UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Jonathan Ortiz Preuss

# DETECÇÃO DE ANOMALIAS EM INTERNET DAS COISAS: UMA ABORDAGEM UTILIZANDO ANÁLISE DE QUANTIFICAÇÃO DE RECORRÊNCIA

Santa Maria, RS
2020

**Jonathan Ortiz Preuss**

# DETECÇÃO DE ANOMALIAS EM INTERNET DAS COISAS: UMA ABORDAGEM UTILIZANDO ANÁLISE DE QUANTIFICAÇÃO DE RECORRÊNCIA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Orientador: Prof. Dr. Raul Ceretta Nunes

Santa Maria, RS

2020

Ortiz Preuss, Jonathan

DETECÇÃO DE ANOMALIAS EM INTERNET DAS COISAS: Uma abordagem utilizando Análise de Quantificação de Recorrência / por Jonathan Ortiz Preuss. – 2020.
98 f.: il.; 30 cm.

Orientador: Raul Ceretta Nunes
Dissertação (Mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Pós-Graduação em Ciência da Computação , RS, 2020.

1. Detecção de Anomalias. 2. Internet da Coisas. 3. Análise Quantitativa da Recorrência. I. Ceretta Nunes, Raul. II.DETECÇÃO DE ANOMALIAS EM INTERNET DAS COISAS: Uma abordagem utilizando Análise de Quantificação de Recorrência.


E-mail: jonathan.preuss@redes.ufsm.br

**Jonathan Ortiz Preuss**

# DETECÇÃO DE ANOMALIAS EM INTERNET DAS COISAS: UMA ABORDAGEM UTILIZANDO ANÁLISE DE QUANTIFICAÇÃO DE RECORRÊNCIA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

**Aprovado em 31 de março de 2020:**

**Raul Ceretta Nunes, Dr. (UFSM)**
(Presidente/Orientador)

**Rogério Correa Turchetti, Dr. (UFSM)**

**Érico Marcelo Hoff do Amaral, Dr. (UNIPAMPA)**

Santa Maria, RS

2020

# AGRADECIMENTOS

Agradeço a minha família, principalmente à minha mãe Terezinha e à meu Pai Valton pelo apoio e incentivo.

Ao meu orientador, Professor Dr. Raul Ceretta Nunes, por todo suporte e incentivo na condução e desenvolvimento desse trabalho e também pelos conselhos para vida.

A todos os professores que fizeram parte da minha formação até o momento, desde as professoras das séries inciais até os professores/as do programa de pós-graduação.

Agradeço aos amigos verdadeiros que se mantiveram me apoiando em todo o momento, em especial para os amigos e colegas do Grupo de Pesquisa GTSeg.

Agradeço a Deus pela vida e conforto concedido nos momentos difíceis.

*”Memento mori”*

# RESUMO

## DETECÇÃO DE ANOMALIAS EM INTERNET DAS COISAS: UMA ABORDAGEM UTILIZANDO ANÁLISE DE QUANTIFICAÇÃO DE RECORRÊNCIA

AUTOR: JONATHAN ORTIZ PREUSS
ORIENTADOR: RAUL CERETTA NUNES

Ambientes de Internet das Coisas estão sendo alvo de um grande número de cyber ataques, principalmente devido a simplicidade de projeto dos equipamentos envolvidos e as vulnerabilidades decorrentes disto. Segundo a literatura, soluções tradicionais de segurança não são eficazes para redes IoT e é preciso o desenvolvimento de novas técnicas e modelos para segurança. As soluções de segurança que vem sendo propostas na literatura em sua maioria utilizam técnicas de *Machine Learning*, tratam do tráfego do ambiente IoT de forma agregada e estão ligadas a aplicações e tecnologias específicas. Soluções capazes de lidar com as características e comportamentos heterogêneos dos ambientes IoT ainda são um desafio. Este trabalho propõe um método para detecção e identificação de dispositivos anômalos, através da segmentação do ambiente IoT em classes de dispositivos e do emprego, sobre essas classes, da técnica de análise quantitativa da recorrência em conjunto com classificador adaptativo. Para o processo de validação, o método foi empregado em dois cenários de redes IoT, um cenário analisando o trafego de forma agregado e o outro cenário com o trafego tratado de forma segmentada de acordo com as classes comportamentais,ambos cenários com traços de ataques de ḿalwares e DDoS. Para fins de comparação da capacidade classificação, outros dois métodos foram implementados e executados (em cenários segmentados e agregados). A série de experimentos realizados demonstra os benefícios de tratar o tráfego de forma segmentada, bem como as elevadas taxas de acurácia e precisão alcançadas pelo método proposto, aonde foi alcanço uma taxa de 91,66% de precisão para o método AIDA em um ambiente segmentado e 68% de precisão quando método utilizado em um ambiente agregado, em relação aos demais métodos experimentados, o AIDA se demonstra superior e a diferença de acurácia variam de 0,55% a 37,24% (agregado) e 19,26% a 37.82%(segmentado).

**Palavras-chave:** Detecção de Anomalias. Internet da Coisas. Análise Quantitativa da Recorrência.

# ABSTRACT


## ANOMALIES DETECTION OF THINGS INTERNET DEVICES: AN APPROACH USING RECURRENCE QUANTIFICATION ANALYSIS

AUTHOR: JONATHAN ORTIZ PREUSS
ADVISOR: RAUL CERETTA NUNES

Internet of Things environments are the target of a large number of cyber attacks, mainly due to the simplicity of design of the equipment involved and the vulnerabilities resulting from this. According to the literature, traditional security solutions are not effective for IoT networks and it is necessary to develop new techniques and models for security. The security solutions that have been proposed in the literature mostly use Machine Learning techniques, deal with the traffic of the IoT environment in an aggregated way and are linked to specific applications and technologies. Solutions capable of dealing with the heterogeneous characteristics and behaviors of IoT environments are still a challenge. This work proposes a method for the detection and identification of anomalous devices, through the segmentation of the IoT environment in device classes and the use, on these classes, of the technique of quantitative analysis of recurrence in conjunction with an adaptive classifier. For the validation process, the method was used in two scenarios of IoT networks, one scenario analyzing traffic in an aggregate manner and the other scenario with traffic treated in a continued manner according to the behavioral classes (both scenarios with malware and DDoS attacks). For the purpose of comparing the classification capacity, two other methods were implemented and executed (in segmented and aggregated scenarios). The series of experiments carried out demonstrates the benefits of treating traffic in a segmented manner, as well as the high rate of accuracy and precision achieved by the proposed method, where a rate of 91.66% accuracy was achieved for the AIDA method in an environment segmented and 68% accuracy when used in an aggregate environment, in relation to the other tested methods, AIDA is superior and the difference in accuracy varies from 0.55% to 37.24% (aggregate) and 19 , 26% to 37.82% (segmented).



**Keywords:** Anomaly Detection. Internet of Things. Quantitative Recurrence Analysis.

# LISTA DE FIGURAS

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| IoT | Internet of Things |
| AQR | Análise da Quantificação da Recorrência |
| GR | Gráfico da Recorrência |
| MQR | Medida de Quantificação da Recorrência |
| RR | Taxa de Recorrência |
| DET | Determinismo |
| ENT | Entropia de Shannon |
| TREND | Tendência |
| LAM | Laminaridade |
| L | Comprimento Médio das Linhas Diagonais Comprimento |
| Lmax | Comprimento Máximo das Linhas Diagonais |
| TT | Comprimento Médio das Estruturas Verticais |
| MD | Matriz de Distâncias |
| MR | Matriz de Recorrência |
| ST | Série Temporal |
| IP | Internet Protocol |
| DDOS | Distributed Denial of Service |
| AIDA | Anomaly Internet of Things Detection Alert |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| SSH | Secure Socket Shell |
| FTP | File Transfer Protoco |
| NTP | Network Time Protocol |
| ICMP | Internet Control Message Protocol |
| DNS | Domain Name System |

# SUMÁRIO

# 1 INTRODUÇÃO

Redes de Internet das Coisas, do inglês *Internet of Things* (IoT), tem sido alvo de constantes ciberataques, devido as características comuns desses ambientes, como dispositivos com limitações de *hardware* e *software* (JING; ATHANASIOS V. VASILAKOS J. WAN, 2014). Dispositivos comprometidos por ataques, oferecem uma série de riscos ao ambiente IoT, sendo esses: o roubo e distorção de dados (LIN; REN, 2016) (MOSENIA ARSALAN JHA, 2017), acesso indevido para rede interna (MIETTINEN; S. MARCHAL I. HAFEEZ, 2017) ou utilização como vetor para ataques contra dispositivos vizinhos ou terceiros (KOLIAS CONSTANTINOS KAMBOURAKIS, 2017) (SPOGNARDI ANGELO DONNO, 2017) (ALABA; M. OTHMAN IBRAHIM ABAKER T. HASHEM, 2017). Tendo em mente que muitos sistemas IoT estão operando com dados e aplicações críticas, é preciso garantir a integridade dos dispositivos e aplicações IoT, de forma que a agilidade na detecção de anomalias e falhas nesses sistemas e dispositivos possa evitar desastres.

Outras características de ambientes IoT como, diversidade de tecnologias, velocidade que nodos são adicionados e removidos desse ambiente (MIETTINEN; S. MARCHAL I. HAFEEZ, 2017) e o volume de dados gerados em suas comunicações (AHSAN; BAIS, 2016), tornam redes IoT sistemas dinâmicos e com comportamentos não lineares, no qual os sistemas tradicionais de segurança de rede não são eficazes (ALABA; M. OTHMAN IBRAHIM ABAKER T. HASHEM, 2017) (SHEIKHAN; BOSTANI, 2016).

A literatura apresenta diversos trabalhos para detecção de anomalias em ambientes e dispositivos de IoT. Grande parte são desenvolvidos para aplicações IoT específicas, como os demonstrados por (R.FU; K. ZHENG D. ZHANG, 2011) (PACHECO; HARIRI, 2016), aonde são analisados dados coletados por sensores, baseado nas variações desses dados, indicando a ocorrência de anomalias nos sensores. Já em (CAMINHA; A. PERKUSICH, 2018) os autores buscam a identificação de dispositivos anômalos através de um método de reconhecimento de confiança dos elementos IoT, utilizando técnicas de *machine learning* para analisar dados de uma aplicação em uma determinada janela de tempo deslizante. A abordagem adotada é limitada por dois fatores, primeiramente por ser preciso ter um amplo conhecimento da aplicação IoT que está sendo monitorada e, segundo, pela necessidade de um sistema de monitoramento distinto para cada aplicação IoT presente em uma rede.

