 20 Bytes

É possível concluir, juntamente com outros parâmetros, que o dispositivo se encontra numa posição geográfica fixa.

4 Implementação

Neste capítulo é apresentada a implementação de um sistema composto por um IDS open-source, uma base de dados e programas em Python de forma a executar algoritmos de *Machine Learning*.

Como IDS open-source optou-se pelo Suricata, este permite analisar pacote a pacote à procura de intrusões com base em assinaturas (SIDS), como também executar scripts em LUA e desta forma interligar bases de dados e executar outros scripts baseados em Python. Tornando assim o IDS apto para analisar comportamentos (AIDS).

A base de dados escolhida foi o CrateDB, baseada em SQL, relacional, escalável, adequada para *Machine Learning* e para dados baseados no tempo.

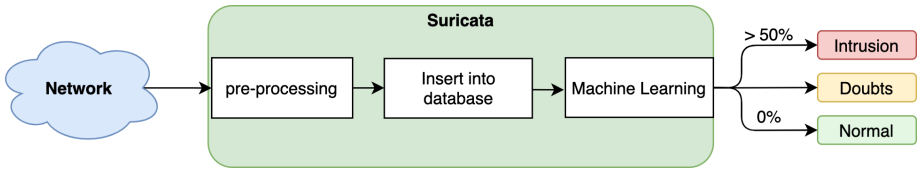

Fig. 9. Arquitetura Funcional

De acordo com a Figura 9 o Suricata é responsável por analisar os pacotes provenientes da rede, realizar o pré-processamento, inserir os valores relevantes na base de dados e por fim executar as funções de *Machine Learning*.

O bloco de pré-processamento tem a função de preparar as amostras de LoRaWAN, escolhendo apenas os parâmetros necessários e passar para o próximo bloco. Os blocos seguintes são funções em Python executadas pelo Suricata. O primeiro realiza a inserção dos parâmetros para a base de dados CrateDB e o último bloco tem como função executar o modelo de *Machine Learning*, que para este caso testou-se com o KNN.

Como parâmetros relevantes tem-se: *a)* **tmst** (*timestamp*); *b)* **latitude**; *c)* **longitude**; *d)* **chan** (canal de comunicação); *e)* **bw** (largura de banda); *f)* **sf**

(spreading factor); *g)* **rssi** (potência de sinal recebida); *h)* **lsnr** (relação sinal-ruído); *i)* **lenpayload** (tamanho em bytes do *payload*); *j)* **payload** (conteúdo do *payload*); *k)* **flag** (indicação de intrusão).

O parâmetro "flag" é adicionado pelo Suricata de forma a classificar o pacote, numa primeira instância como não intrusão. Posteriormente ao ser analisado pelo algoritmo pode passar a intrusão.

4.1 Suricata

Configurou-se em primeiro lugar uma assinatura de forma a que, quando a gateway LoRa envia um pacote para o NS, este execute um script em LUA para pré-processar o pacote recebido. Optou-se por realizar a inserção dos parâmetros no CrateDB com recurso a script Python devido à existência de uma biblioteca já desenvolvida para o propósito.

Até ao momento, os scripts estão a ser executados cada vez que é detetado um pacote com a direção gateway LoRa → NS, o que pode ser ineficiente tendo em conta que o CrateDB permite "bulk inserts", ou seja inserir na base de dados um bloco de informação, e a execução do KNN também pode ser realizada para um conjunto de pacotes. No entanto, tendo em conta que o débito binário em LoRaWAN não é muito elevado, em comparação com redes tradicionais, e que o Suricata tira proveito dos vários núcleos do CPU para processar os vários pacotes, optou-se por detetar o mais rapidamente possível intrusões no sistema.

4.2 Classificação de pacotes

Relativamente à utilização de algoritmos de inteligência artificial, a implementação foi segmentada em duas partes. A primeira parte consiste na apresentação dos fluxogramas simplificados e a segunda parte contém a escolha do modelo para dar suporte à análise comportamental.

A Figura 10 ilustra o fluxo dos dados na sua maioria do tempo, ou seja, um pacote quando é detetado pelo Suricata, este insere na base dados e executa este fluxograma. De seguida este script vai à base de dados recolher o modelo atualizado, de acordo com o DevAddr, (previamente treinado) e executa uma função "prob_predict". Esta função retorna a probabilidade do pacote analisado ser uma intrusão, logo caso seja 0% o programa é terminado, se estiver entre 0% e 50% entra num caso de dúvidas em que o utilizador tem de indicar ao modelo se é realmente uma intrusão ou se é comportamento normal (modelo supervisionado), por fim tudo o que for acima de 50% é considerado uma intrusão. Neste último caso, é enviado uma mensagem para o Suricata a alertar da intrusão e é necessário alterar a "flag" do pacote para intrusão na base de dados.