Por outro lado, existem iniciativas para detecção de anomalias em ambientes IoT a nível

de redes, porém a maioria dos autores detêm o seu foco em redes de sensores sem fio, e visam apenas alguns conjuntos de ataques característicos desses ambientes. Por exemplo, os trabalhos apresentados por (SHEIKHAN; BOSTANI, 2016) (RAZAA; L. WALLGRENA, 2013) propõem métodos híbridos para detecção de anomalias entre redes 6LowPAN e LAN. Outros autores apresentam abordagens mais inclusivas, analisando características dos pacotes de rede de forma mais ampla, como por exemplo, (INDRE; LEMNARU, 2016) que apresenta um sistema para detecção de anomalias através da análise de características e correlações histórica de pacotes, e (PROKOFIEV; Y. S. SMIRNOVA, 2018) que apresenta um método para detecção de ação de *botnets* em redes IoT, que utiliza técnicas de regressão logística para analisar características da rede.

Já em um contexto para detecção de dispositivos anômalos utilizando dados de rede, exitem alguns trabalhos como (FERRANDO; STACEY, 2017), no qual é demonstrado um modelo para classificação de comportamentos dos dispositivos utilizando a medida da distância euclidiana para analisar a similaridade de características de tráfego, e como (LIN; REN, 2016) no qual os autores demonstram um método para avaliar e quantificar a confiança de novos elementos ingressados em uma redes IoT, que utiliza redes Baysianas e dados de interação entre os sensores da rede.

No contexto de ambientes IoT, métodos mais abrangentes capazes de identificar dispositivos anômalos em sistemas heterogêneos e de comportamento dinâmico são pouco explorados. A literatura apresenta soluções com resultados na detecção de anomalias sobre o tráfego de rede, mas pouco aborda sobre soluções para identificação dos dispositivos responsáveis por tal.

Apesar do ambiente IoT ser heterogêneo, os subconjuntos de dispositivos tem comportamentos regulares, pois dispositivos IoT possuem um número limitado de função, estados e endereços de comunicação (DOSHI; N. APTHORPE, 2018). De acordo com (HAIBO; Z. KECHEN, 2018), padrões de comunicação como frequência, tamanho e volume de pacotes transmitidos podem ser usados para criar um grau de similaridade recorrente no tráfego desses dispositivos.

O conceito de Análise da Quantificação da Recorrência (AQR) é um conceito matemático que viabiliza a análise de comportamentos de tráfegos não lineares que tendem a se repetir no decorrer de um intervalo de tempo (WEBBER; MARWAN, 2015). Esse conceito foi empregado com sucesso na detecção de anomalias em tráfego de redes em (RIGHI; NUNES, 2016). Segundo os autores, AQR pode fornecer meios de análise eficazes e de tempo real para

aplicações de segurança de redes.

Frente ao exposto, embora a diversidade de dispositivos IoT seja grande, poucos trabalhos de detecção de anomalias consideram este fator sem apresentar soluções que exigem prévio conhecimento dos dispositivos ou aplicações. Ainda, embora os dados de rede podem apresentar recorrência, não foi encontrado nenhum trabalho que explore técnicas de análise da recorrência nos dados de uma rede IoT.

## 1.1 OBJETIVOS

Este trabalho teve como objetivo apresentar um método para detecção de dispositivos anômalos em redes IoT empregando segmentação e a técnica de Análise da Quantificação da Recorrência (AQR). Tal método é baseado na segmentação de classes de dispositivos IoT seguido da extração e análise de características dinâmicas (MQRs) por classe. A detecção, centrada na Clusterização Adaptativa das MQRs, visa alcançar uma melhor eficiência na detecção de dispositivos anômalos.

Para que o objetivo elencado fosse alcançado, os seguintes objetivos específicos foram percorridos:

- projetar um método para detecção de anomalias em redes IoT apoiado pela técnica de AQR aplicada a classes de dispositivos IoT;
- mapear e identificar quais características de tráfego de uma rede IoT que melhor descrevem o comportamento de dispositivos IoT, bem como quais as características dinâmicas (MQRs) possibilitam a identificação de anomalias no comportamento desses dispositivos;
- implementar o protótipo inicial do sistema para detecção de anomalias em redes IoT e realizar o processo de ajustes e calibragem do sistema;
- validar a solução proposta utilizando bases de dados de redes IoT, analisando se o emprego do método colabora para detecção e dispositivos IoT anômalos.

## 1.2 LIMITAÇÃO DE ESCOPO

A pesquisa que norteia esse trabalho objetiva desenvolver um método para detecção de anomalias em redes IoT IP. Para isso, o método desenvolvido opera com dados de tráfego, ou

seja, dados da camada de rede. Nessa versão, o presente método não é projetado para operar com dados da camada de aplicação dos ambientes IoT, tais como dados textuais ou numéricos de sensores ou atuadores.

Cabe salientar que o método projetado limita-se em identificar e relatar quais grupos de dispositivos apresentam alguma anomalia em seu comportamento, sem o objetivo de executar medidas para conter ou identificar a causa e dispositivo de tal anomalias, como por exemplo classificar tal anomalia como um tipo de ataque.

## 1.3 CONTRIBUIÇÃO E JUSTIFICATIVA

No âmbito deste trabalho, a principal contribuição é o desenvolvimento de um novo método para detecção de anomalias em redes IoT empregando técnicas de Análise da Quantificação da Recorrência (AQR) a dados segmentados por classe de comportamento de dispositivos. A detecção de anomalias em ambientes IoT é tema de muitas pesquisa, porém ainda apresenta limitações devido a características desses ambientes. A utilização de AQR empregado para detecção de anomalias é um assunto pouco explorado na literatura e os trabalhos existentes apresentam resultados promissores quando aplicado AQR em redes convencionais. Este trabalho é o primeiro, de nosso conhecimento, que explora o uso de AQR em redes IoT.

A necessidade do desenvolvimento de novos métodos para detecção de anomalias que são capazes de operar sobre as características de redes IoT, bem como a pouca exploração de AQR e seus atributos em relação a detecção de anomalias, justificam e propiciam o campo de investigação cientifica abordado nesse trabalho e o método resultante da pesquisa desenvolvida.

## 1.4 ESTRUTURA DO TRABALHO

O presente texto está assim organizado: no Capítulo 2 são expressos conceitos técnicos de Internet das Coisas, Análise da Quantificação da recorrência e os trabalhos relacionados; no Capítulo 3 é apresentada a proposta do método denominado AIDA (*Anomaly Internet of Things Detection Alert*) para detecção de anomalias, bem como sua estrutura de funcionamento; Ao longo do Capítulo 4 é abordado o processo de implementação e calibragem do método AIDA; o Capítulo 5 apresenta a série de experimentos e validações do método e os resultados obtidos; e, por fim, o Capítulo 6 expõe as considerações finais e os trabalhos futuros.

# 2 REFERENCIAL TEÓRICO

Neste capítulo são descritos os conceitos técnicos e teóricos presentes na literatura sobre os temas que norteiam o desenvolvimento da presente pesquisa. Tais conceitos são necessários para o entendimento deste trabalho. Esse capítulo está organizado da seguinte forma: a Seção 2.1 apresenta uma revisão teórica sobre o conceito de Internet das Coisas e a Seção 2.2 sobre segurança em redes IoT; a Seção 2.3 detalha a técnica de AQR, bem como sua estrutura e seus parâmetros de operação; a Seção 2.4 traz uma revisão dos trabalhos relacionados e a Seção 2.5 faz as considerações parciais relacionadas a este capítulo.

## 2.1 INTERNET DAS COISAS

O termo Internet das Coisas (IoT) teve sua primeira menção na década de 90, durante a apresentação do emprego da tecnologia de RFID e seus benefícios na indústria. O termo IoT foi utilizado para descrever o paradigma que possibilita de interligação, interação e troca de informações de elementos do mundo físico, através de tecnologias como RFID e WSN. A partir disso, com o surgimento de novas tecnologias o conceito de IoT começou a evoluir e se difundir, e novas definições foram cunhadas. IoT é entendida como uma rede de dispositivos distribuídos (HODO; X. BELLEKENS, 2016), onde por meio da coleta e do processamento dos dados em tempo real, cria-se uma plataforma que permite a interação entre sistemas e elementos ciber-físicos.

Semelhante a uma rede tradicional, a internet das coisas divide seus elementos em um sistema de camadas, porém não possuem um modelo padrão de referência. A literatura apresenta modelos de redes IoT com cinco, quatro e até mesmo três camadas (JING; ATHANASIOS V. VASILAKOS J. WAN, 2014), sendo o modelo mais comum e referido entre os pesquisadores o modelo de três camadas. Para que seja possível a interação entre os elementos de rede é necessário o uso de um elemento chamado *gateway* (QIAN ZHU RUICONG WANG, 2010). Esse elemento é localizado mais próximo dos elementos finais da rede IoT. O *gateway* possui uma maior capacidade computacional em relação aos demais elementos do ambiente IoT. O *gateway* IoT é responsável por receber e realizar um primeiro pré-processamento dos dados oriundo dos elementos conectados ao mesmo, antes de enviar esses dados para uma próxima aplicação ou realizar a interligação entre dispositivos e aplicações de tecnologias diferentes.

Uma característica predominante de uma rede IoT é o seu tráfego com comportamento heterogêneo e algumas vezes volumoso (P; C., 2017) (DOSHI; N. APTHORPE, 2018). Esse tráfego deriva da presença de subconjuntos distintos de dispositivos que compõe uma rede IoT (NIE; MA, 2012). Cada subconjunto IoT apresenta um dado comportamento, visto que são formulados para desempenhar funções especificas. Por serem específicos, os dispositivos desses subconjuntos são desenvolvidos com (DOSHI; N. APTHORPE, 2018)(HAIBO; Z. KECHEN, 2018): limitações de *hadware* e *software*, um número restrito de funções e estados, bem como heterogeneidade na periodicidade de comunicação.

Em ambientes IoT, é preciso identificar e analisar os subconjuntos individualmente, uma vez que se a análise do ambiente for realizada de uma forma unificada o comportamento de um subconjunto pode induzir distorções no resultado final (NIE; MA, 2012) (DOSHI; N. APTHORPE, 2018). Observar característica do tráfego de rede é uma das forma de identificar esses subconjunto e seus dispositivos. Tal condição é reforçada em (MIETTINEN; S. MARCHAL I. HAFEEZ, 2017) (SIVANATHAN A. SHERRATT D.; V., 2017) (SHAIKH; E. BOUHARB J. CRICHIGNO, 2018) (T. GARRETT S. DUSTDAR, 2018) (MEIDAN; M. BOHADANA A. SHABTAI1, 2017) (P; C., 2017) (J.CANEDO; SKJELLUM, 2016) (FERRANDO; STACEY, 2017) aonde os autores demonstram que é possível identificar e classificar grupos e dispositivos IoT, observando características tais como endereço MAC, portas e endereços IP de origem e destino, entre outros.

## 2.2 SEGURANÇA EM REDES IOT

Redes de IoT constantemente estão envolvidas em incidentes de segurança. Pelo que se sabe, o incidente de maior proporção ocorreu em 2016. Dispositivos IoT's foram utilizados para orquestrar um dos maiores ataques de negação de serviço distribuído (DDoS) já registrado nos últimos anos. O ataque utilizou um *malware* chamado Mirai, o qual disparou um ataque contra os serviços de uma empresa (Dyn) de DNS, afetando diretamente a disponibilidade de serviços como *Amazon*, *Spotify*, *Netflix*, *Twitter* entre outros (KOLIAS CONSTANTINOS KAMBOURAKIS, 2017) (SPOGNARDI ANGELO DONNO, 2017) (SINANOVIC; MRDOVIC, 2017).

Dado o número de dispositivos IoT conectados na internet, que está aumentando exponencialmente, aumenta também a preocupação e a necessidade de serviços de segurança para esses ambientes, uma vez que orquestrados esses dispositivos demonstram grande poder de ação e as consequências de um ataque podem causar danos severos em aplicações ou em vidas

humanas (PACHECO A. B. LUIS GONDIM J. C. JOAO; E., 2016) (LINDQVIST, 2017).

De acordo com (S. BABAR A. PRASAD, 2011), as principais ameaças presentes nas redes podem ser classificadas em:

- **Ataques Físicos**: todo o tipo de ataque que causa avarias ou modificações não autorizadas e que comprometem a integridade do hardware dos dispositivos;
- **Ataque de Rede**: ataques executados a nível de rede de comunicação, focando na captura do tráfego e utilização do poder de comunicação dos dispositivos para a execução de outros ataques, contra outros dispositivos ou aplicações;
- **Ataques aos Softwares**: o objetivo desse tipo de ataque é identificar e explorar falhas nos softwares em execução nos dispositivos da rede IoT;
- **Ataques aos Canais de comunicação**: consistem na escuta do canal de comunicação dos dispositivos IoT, com o intuito de coletar informações de interesse do atacante. Em caso do ambiente implementar comunicação criptografada o atacante busca identificar e obter dados dos dispositivos responsáveis por cifrar as informações em busca de obter a chave utilizada na criptografia dos dados;
- **Ataque de análise de criptografia**: esse tipo de ataque segue o mesmo princípio do ataque aos canais de comunicação, diferenciando-se apenas no seu objetivo que é focado em capturar a chave de criptografia para conseguir ler as informações capturadas.

Uma vez que observado essa classificação de ameaça IoT de forma mais ampla, em (Q. ASHRAFN, 2015) os autores analisam e identificam de forma mais especifica quinze ataques mais comuns no cenário de redes IoT, sendo eles:

- ***Jamming***: é um ataque contra ambientes IoT sem fio, que visa danificar a disponibilidade do meio de comunicação, dificultando ou anulando a utilização do meio pelos dispositivos da rede. Esse ataque explora às características da implantação remota e, principalmente, a falta de monitoramento desses dispositivos da IoT;
- ***Tampering***: é um tipo de ataque focado na confidencialidade e disponibilidade dos dispositivos através da violação dos dados do mesmo, aonde o atacante obtém acesso físico ao dispositivo e modifica, adiciona ou exclui dados no próprio dispositivo. Posteriormente o dispositivo comprometido é reintegrado na rede para atacar a rede internamente ou coletar novos dados do ambiente IoT;

- **Desativação**: basicamente é a destruição física do dispositivo ou aplicação de forma não autorizada. É um ataque que impacta diretamente na disponibilidade da rede IoT;

- **Colisão**: é um tipo de ataque no qual a falha de comunicação causada pela existência de sinais concorrentes pode ser resultado de um mal planejamento da rede, ou por conta de um atacante. Nesse segundo caso os pacotes de dados transmitidos pelo atacante podem ser interrompidos e retomados, criando uma transmissão assíncrona, o que pode resultar em incompatibilidade de soma de verificação das mensagens efetuadas por alguns protocolos. Repetidos ciclos de mensagens colidindo, afetam a disponibilidade de uma aplicação, podendo resultar em um ataque de negação de serviço;

- **Exaustão**: esse tipo de ataque normalmente é focado em dispositivos que utilizam baterias, aonde o objetivo do atacante e causar o esgotamento, seja da energia ou da capacidade de resposta do dispositivo, esgotando seu recurso através do aumento de funções ou requisições de execução;

- **Dessincronização e repetição**: esse ataque tem como base a solicitação de retransmissão de quadros perdidos, forçando repetidamente as mensagens para retransmissão e sincronização de dados. Normalmente nesse tipo de ataque o invasor armazena dados transmitidos anteriormente (solicitados pelas sucessíveis mensagens de retransmissão) e os repete posteriormente para um determinado dispositivo receptor, com o objetivo de enganar o nodo receptor com dados falsos;

- ***Hello flood***: esse ataque se utiliza do modo de operação de alguns protocolos de roteamento, aonde esses protocolos requerem que os nodos transmitam uma mensagem (*hello*) para se anunciarem a seus vizinhos. Um atacante nesse cenário transmite mensagens de anunciação para todos os nós de uma rede, fazendo com que esses nós enxerguem o nó atacante como o vizinho e envia para o mesmo todos os dados, dessa forma causando a perda de dados dos nós distantes;

- ***Sinkhole***: esse ataque e mais frequente em redes de sensor sem fio e seu funcionamento consiste em um atacante comprometer e inviabilizar o nó central da rede. Sem o nó central para comandar a rede as mensagens acabam sendo perdidas, podendo ocasionar um ataque de negação de serviço na aplicação da rede;

- ***Sybil***: é um ataque voltado para ambientes IoT que empregam regras de reputação e

confiabilidade de dispositivos. Em um ataque do tipo *Sybil* o dispositivo atacante forja um grande número de identidades, procurando se apresentar como a melhor opção entre todos os dispositivos (um nó de maior importância da rede). Quando aplicado a tabelas de roteamento, pode resultar na remoção de todos os vizinhos originais da tabela de nós sensores ativos na tabela de roteamento;

- **Encaminhamento seletivo**: esse ataque consiste em uma ação orquestrada de alguns nós maliciosos presentes na rede, os quais podem se recusar a encaminhar algumas mensagens da rede ou da aplicação IoT, causando assim uma série de atrasos ou sobrecarga na largura de banda da rede e consequentemente afetando a confidencialidade e a disponibilidade da rede IoT;

- ***Eavesdropping***: esse tipo de ação não é um ataque propriamente dito mas sim um princípio utilizado como base para outros ataques. O ataque do tipo *Eavesdropping* é classificado como passivo ou ativo. Nos ataques do tipo passivo, o atacante analisa o meio de transmissão de uma rede, captura e extrai informações vitais do tráfego de rede. Os ataques do tipo ativo, o atacante envia mensagens de controle como consultas para iniciar processos e posteriormente analisar as respostas dos dispositivo de destino. O resultado de ambos tipos é utilizado para iniciar outros ataques;

- ***Flooding***: nesse tipo de ataque o agente mal intencionado visa o esgotamento dos recursos do dispositivo ou aplicação IoT, através do envio de inúmeras solicitações para estabelecimento de conexão;

- ***Malware***: os ataques de *malwares* são causados por pequenos programas maliciosos inseridos no sistema de um dispositivo na rede IoT, e é um ataque contra a confidencialidade das informações e do funcionamento do dispositivo.

- ***Spoofing and message forging***: esse tipo de ataque procura forjar a identificação de um nodo ou reciar mensagens, com objetivo de se passar por outro;

- **Interseção**: esse ataque se baseia na obtenção de informações auxiliares sobre o sistema ou dispositivo IoT. As informações podem estar contidas como registros públicos da Web ou de terceiros. Esse tipo de ataque é voltado para anular a privacidade do sistema, uma vez que se tem informações sobre o funcionamento do sistema.

## 2.3 ANÁLISE DA QUANTIFICAÇÃO DA RECORRÊNCIA (AQR)

O conceito de recorrência para sistemas dinâmicos foi apresentado em 1890 por (POINCAR'E; HENRI, 1890). O autor cria o Teorema da Recorrência, no qual afirma que ao longo da execução de sistemas dinâmicos, as trajetórias resultantes tendem a retornar à regiões vizinhas ou adjacentes aos pontos de inicio. Uma técnica para analisar a recorrência de sistemas dinâmicos é a partir do gráfico da recorrência (RIGHIM; NUNES, 2014). Esta técnica foi proposta inicialmente em (ECKMANN; S. OLIFFSON KAMPHORST, 1987) e torna possível a visualização do comportamento da trajetória do espaço de fases multidimensional (WEBBER; MARWAN, 2015).

Ao longo dessa Seção é descrita a técnica de Análise da Quantificação da Recorrência (AQR) e os elementos que compõem a estrutura da AQR. A Seção está organizada dá seguinte maneira: na Seção 2.3.1 é apresentada a definição de sistemas dinâmicos; a Seção 2.3.2 aborda a definição e os métodos de obtenção do Raio da Vizinhança; na Seção 2.3.3 está expresso os fundamentos para construção do Gráfico da Recorrência e a Seção 2.3.4 detalha as Medidas de Quantificação da Recorrência (MQR).

### 2.3.1 Sistemas Dinâmicos

Um sistema dinâmico é um sistema que possui múltiplos estados e que esses estados apresentam relações entre si durante um período de tempo. Segundo (OTT; SAUER, 1994) tal sistema pode ser expresso como um modelo matemático, o qual descreve o progresso dos estados do sistema durante um intervalo de tempo. As variáveis desse modelo são chamadas de espaço de fases e são usadas para expressar a evolução do sistema dinâmico, e o valor dessas variáveis correspondem a um conjunto de coordenadas que indicam uma posição no espaço de fases, indicando um determinado instante de execução.

Para operar com sistemas dinâmicos é necessário conhecimento sobre algumas propriedades. Tais propriedades são descritas ao longo dessa seção, a qual está organizada da seguinte forma: a Seção 2.3.1.1 apresenta a concepção de Dimensão da Imersão; a Seção 2.3.1.2 descreve a propriedade de *delay* ou atraso; e, por fim, a Seção 2.3.1.3 traz a definição e fundamento da Janela de Tempo do Espaço Amostral.

#### *2.3.1.1 Dimensão de Imersão*

Para recompor e representar um espaço de fase de uma série temporal, é preciso um certo número de coordenadas do espaço de fase, chamado dimensão de imersão. A variável que indica o número mínimo de coordenadas necessários para reconstruir um determinado espaço de fase é chamada de Dimensão da Imersão (TAKENS, 1980). O processo de reconstrução de um espaço de fase, a partir de coordenadas de uma única série temporal é realizado através do Teorema de Tankes (TAKENS, 1980).