A Figura 11 descreve o fluxograma para a reaprendizagem do modelo que pode ser executado de diferentes formas. É necessário ir à base de dados onde se encontra o histórico dos valores do dispositivo e executar a função "fit" para criar o novo modelo com base nos novos valores. Por fim, insere-se na base de dados o novo modelo.

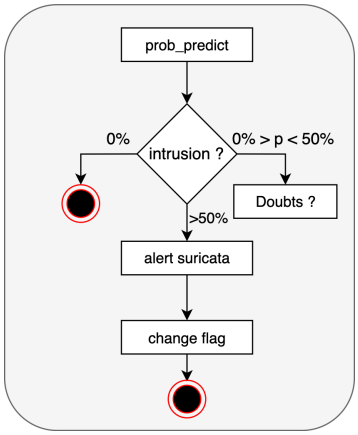

Fig. 10. Fluxograma para predict

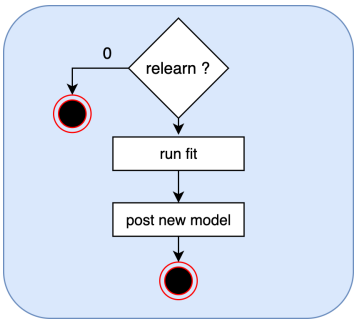

Fig. 11. Fluxograma para realizar reaprendizagem

É importante realçar que a utilização desta arquitetura pressupõe que o modelo esteve um tempo inicial a aprender o funcionamento normal da rede e que após esse tempo está apto para detetar intrusões.

5 Resultados

De maneira a validar a arquitetura proposta, esta secção apresentada diferentes resultados, nomeadamente sobre a escolha dos parâmetros do modelo KNN, as métricas utilizadas e o resultado da análise.

O modelo treinado tendo em conta o conjunto de dados, divididos entre dados de treino e dados de teste, pode ser avaliado usando várias métricas, sendo as mais comuns a precisão média, *recall* e a F1. As duas primeiras relacionam o número de positivos verdadeiros com os negativos verdadeiros e falsos, a medida F1 é uma média que tem em conta as métricas anteriores. Em qualquer caso os valores podem variar entre 0 e 1, sendo o modelo treinado apresenta valor sempre acima de 0.91 o que é bastante promissor. Na Figura 12 são apresentados os dados usados para teste do modelo. Dos vários valores envolvidos nos mais de 500 pacotes usados para teste, e não usados para treino, destacamos os campos relativos à dimensão do pacote, largura de banda, LSNR e RSSI, os quais foram adulterados ao conjunto para representar situações anómalas.

Na Figura 13 são apresentados os resultados de analisar cada um dos pacotes anteriores. Quando o sistema considera o pacote não suspeito é atribuído o valor "1", quando é suspeito é atribuído o valor "2" e quando há dúvidas é atribuído o valor "3". Em linha com os valores altos de precisão e media F1, o sistema apresenta uma elevada capacidade de distinguir os diferentes pacotes mesmo tendo em conta campos muito variáveis como é o caso do LSNR e RSSI.