Existem duas técnicas para determinar o valor da dimensão de imersão, a primeira técnica é chamada de falsa vizinhança, na qual é realizado um aumento gradual no valor de *m* e observa-se quais pontos vizinhos se distanciam, indicando um "falso vizinho", uma vez que dados reais que orbitam uma mesma região de vizinhança tendem a permanecer agrupados, mesmo com o aumento da dimensão de imersão (B.; BROWN, 1992). A outra forma é chamada de Saturação, uma técnica que consiste em aumentar gradativamente a dimensão de imersão e observar as formas geométricas geradas pelo agrupamento dos dados do sistema, em busca de mudanças repentinas nas formas geométricas.

#### *2.3.1.2 Delay - Tempo de Atraso*

Ao reconstruir um espaço de fase é importante definir o intervalo de tempo entre cada análise de um novo estado de fase. Esse intervalo é chamado de *delay* $(\tau)$. Para estipular o *delay* deve-se observar duas situações. Se definir um valor de tempo muito longo a correlação de estados tende a ser defasada, refletindo nos valores de MQR, os quais podem ser erroneamente lidos. Caso o valor de *delay* for muito curto, aumentará o número de verificação de estado ocasionando uma alta taxa de correlações entre os valores da série temporal. Para definir um valor de *delay* existem dois métodos presentes na literatura: o método da Informação Mútua Média (IMM) e o método baseado na Função de Autocorrelação (FRASER; SWINNEY, 1986).

#### *2.3.1.3 Janela de Tempo do Espaço Amostral da ST*

A janela de espaço amostral é uma variável importante para a geração do gráfico da recorrência (GR), indicando o tamanho, em medida de tempo, para coleta de dados de uma série temporal de cada atributo de rede que está sendo analisado. Até a escrita desse trabalho, a literatura não apresenta técnicas ou métodos próprios para a escolha do valor de uma janela amostral.

Entretanto, em (RIGHI; NUNES, 2016) foi utilizado um conjunto de janelas amostrais variando de 15 a 105 segundos, separadas em intervalos de 15 segundos.

### 2.3.2 Raio da Vizinhança

O parâmetro de Raio da Vizinhança $(\varepsilon)$ é responsável por estipular qual o nível de dispersão entre os pontos recorrentes expressos em um gráfico da recorrência (GR), ou seja, o cálculo do Raio da vizinhança sob os estados de um sistema dinâmico determina quais pontos pertencem a uma vizinhança indicando recorrência ou não de um estado. Esse parâmetro é muito sensível e o valor adotado em uma determinada aplicação pode induzir a erros de classificação, uma vez que existe duas possibilidades (MARWAN; M.CARMEN ROMANO MARCOTHIEL, 2007): se for adotado um valor excessivamente alto para $\varepsilon$ serão encontrados muitos pontos recorrentes e assim falsos pontos de recorrência serão acusados; e se for adotado um valor baixo para $\varepsilon$ poucos ou nenhum ponto de recorrência será encontrado.

Para estipular o raio de vizinhança em uma aplicação, existe alguns métodos indicados pela literatura:

- **Escala logarítmica** – consiste em calcular MQRs sob múltiplos valores de $\varepsilon$, coletar e observar os resultados em busca de identificar uma região a qual apresenta mudanças repentinas em relação a tendência de valores anteriormente apresentada (M.MINDLIN; R.GILMORE, 1992);

- **Porcentagem** – consiste em utilizar uma porcentagem do diâmetro máximo do espaço de fase do sistema dinâmico em questão como valor de $\varepsilon$ (M.MINDLIN; R.GILMORE, 1992) (WEBBER; ZBILUT, 1994);

- **Intervalo percentual** – consiste em submeter valores para o Raio da Vizinhança e selecionar aqueles que resultam em uma Taxa de Recorrência com um percentual de 0,1 até 2% (WEBBER; ZBILUT, 2005);

- **Desvio padrão do ruído** – empregado em séries temporais que possuem ruídos, consiste em determinar o valor mínimo de $\varepsilon$ pelo resultado do cálculo de cinco vezes o valor do desvio padrão $(\sigma)$ do ruído da série temporal analisada (THIEL; M. CARMEN ROMANO J. KURTHS, 1992);

### 2.3.3 Gráfico da Recorrência

O Gráfico de Recorrência (GR) corresponde a a uma matriz quadrada, preenchida com pontos pretos e brancos, que descrevem a movimentação dos estados de um sistema dinâmico, tal como exemplificado na Figura 1. Para representação do estado do sistema dinâmico, cada ponto branco indica a não-recorrência e cada ponto preto indica existência da recorrência de um estado. Os estados do sistema dinâmico referentes aos pontos pretos orbitam em regiões (denominadas de raio da vizinhança) próximas entre si durante a trajetória do espaço de fases.

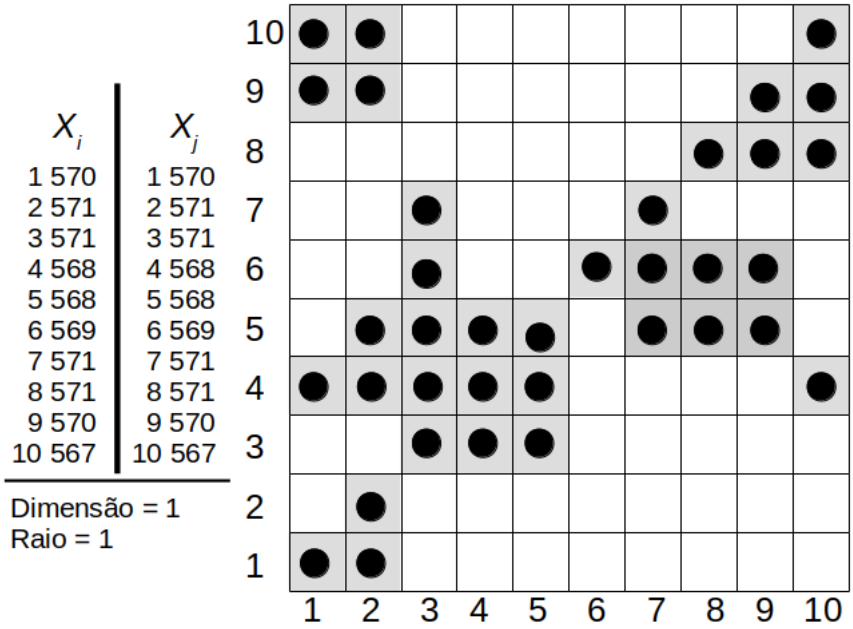


Figura 1 – Gráfico da recorrência baseado em (BAPTISTA, 2011)

O gráfico da recorrência é gerado com base em uma série temporal $X$ (Equação 2.1), composta por estados $x_i$, sendo $n$ o número de amostras (ECKMANN; S. OLIFFSON KAMPHORST, 1987). O cálculo dos pontos de recorrência nos GRs, realizado conforme a Equação 2.2, utiliza uma série temporal atrasada (Equação 2.3) em relação à cada estado $x_i$. Nessa série o número de estados $N$ é definido pela equação 2.4, aonde $m$ representa a dimensão de imersão e $\tau$ o tempo de atraso.

$$X = \{x_i, \ .., x_n\} \ , i = \{1, 2, 3, ..., n\} \tag{2.1}$$

$$R_{i,j} = \theta \left(\varepsilon - \parallel x_i - x_j \parallel\right). \tag{2.2}$$

$$x_j = [x_j, x_{j+\tau}, ..., x_N], j = 1, 2, 3, ..., N \tag{2.3}$$

$$N = n - (m-1)\tau \tag{2.4}$$

No cálculo da recorrência (Equação 2.2), o elemento $\varepsilon$ corresponde ao limiar utilizado na comparação entre dois estados, onde $x_i$ e $x_j$ correspondem aos estados do sistema no espaço de fase ($m$) sob análise. O operador $\theta$ representa a função de grau unitário (vide Equação 2.5), sendo que se a distância resultante entre dois estados $e_i$ e $e_j$ for menor que o $\varepsilon$, o valor de $\theta(\varepsilon)$ será 1, identificando que no GR na posição *i, j* terá um ponto preto. Caso contrário, se o valor for maior, o valor será 0 e será marcado um ponto branco no GR.

$$\theta\left(f\left(\varepsilon\right)\right) = \begin{cases} 0\left(\varepsilon - \parallel e_i - e_j \parallel\right) \leq 0 \\ 1\left(\varepsilon - \parallel e_i - e_j \parallel\right) > 0 \end{cases} \tag{2.5}$$

### 2.3.4 Medida da Quantificação da Recorrência (MQR)

A quantificação da recorrência pode expressar de forma numérica os dados apresentados pelo gráfico da recorrência. Essa quantificação é realizada por Medida de Quantificação da Recorrência (MQR). Algumas medidas tradicionais são (WEBBER; MARWAN, 2015) (MARWAN; M.CARMEN ROMANO MARCOTHIEL, 2007) (RIGHIM; NUNES, 2014): a razão da recorrência (RR), o determinismo (DET), a entropia (ENTR), a tendência (TREND), a laminaridade (LAM), o comprimento médio das linhas diagonais (L), o comprimento máximo das linhas diagonais (Lmax) e o comprimento médio das estruturas verticais (TT). Onde:

- **Razão de Recorrência (RR)** - corresponde a densidade dos pontos de recorrência no GR;

$$RR = \begin{cases} 0\left(\varepsilon - \parallel e_i - e_j \parallel\right) \leq 0 \\ 1\left(\varepsilon - \parallel e_i - e_j \parallel\right) > 0 \end{cases} \tag{2.6}$$

- **Determinismo (DET)** - mede a periodicidade do sistema, ou seja, a razão entre os pontos de recorrência que formam as estruturas diagonais e todos os pontos de recorrência apontados no GR;

- **Comprimento médio das linhas diagonais (L)** - indica a média de tempo que dois segmentos de uma trajetória se mantêm próximos entre sí;

- **Comprimento máximo das linhas diagonais (Lmax)** - indica o tempo máximo que dois segmentos permanecem próximos entre sí;
- **Entropia de Shannon (ENTR)** - permite compreender a distribuição de frequências dos comprimentos das linhas diagonais, ou seja, a complexidade da periodicidade de curta e longa duração (baixa entropia baixa complexidade);
- **Tendência (TREND)** - indica a não-estacionariedade do sistema, baseado no resultado do coeficiente de regressão linear sobre a densidade dos pontos de recorrência das diagonais paralelas a diagonal principal;
- **Laminaridade (LAM)** - mede a estabilidade do sistema (razão entre os pontos de recorrência que formam as estruturas verticais e todos os pontos de recorrência presentes plotados no GR);
- **Comprimento médio das estruturas verticais (Trapping Time - TT)** - indica o tempo médio que o sistema permanece em um determinado estado específico.

## 2.4 TRABALHOS RELACIONADOS

Diferentes abordagens e técnicas têm sido propostas para observar o tráfego de rede e identificar anomalias que correspondam a ataques aos serviços disponíveis na rede de computadores (RAUT S. ABHINAV, 2014). Na internet das coisas o foco também tem sido observar se dispositivos da rede estão servindo de fonte para ataques. As principais abordagens para a observação do tráfego correspondem a análises de dados por técnicas estatísticas ou por técnicas baseadas em aprendizado de máquina (inclui-se aqui técnicas de mineração de dados e deep-learning). O objetivo básico em todas as abordagens e técnicas é verificar se o comportamento observado deriva consideravelmente do comportamento normal esperado. O desafio é identificar com precisão anomalias decorrentes de atividade maliciosas e mitigar falsos positivos.

O tráfego de redes IoT apresenta comportamentos e características estatísticas diferentes das redes convencionais (BIKMUKHAMEDOV R. F., 2019). O tratamento do tráfego agregado (tráfego IoT agregado com tráfego não IoT) pode induzir a análises erroneas e até mesmo mascarar incidentes de segurança (SIVANATHAN A. HASSAN L., 2018). Adicionalmente, numa rede IoT há diferentes dispositivos, que por si só, possuem comportamentos diferentes entre si (NGUYEN; ARMITAGE, 2008), resultando em tráfegos também com características distintas.

Neste sentido, métodos diferentes para detecção de comportamento anômalo de dispositivos IoT têm sido propostos na literatura, assim como métodos para classificação automática de tráfego ou dispositivos IoT. Esta seção detalha alguns destes trabalhos e como eles se relacionam com a proposta desta dissertação.

Em (PROKOFIEV; Y. S. SMIRNOVA, 2018) é proposto um método para detecção da ação de *botnets* sobre dispositivos IoT. Assumindo que os principais alvos de *botnets* que exploram o ambiente IoT são sistemas de câmeras de vigilância IP, e que a maior parte das *botnets* buscam acesso a dispositivos IoT através de ataques de força bruta nos serviços SSH e TELNET dos mesmos, os autores propõem o uso da técnica de regressão logística para estimar a probabilidade de que um dispositivo IoT que está iniciando uma conexão faça parte de uma *botnet*. A regressão logística é um modelo estatístico usado para estimar probabilidades de um evento baseado nos valores de um conjunto de variáveis e é expresso como uma função linear de $n$ variáveis de entrada. Diferente da abordagem proposta nesta dissertação, esta abordagem é de difícil generalização, pois para cada tipo de ataque irá exigir um dado ajuste nos parâmetros do modelo regressivo.

Em (FERRANDO; STACEY, 2017) os autores utilizam uma abordagem estatística para classificar o comportamento de tráfego dos dispositivos IoT, em normal e anormal. O método realiza a análise e correlação temporal de características do tráfego de rede IoT para identificar desvios comportamentais. Baseando-se em uma série temporal do tráfego IoT, o método gera um gráfico 2D do atual comportamento dos dispositivos e compara posteriormente esse gráfico com um novo gráfico gerado a partir de uma série temporal futura. A análise de similaridade entre os dois gráficos é realizada através da medida da distância euclidiana. Caso exista uma discrepância o tráfego é classificado como anômalo. Entretanto, não é claro como a escolha de uma formulação de distância irá afetar a precisão da detecção e a complexidade da computação (HOANG H. DANG, 2019). Visando a redução de complexidade para IoT, outra abordagem estatística foi explorada por (HOANG H. DANG, 2019), onde os autores buscam um cálculo mais eficiente de distância para computo de uma Análise dos Componentes Principais (ACP). A ACP é uma técnica de análise multivariada que realiza a redução de dimensão ao transformar um conjunto inicial de variáveis correlacionadas num conjunto menor de variáveis não correlacionadas. Com foco no uso das técnicas estatísticas, diferentes desse trabalho não foram exploradas as características dinâmicas do tráfego.

A Análise da Quantificação da Recorrência, técnica estatística que permite avaliar siste-

mas dinâmicos, também vem sendo empregada para análises de tráfego de rede com intuito de identificar anomalias. Em (JEYANTHIM; J. VINITHRA SNEHA, 2011) (KUMAR; K. BHARGAVI, 2012) (JEYANTHI; R. THANDEESWARAN, 2014) as características dinâmicas são extraídas do fluxo de rede e os gráficos da recorrência resultantes são analisados, de forma empírica, em busca de distorções visíveis nos valores dessas características, e que configurem uma anomalia. Porém, pontos como execução em tempo real, falsos positivos e eficácia, não são abordados claramente. Em (YUAN; R. YUAN, 2014) as características dinâmicas são analisadas utilizando o algoritmo de clusterização K-means em conjunto com a Transformada Wavelet e a AQR. Mas a utilização de um algoritmo de clusterização com um número fixo de clusters, acaba inferindo na eficiência de classificação. Em (RIGHI M. A.; NUNES, 2019) os autores utilizam a AQR em conjunto com algoritmos de clusterização adaptativa A-Kmeans, para sobrepassar comportamentos dinâmicos não lineares e não estacionários no tráfego de rede para detecção de ataques DDoS. A combinação resultou numa melhor eficácia na detecção de anomalias no tráfego de rede, com um baixo índice de falso positivos. Este trabalho adota esta abordagem, mas especializando a análise no tráfego de rede IoT já diferenciado.

Pelos trabalhos analisados, percebe-se que os esforços voltados para segurança de ambientes IoT, que utilizam dados de redes, estão focados em detectar apenas tráfego anômalos, porém sem identificar os dispositivos responsáveis (INDRE; LEMNARU, 2016) (SOUSA; Z. ABDELOUAHAB D. CICERO P. LOPES, 2017) (FERRANDO; STACEY, 2017) (PROKOFIEV; Y. S. SMIRNOVA, 2018), ou são soluções para tecnologias e ataques específicos (RAZAA; L. WALLGRENA, 2013) (SHEIKHAN; BOSTANI, 2016). Neste sentido, alguns autores advogam que há necessidade de identificação e segmentação do tráfego de ambientes IoT para uma classificação de forma correta (CISCO, 2017) (BIKMUKHAMEDOV R. F., 2019) (SIVANATHAN A. HASSAN L., 2018) (SIVANATHAN A. SHERRATT D.; V., 2017).

Em (SIVANATHAN A. SHERRATT D.; V., 2017) é proposto um método para distinguir dispositivos IoT de dispositivos não IoT. O método analisa traços de rede usando uma abordagem baseada na extração de dados estatísticos que permite identificar e classificar tráfego de dispositivos IoT. Visando caracterizar o comportamento de dispositivos IoT e posteriormente separar o tráfego de dispositivos IoT de não-IoT, os autores apresentam um modelo de classificação utilizando algoritmos de datamining em conjunto com atributos de rede. Em suas observações os autores identificam que a rede atinge picos de tráfego de 17Mbps e mantém uma média de 400Kbps, porém se observado só o tráfego dos elementos IoT o pico de tráfego

fica em torno de 1Mbps e com média de 66Kbps, ou seja, o volume do tráfego IoT observado cai 83.5% em relação ao tráfego observado de forma agregada (IoT e não-IoT), dessa forma salientando o quanto o tráfego agregado de uma rede pode mudar em relação ao tráfego apenas de um ambiente IoT.

Para realiza a diferenciação entre do tráfego, os autores empregam o uso de algoritmos de data-mining Kmeans e Random Forest sobre características de tráfego como: tamanho médio de pacotes, requisições de DNS, intervalo de pacotes DNS e NTP, tempo de inatividade e volume de atividade dos dispositivos, número de serviço e protocolos utilizados. Em um primeiro momento os autores submetem as características de tráfego ao algoritmo Kmeans para identificar quais características descrevem melhor o comportamento dos dispositivos IoT de não IoT, e ao analisar os clusters resultantes os autores listam as características de tráfego na seguinte ordem de importância para diferenciação entre IoT e não-IoT; volume de atividade, intervalo de pacotes DNS, tamanho médio de pacotes, intervalo de inatividade, número de servidores utilizados, protocolos, requisições DNS, Intervalo de comunicação NTP e taxa de atividade dos dispositivos.

Em um segundo momento os autores já conhecendo quais características de tráfego são úteis para mapear dispositivos IoT e não-IoT, utilizam o algoritmo Random Forest para melhorar o desempenho do seu modelo de classificação, além das características utilizadas anteriormente, os autores acrescentam dois novos campos nessa etapa, um campo para a porta mais utilizada pelo dispositivo e um campo apara identificação de dispositivo IoT de não-IoT, sendo os autores o algoritmo resultou em uma precisão de 97% para a identificação e classificação de dispositivos IoT de não-IoT.

Já em (SIVANATHAN A. HASSAN L., 2018), os autores apresentam uma arquitetura de múltiplos níveis para a classificação de dispositivos IoT em tempo real, a arquitetura apresentada opera com dados do tráfego de rede e caracteriza o tráfego dos dispositivos em dois pilares, o padrão de atividade e o padrão de sinalização utilizado na comunicação dos dispositivos IoT. A caracterização do padrão de atividade baseia-se em quatro atributos primários para caracterizar um dispositivos IoT, sendo esses: o tempo de inatividade do dispositivo, o volume do fluxo de dados produzido pelo dispositivo, o tempo de duração do fluxo e, por fim a taxa média de fluxo. Para analisar o padrão de sinalização de um dispositivo durante sua comunicação, os autores analisam dados como: consultas DNS e NTP, o número de portas de comunicação mais utilizadas (tanto local quanto externa). Por fim, os autores analisam o padrão de troca de cifras para

comunicação criptografada entre dispositivos que implementam a comunicação via TLS/SSL[1]. No primeiro estágio da arquitetura apresentada por (SIVANATHAN A. HASSAN L., 2018), é utilizado o algoritmo de Naive Bayes para tratar todos os atributos nominais, sendo eles as consultas DNS, números de portas e cifras de comunicação criptografada, segundo os autores o Naive Bayes foi utilizado por apresentar um bom desempenho na classificação de texto com um grande número de palavras. Ao final o classificador retorna um valor numérico, representando a probabilidade de uma dada palavra (contida em um dos atributos) pertencer a uma classe de dispositivo. O segundo estágio da arquitetura é destinado ao tratamento dos demais atributos junto com os valores originados na primeira etapa, porém nesse segundo estágio é utilizado o algoritmo de Random Forest para classificar um determinado dispositivo. Para testar a precisão da arquitetura os autores utilizam um conjunto conhecido de dispositivos, desses dispositivos é gerando um conjunto de instancias dos atributos de redes utilizados pela arquitetura, 70% das instancias geradas são utilizadas para o treinamento dos algoritmos, os 30% restantes são utilizado para testar e aferir o funcionamento da arquitetura, ao final os autores alcançam uma precisão de 99.88%.