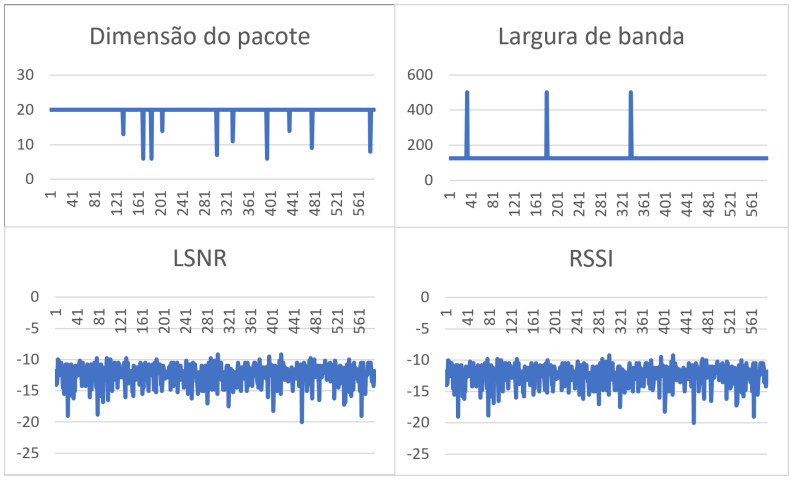

Fig. 12. Características dos pacotes e resultados da aplicação do modelo

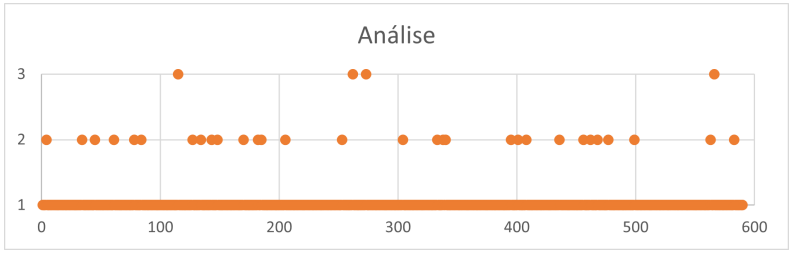

Fig. 13. Características dos pacotes e resultados da aplicação do modelo

6 Conclusões

O desenvolvimento em grande escala de redes da Internet das Coisas (IoT) são cada vez mais uma realidade e o protocolo LoRaWAN é uma das tecnologias IoT mais utilizadas. O seu uso disseminado e a sua natureza distribuída introduz novas ameaças. Este artigo apresenta um sistema para deteção de anomalias em redes LoRaWAN usando um algoritmo de classificação supervisionado, concretizado com o IDS Suricata. Como caso de uso foi usado um conjunto de dados obtido com mais de 500 mil mensagens de sensores localizados na cidade de Lisboa. O modelo treinado apresenta uma baixa taxa de falsos positivos. Como trabalho futuro o modelo será adaptado a cenários em que os sensores ou a *gateway* possam não estar fixos, como é o caso de equipamentos em transportes de passageiros ou mercadorias.

References

1. Albin, E., Rowe, N.C.: A realistic experimental comparison of the suricata and snort intrusion-detection systems. In: 2012 26th International Conference on Advanced Information Networking and Applications Workshops. pp. 122–127 (2012). https://doi.org/10.1109/WAINA.2012.29
2. Bouziani, O., Benaboud, H., Chamkar, A.S., Lazaar, S.: A comparative study of open source idss according to their ability to detect attacks. In: Proceedings of the 2nd International Conference on Networking, Information Systems amp; Security. NISS19, Association for Computing Machinery, New York, NY, USA (2019)
3. Cuomo, F., Garlisi, D., Martino, A., Martino, A.: Predicting lorawan behavior: How machine learning can help. Computers **9**(3) (2020). https://doi.org/10.3390/computers9030060, https://www.mdpi.com/2073-431X/9/3/60
4. Danish, S.M., Nasir, A., Qureshi, H.K., Ashfaq, A.B., Mumtaz, S., Rodriguez, J.: Network intrusion detection system for jamming attack in lorawan join procedure. In: 2018 IEEE International Conference on Communications (ICC). pp. 1–6 (2018). https://doi.org/10.1109/ICC.2018.8422721
5. Fattah, H.: 5G LTE Narrowband Internet of Things (NB-IoT). CRC Press, Inc., USA, 1st edn. (2018)
6. Febriyandi, F., Arifin, A.S., Nashiruddin, M.I.: Sigfox based network planning analysis for public internet of things services in metropolitan area. In: 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). pp. 21–27 (2020). https://doi.org/10.1109/IAICT50021.2020.9172012
7. Khraisat, A., Alazab, A.: A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecur. **4**(1), 18 (2021). https://doi.org/10.1186/s42400-021-00077-7, https://doi.org/10.1186/s42400-021-00077-7
8. Mudgerikar, A., Sharma, P., Bertino, E.: Edge-based intrusion detection for iot devices. ACM Trans. Manage. Inf. Syst. **11**(4) (Oct 2020). https://doi.org/10.1145/3382159, https://doi.org/10.1145/3382159
9. Network, T.T.: Lorawan architecture, https://www.thethingsnetwork.org/docs
10. Network, T.T.: Abp vs otaa (2021), https://www.thethingsindustries.com/docs/devices/abp-vs-otaa/
11. Ratasuk, R., Mangalvedhe, N., Ghosh, A., Vejlgaard, B.: Narrowband lte-m system for m2m communication. In: 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall). pp. 1–5 (2014). https://doi.org/10.1109/VTCFall.2014.6966070
12. Safaric, S., Malaric, K.: Zigbee wireless standard. In: Proceedings ELMAR 2006. pp. 259–262 (2006). https://doi.org/10.1109/ELMAR.2006.329562
13. Stančin, I., Jović, A.: An overview and comparison of free python libraries for data mining and big data analysis. In: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 977–982 (2019). https://doi.org/10.23919/MIPRO.2019.8757088