A heterogeneidade de conectividade e de padrão de tráfego dos dispositivos IoT adicionam complexidade na tarefa de detecção de anomalias em ambientes IoT e foram tratadas em (SANTOS M., 2017) com uma proposta de ferramenta de monitoramento de rede que explora a classificação e identificação de tráfego de dispositivos IoT. A ferramenta proposta pelos autores realiza a análise do tráfego em dois estágios. No primeiro estágio é feita uma análise passiva da rede, aonde é coletado e extraído dados da rede como lista de portas, GeoIP, caracterização do tráfego e URL's presentes nos pacotes capturados. O segundo estágio é destinado para a classificação do tráfego. Segundo os autores, a arquitetura não está vinculada a um tipo específico de classificador, dessa forma a ferramenta é livre para ser adaptada para utilizar classificadores de tráfego conforme demanda.

Diferente dos demais, neste trabalho a metodologia proposta combina metodologias estatísticas e de aprendizado de máquina bem estabelecidas para dados de tráfego de rede, a fim de detectar um comportamento anormal dos dispositivos IoT. Em (SIVANATHAN A. SHERRATT D.; V., 2017) a classificação do tráfego de rede é visando separar o tráfego IoT de não-IoT, já em (SIVANATHAN A. HASSAN L., 2018) o objetivo dos autores é modelar e classificar o tráfego de dispositivos de forma individual, neste trabalho o processo de classificação do trá-

[1] Transport Layer Security/Secure Sockets Layer- RFC: https://tools.ietf.org/html/rfc5246

fego do ambiente IoT tem o foco de separar os dispositivos IoT de acordo com sua função e comportamento, uma vez identificados os grupos de dispositivos do ambiente IoT, a detecção de anomalias no tráfego é detectada através da metodologia estatística de análise quantitativa da recorrência.

## 2.5 CONSIDERAÇÕES PARCIAIS

Esse capítulo concentra as revisões teóricas dos temas abordados ao longo desse trabalho, expondo as definições que moldam o conceito de Internet das Coisas, a sua composição, a forma de comunicação desse tipo de rede. Foi realizado também uma síntese sobre como são classificadas as ameaças de segurança e descrito os principais ataques presentes no cenário de redes IoT. O capítulo também apresentou as definições de análise da quantificação da recorrência, detalhes de como é gerado o gráfico de quantificação da recorrência e como esse gráfico é analisado através de medidas de quantificação da recorrência. Finalmente, foram apresentados os trabalhos relacionados ao proposto nessa dissertação.

# 3 DETECÇÃO DE DISPOSITIVOS ANÔMALOS UTILIZANDO AQR E CA

Este capítulo propõe o método AIDA (*Anomaly Internet of Things Detection Alert*), um método para detecção de dispositivos IoT anômalos. O método utiliza a combinação de Análise da Quantificação da Recorrência (AQR) e Clusterização Adaptativa (CA) aliada ao conceito de segmentação de dispositivos IoT por classes de comportamentos.

O capítulo está estruturado da seguinte forma: a Seção 3.1 apresenta uma discussão e modelagem do problema abordado, bem como a forma básica da solução proposta (funcionamento básico do método AIDA); a Seção 3.2 detalha a primeira fase do método AIDA, de coleta e segmentação do tráfego de dispositivos por classe comportamental; a Seção 3.3 apresenta os detalhes da estrutura de detecção baseada em AQR/CA; e a Seção 3.4 traz uma síntese e as considerações parciais dos temas abordados ao longo deste capítulo.

## 3.1 MODELAGEM DO PROBLEMA/SOLUÇÃO

Considere uma rede IoT $C$ composta por conjuntos $c_i$ de dispositivos eletrônicos $d$, interligados e em diferentes configurações. Os conjuntos $c_i$ são distintos entre si e cada $c_i$ opera sobre uma certa tecnologia, desempenhando uma determinada funcionalidade $f$. O funcionamento de uma rede IoT durante um período de tempo $t$, gera uma série temporal não estacionária de tráfego $S$, aonde $S$ é formada pela agregação dos tráfegos $s_i$ gerados por cada conjunto $c_i$, de forma que $S = \{s_1, s_2, .., s_N\}$ para todo $1 \leq i \leq N$, onde $N$ corresponde ao número de conjuntos de dispositivos com diferentes comportamentos. Devido a variedade de tecnologias, aplicações e funcionalidades de cada $c_i$, as suas respectivas séries de tráfego $s_x$ apresentam características comportamentais distintas umas das outras.

Mesmo diante de comportamento heterogêneo das séries $s_i$, existem características $K$ do tráfego que tendem a se repetir ao longo do tempo de operação de uma rede IoT (DOSHI; N. APTHORPE, 2018). Essas características $K$ podem ser inspecionadas através do método de análise quantitativa da recorrência (RIGHI; NUNES, 2016). Para identificar anomalias no tráfego $s_i$, o método AQR analisa medidas da recorrência, aqui representadas por $M$. Para identificar as anomalias é preciso comparar as medidas $M$ com uma série $O$ de características recorrentes, extraídas de um tráfego de um conjunto $c_i$ de dispositivos livre de anomalias. A diferença de valores entre essas medidas, quando significativa, indica a presença de anomalias

$A$.

Em síntese, pode-se formalizar a composição de uma rede IoT pela equação 3.1, ou seja, a rede $C$ corresponde a todos os conjuntos $c_i$ de dispositivos, independente de suas funcionalidades, onde $N$ indica o número de conjuntos distintos.

$$C = \sum_{i=1}^{N} c_i \tag{3.1}$$

De maneira similar, a equação 3.2 retrata o comportamento dinâmico da rede através das suas medidas de quantificação da recorrência $M$, dada as séries $s_i$ de tráfego IoT e suas características $K$.

$$M = \sum_{i=1}^{N} K * s_i \tag{3.2}$$

Finalmente, a equação 3.3 representa a condição do diagnóstico de anomalias em um tráfego IoT, ou seja, a de que os valores de $M$ resultantes da equação 3.2 relativa a um tráfego sob análise devem diferir (considerando um dado limiar) dos valores pré-calculados $O$ de um tráfego IoT modelo.

$$A \rightarrow O \neq M \tag{3.3}$$

Neste contexto, assumindo a presença de diferentes dispositivos e diferentes padrões de tráfego na rede IoT, o método AIDA (*Internet of Things Anomaly Detection Alert*) corresponde a um detector de anomalias comportamentais que opera em duas fases: ($i$) coleta e segmentação da rede IoT em conjuntos $c_i$, onde é realizada a identificação dos comportamentos de rede de cada conjunto $c_i$ de dispositivos em uma situação livre de anomalias; e ($ii$) avaliação em tempo de execução, via Análise da Quantificação da Recorrência e Clusterização Adaptativa, se um dispositivo $d$ de um dado conjunto $c_i$ está se comportando como esperado. A análise de anomalias, após o processo de segmentação, permite uma detecção mais direcionada por segmento de dispositivos e consiste numa das principais contribuições deste trabalho.

## 3.2 MÓDULO COLETA E SEGMENTAÇÃO

O Módulo Coleta e Segmentação é o primeiro módulo presente no AIDA e está ilustrado na Figura 2. Esse módulo deve ser posicionado junto ao *gateway* da rede IoT, pelo fato do

*gateway* ser um elemento de rede mais robusto (maior capacidade computacional e de armazenamento) e um concentrador de comunicações de redes IoT. Na Figura 2, a coleta via *gateway* está ilustrada pelo bloco inferior, que mostra a concentração de fluxos de dispositivos sobre um dispositivo de rede.

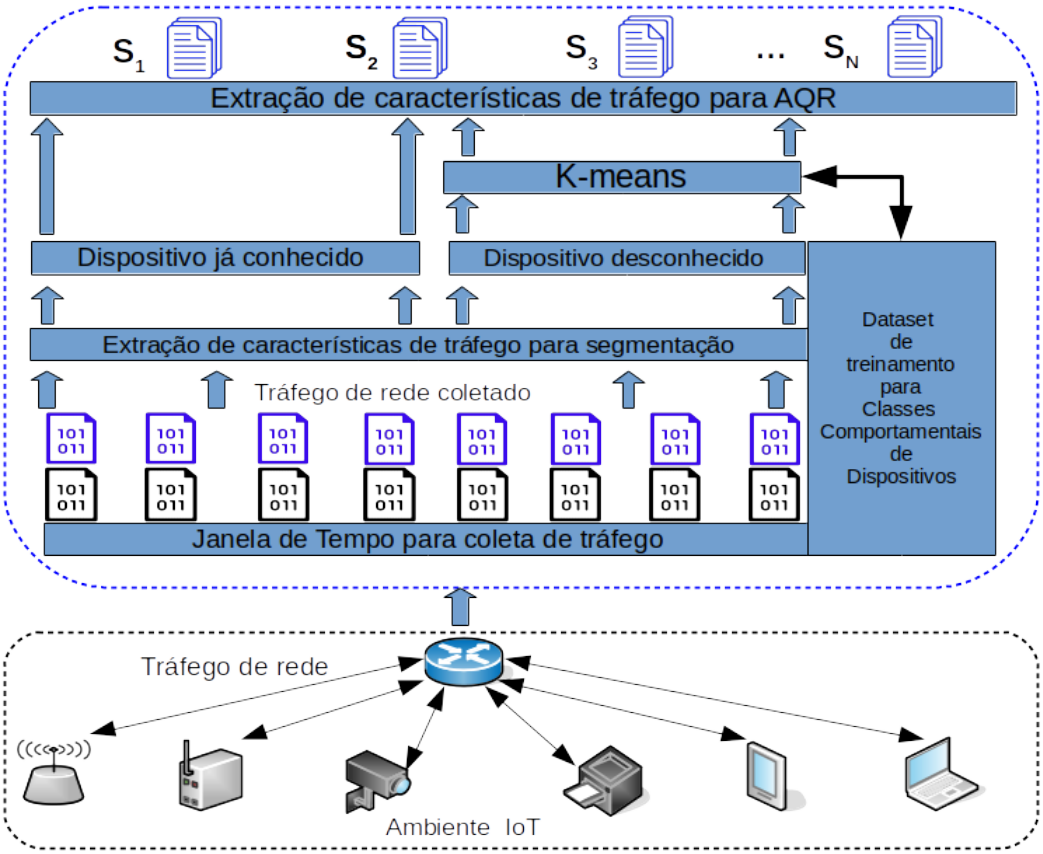


Figura 2 – Arquitetura do Módulo Coleta e Segmentação do AIDA

Fonte: o próprio autor

A função básica do Módulo Coleta e Segmentação é coletar o tráfego bruto de rede IoT (bloco inferior da Figura 2), extrair atributos de interesse e convertê-los em séries de dados organizados para serem utilizadas no processo de segmentação (identificação da classe comportamental do dispositivo) e no processo de detecção (classificação do comportamento normal ou anômalo).

A fase de coleta de dados corresponde a captura do tráfego da rede do ambiente IoT. Essa captura é efetuada durante uma janela de tempo previamente estabelecida. A janela de tempo deve ter um tempo de duração que possibilite capturar o tráfego de um ciclo completo de operação de todos os dispositivos da rede IoT. Dessa forma, o tráfego capturado terá traços suficientes para modelar e classificar, com maior precisão, os dispositivos presentes na rede.

Cabe ressaltar que o tamanho da janela de tempo pode ser diferente para cada ambiente IoT.

Do tráfego amostrado são extraídos dois conjuntos de atributos: um que é usado para determinar as classes dos dispositivos (segmentação) e outro para ser usado na análise de comportamento anômalo (detecção). Cada série de dados resultante do Módulo Coleta e Segmentação (séries $S_1$, $S_2$, $S_3$, ..., $S_n$ da Figura 2), e que será usada pela AQR, pertence a um dispositivo já associado a uma classe de dispositivos IoT presentes na rede analisada. Quando o dispositivo não é conhecido, aplica-se o processo de segmentação usando um classificador *k-means*.

Em síntese, o Módulo Coleta e Segmentação tem duas tarefas principais. A primeira é a de segmentação do tráfego da rede IoT, aonde é identificado a classe comportamental de um dispositivo IoT, e a segunda é a extração de dados por dispositivo de classe comportamental do tráfego coletado. Essas funções são detalhadas a seguir.

### 3.2.1 Segmentação do Tráfego por Classe Comportamental

Para identificar a qual classe $c_i$ um determinado dispositivo IoT $d$ pertence é necessário modelar o comportamento das classes através de atributos relacionados ao tráfego de rede. Deste modo, a Seção 3.2.1.1 detalha o modelo comportamental considerado, a Seção 3.2.1.2 detalha como os atributos são extraídos para a segmentação e a Seção 3.2.1.3 detalha o método proposto para identificação da classe dos dispositivos IoT (segmentação).

#### *3.2.1.1 Modelo de Classes comportamentais de ambientes IoT*

Um fator importante do método AIDA é a definição de um método para identificação e segmentação de conjuntos de dispositivos IoT, de acordo com o seu comportamento, e a seleção das características do tráfego de cada conjunto e respectivos dispositivos. No AIDA, o método de segmentação e reconhecimento de conjuntos de dispositivos IoT é baseado na diferenciação de tráfego, ou seja, na classificação dos dispositivos IoT, em classes, de acordo com o seu padrão de tráfego.

De acordo com (T. GARRETT S. DUSTDAR, 2018), o tráfego/comunicação de um dispositivo numa rede IoT pode ser divido em três classes: periódico, *streaming* e eventual. Na classe de comunicação periódica estão os dispositivos que geram tráfego em intervalos *t* regulares e normalmente com pacotes de tamanho constante. Na classe de comunicação eventual estão todos aqueles dispositivos que iniciam comunicação com base em alguma ação ou evento no meio em que estão localizados, fazendo com que o intervalo de tempo *t* entre os períodos de

comunicação varie de maneira pouco previsível; e (3) na classe de *streaming* estão os dispositivos que têm um fluxo constante de comunicação em relação ao tempo *t*, porém apresentando intervalos sem comunicação entre os dispositivos. A Figura 3 ilustra as classes comportamentais, sendo que a Figura 3(a) ilustra a classe de comunicação periódica, a Figura 3(b) ilustra a classe de comunicação eventual e a Figura 3(c) ilustra a classe de *streaming*.

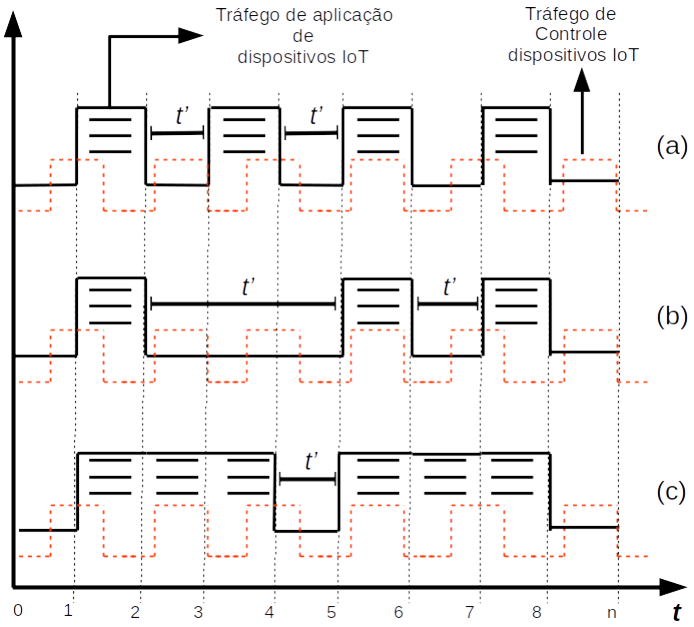


Figura 3 – Padrão de comunicação de classes de dispositivos IoT. (a) Periódico, (b) Eventual e (c) *streaming*

Fonte: o próprio autor

Considerando as classes de tráfego periódico e *streaming*, a Figura 4 apresenta um gráfico com o tráfego de duas classes de dispositivos e exemplifica como o tráfego, quando observado de forma agregada, tem características diferentes em relação as características do tráfego analisadas por classe comportamental. O gráfico representa o fluxo de dispositivos de tráfego *streaming* (linha com marcação triangular), periódico (linha com marcação circular) e o tráfego agregado do ambiente IoT (linha com marcação em quadrados). A base de dados utilizada para gerar o gráfico da Figura 4 foi fornecida pela *School of Electrical Engineering and Telecommunications* da *UNSW Sydney*[2] (A. SIVANATHAN H. HABIBI GHARAKHEILI; SIVARAMAN, 2018) e é melhor detalhada na Seção 4.1. Pela figura observa-se que uma análise com base no tráfego agregado pode dificultar significativamente a identificação de um comportamento anômalo de um dispositivo da classe de comunicação periódica. Esta constatação reforça a hipótese

[2] Disponível em: https://iotanalytics.unsw.edu.au/iottraces.html

de que a detecção de anomalias após segmentação do tráfego é mais eficaz.

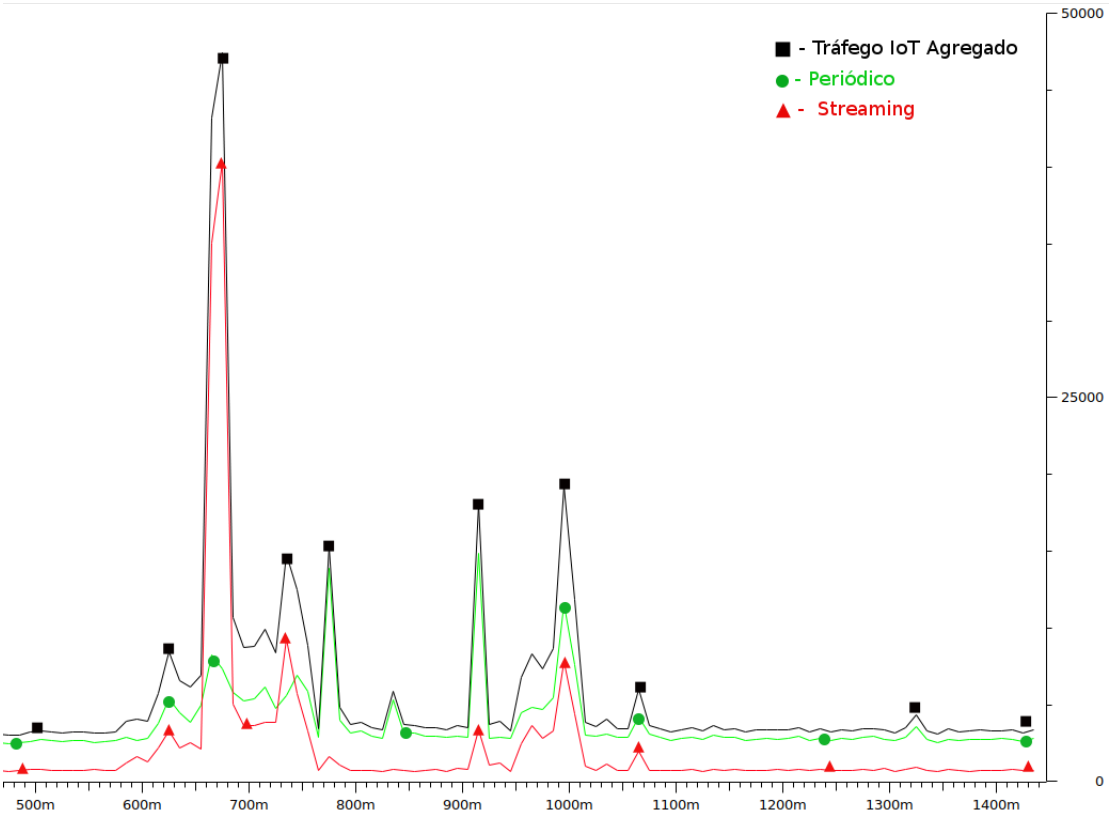


Figura 4 – Comportamentos de fluxo de pacotes de dispositivos com tráfego *streaming* (triângulos) e periódico (círculos), e o fluxo de tráfego agregado de todo o ambiente IoT (quadrados)

Fonte: o próprio autor

A identificação dos atributos de rede que melhor identificam as classes de dispositivos numa rede IoT está fora do escopo deste trabalho. Por outro lado, a literatura (T. GARRETT S. DUSTDAR, 2018) (MIETTINEN; S. MARCHAL I. HAFEEZ, 2017) (BAI; L. YAO S. KANHERE S., 2018) (FERRANDO; STACEY, 2017) aponta que a observação de características como a taxa de pacotes enviados, tamanho médio de pacotes, protocolo e portas utilizadas, bem como a matriz origem/destino de comunicação de endereços IP da rede IoT, é suficiente para identificar a qual classe um determinado dispositivo IoT pertence. Segundo (SIVANATHAN A. SHERRATT D.; V., 2017), este conjunto reduzido de atributos permite identificar a classe do dispositivo porque os dispositivos IoT tem um padrão de comunicação limitado (em média com menos de dez endereços, correspondentes aos seus servidores de aplicação). Este trabalho adota estes atributos como sendo os descritores do modelo comportamental da rede IoT e seus dispositivos.

#### *3.2.1.2 Extração de Atributos para a Segmentação*

Conhecido o conjunto de atributos de interesse, para identificar a qual classe $c_i$ um determinado dispositivo IoT $d$ pertence, primeiramente devem ser identificadas a taxa de pacotes enviados $c_{i_{Pac_{send}}}$, o tamanho médio de pacotes $c_{i_{PacSize_{avg}}}$, os protocolos $c_{i_{Proto}}$ e as portas utilizadas $c_{i_{Port}}$, os quais devem representar o comportamento de cada classe $c_i$ em que será organizado e classificado os dispositivos presentes na rede. Em seguida, dada uma determinada janela de tempo do tráfego de uma rede IoT, para todo e qualquer dispositivo verifica-se se seu tráfego apresenta valores semelhantes para a taxa de pacotes enviados $Pac_{send}$, o tamanho médio de pacotes $PacSize_{avg}$, o protocolo $Proto$, as portas utilizadas $Port$ e os endereços IP de destino $d_{dst}$ por conjunto $c_{i_{dst}}$ de endereços IP. As equações 3.4 e 3.5 representam o modelo de segmentação adotado.

$$X\left(Pac_{send} \approx c_{i_{Pac_{send}}} \wedge PacSize_{avg} \approx c_{i_{PacSize_{avg}}} \wedge Proto \approx c_{i_{Proto}} \wedge Port \approx c_{i_{Port_{Num}}}\right) \tag{3.4}$$

$$d \in c_i \forall \left(d_{dst} \in c_{i_{dst}} \wedge X\right) \tag{3.5}$$

Cabe ressaltar que dispositivos de diferentes fabricantes e com diferentes configurações, podem desempenhar uma mesma funcionalidade $i$, como por exemplo um sistema de monitoramento pode ter diferentes modelos de câmeras de diferentes fornecedores e cada fabricante uma faixa de endereços IP própria. Diante desse fato, um conjunto IoT $c_i$ admite como parte de sua configuração a comunicação de seus dispositivos com um grupo de diferentes endereços IP $c_{i_{dst}}$.

Na prática, no AIDA assume-se que o modelo para identificar a classe comportamental de um dispositivos IoT utiliza os seguintes atributos do tráfego de rede: Taxa de pacotes enviados (Tx_pac_send), Taxa de pacotes recebido (Tx_pac_reciv), Protocolo utilizado (Protocolo_Type), Tamanho médio dos pacotes (Pac_size_avg), Porta de Origem (Src_port) e Porta de Destino (_port). Esses atributos são extraídos pelo método AIDA como segue.

##### 3.2.1.2.1 Taxa de pacotes Enviados/Recebidos

A taxa de pacotes enviados ou recebidos por um dispositivo IoT baseia-se na contabilidade do montante de pacotes de rede que o dispositivo opera durante uma comunicação ao

longo de um período observado de tempo *t'*. O comportamento de uma classe de dispositivos, em parte, pode ser identificado através da frequência de volume de pacotes de aplicação produzidos por um dado dispositivo em um tempo *t'* ao longo de uma janela temporal *t*. Como observado por (A. SIVANATHAN H. HABIBI GHARAKHEILI; SIVARAMAN, 2018), dispositivos IoT podem produzir tráfego autônomo distinto da sua aplicação original, como por exemplo o tráfego de pacotes de NTP e/ou DNS utilizado para controle ou sincronia do dispositivo.

Dessa forma, ao analisar o comportamento de um dispositivo durante uma janela temporal *t*, verifica-se que se a taxa de pacotes de aplicação contida por cada tempo *t'* apresentar variações regulares em seu volume entre *t'* e *t'*+1 ao longo de *t*, indica que o dispositivo comunicante tem um comportamento periódico, conforme representado na Figura 3(a). Em casos que a taxa de pacotes contida por cada tempo *t'* apresentar variações assimétricas em seu volume contidos no tempo *t'* em relação ao tempo *t'*+1, podendo ter períodos consecutivos de *t'* sem tráfego de aplicação, o dispositivo em questão indica um comportamento por gatilho/evento, conforme exemplificado na Figura 3(b). Caso a taxa de pacotes de aplicação apresentada em cada período observado de tempo *t'* se manter constante ao longo de *t*, trata-se de um dispositivo com um comportamento de *streaming*, conforme exemplificado na Figura 3(c).

#### 3.2.1.2.2 Protocolo utilizado

O Protocolo de comunicação utilizado é um parâmetro empregado como forma de auxiliar o processo de identificação da classe comportamental de um dispositivo. Como observado por (SIVANATHAN A. SHERRATT D.; V., 2017) haverá protocolos dominantes na comunicação em um ambiente IoT. O protocolo dominante normalmente será para os dispositivos com um maior fluxo de comunicação ou com maior número de dispositivos presente no ambiente. Segundo (E. NTHI L. WILLIAMS, 2019) dispositivos com uma comunicação frequente tendem a utilizar protocolos como UDP, como por exemplo uma câmera de vigilância, já dispositivos com uma comunicação eventual e periódica tendem a utilizar protocolos de mais robustos como TCP, pois normalmente esses dispositivos desempenham funções que requerem confiança recebimento de suas mensagens, como por exemplo sensores de movimento e monitores de saúde, ar e alarmes de incêndio.

#### 3.2.1.2.3 Tamanho médio dos pacotes

O tamanho médio de pacote, gerado por um dado dispositivo, auxilia em sua classificação, como observado em (SIVANATHAN A. SHERRATT D.; V., 2017). Dispositivos com uma comunicação, de aplicação, tipicamente periódica, como por exemplo uma estação de meteorologia, que enviam relatórios a uma central ou consultam um dado endereço para sincronizar informações, tendem a manter um tamanho padrão e mais elevado para o seus pacotes, tendo em vista que os mesmos utilizam protocolos mais robustos e com mais campos utilizados para construção do pacote de rede. Outros dispositivos que mantém uma frequência de comunicação maior, em relação aos de comportamento periódico, tendem a ter pacotes com tamanho menor, dado que os mesmos utilizam protocolos com menos campos utilizados para construção do pacote de rede.

#### 3.2.1.2.4 Portas de Origem/Destino

O atributo de número de portas auxilia para classificação do tráfego de dispositivos a nível de camada de aplicação. Esse atributo é utilizado em (SIVANATHAN A. HASSAN L., 2018) como descritor dos protocolos utilizados pelo dispositivo na camada de aplicação. Segundo os autores, o fato dos dispositivos, em sua maioria, se comunicarem apenas com um número reduzido de portas, a análise desse atributo auxilia na identificação do tráfego de aplicação e tráfego de controle. Por exemplo, o tráfego de DNS utiliza normalmente a porta 53 e o tráfego NTP utiliza a porta 123.

Devido ao range de possibilidades para os valor das portas que podem ser adotadas para as aplicações IoT, é preciso organizar em classes os números das portas. A organização para os valores de portas está expresso na Tabela 1 e seguem o molde de mapeamento realizado por (MIETTINEN; S. MARCHAL I. HAFEEZ, 2017). Os valores utilizados para a representação das classes de portas foram definidos com um intervalo alto entre si, visando acentuar o resultado do cálculo para a definição da classe de portas mais comuns ao longo de uma comunicação (vide Equação 3.6).

Visando mapear a dinâmica das portas utilizadas na comunicação dos dispositivos IoT, foi empregada uma técnica para a identificação da classe de porta mais utilizada por um dispositivo, em seu tráfego, ao longo de um período *t'*. Para cada pacote do dispositivo IoT capturado no período *t'* observado, identifica-se a porta de comunicação e em seguida soma-se o valor de

representação da classe ($X_{ClassePorta}$) referente a porta identificada. Ao final do período *t'* o valor resultante do somatório ($\sum$) do valor de representação da classe ($X_{ClassePorta}$) é dividido pelo número total de pacotes capturados ($X_{NumeroPacoteDoDispositivo}$) ao londo de *t'*. O resultado da divisão é comparado com os valores de representação de classes (vide Tabela 1). A classe com o valor mais próximo do valor obtido pela equação da divisão será a classe que representa as portas mais utilizadas por um dispositivo ao longo do período *t'*.

A técnica utilizada para a identificação de portas é representada pela Equação 3.6. Com base nessa organização, em cada tempo de amostragem *t'* para o atributo de porta de origem e destino será utilizado um valor correspondente a classe de valores de porta que for de maior recorrência dentro do período de *t'* observado.

Tabela 1 – Organização para valores de portas de comunicação

| Grupo | Portas | Valor de Representação para Classe de portas |
|---|---|---|
| Sem portas | - | 1000 |
| Portas controle | 0-79 | 5000 |
| Portas conhecidas | 80-1013 | 25000 |
| Portas registradas | 1024-49151 | 125000 |
| Portas dinâmicas | 49152-65535 | 750000 |

$$t'_{Porta} = \left( \frac{\sum_{i=X_{ClassePorta}}^{n}}{X_{NumeroPacoteDoDispositivo}} \right) \approx ClassePorta \tag{3.6}$$

#### *3.2.1.3 Método de Identificação de Classe Comportamental de Ambientes IoT*

Uma vez definida as classes de dispositivos presentes na rede IoT, o processo de segmentação é divido em três partes: treinamento, descoberta e classificação. O Algoritmo 1 descreve

o processo de segmentação adotado no método AIDA.

**Algoritmo 1:** Algoritmo para a Segmentação de Dispositivos em Classes

**Entrada:** Tráfego de rede IoT
**Saída:** Classificação de anomalias em dispositivos IoT

```
1  início
2      Define dataset de treinamento, com limiares e dispositivos de cada classe
3      Define atributos de tráfego a ser extraídos
4      Define tamanho para janela de tempo usada na coleta do tráfego
5      enquanto existir tráfego na rede IoT faça
6          Coleta tráfego do ambiente IoT durante janela de tempo
7          Armazena janela de tempo do tráfego capturado
8          if Existe dispositivo desconhecido then
9              para cada dispositivo desconhecido, não indicado no dataset faça
10                 Extrai x ← características do tráfego do dispositivo
11                 Extrai centroids ← Kmeans(x, num_clusters)
12                 Armazena limiares para classificação de dispositivos (centroids)
                    no dataset
13                 Classifique dispositivo de acordo com o centroid das classes
                    Atualize dataset
14             fim
15         else
16             Inicia a extração de atributos do tráfego de rede para AQR
17             para cada classes de dispositivos dataset faça
18                 Extração das características de tráfego de cada dispositivo da classe
                    em análise
19                 S← Características extraídas do tráfego
20             fim
21         end
22     fim
23 fim
24 retorna S (Série de dados organizado para AQR)
```

Em uma primeira fase é preciso treinar os algoritmos responsáveis por classificar os dispositivos do ambiente IoT. Esse processo leva em conta informações como, período de comunicação e tamanho de pacotes trocados por um dispositivo dentro de uma janela de tempo estabelecida durante a fase de experimentação. Como resultado é gerado o *dataset* utilizado como guia para classificação dos dispositivos presentes na rede e para a extração dos parâmetros utilizados para o processo de criação dos gráficos da recorrência no módulo AQR.

Para a construção do *dataset*, primeiramente é preciso identificar quais classes comportamentais, e seus respectivos dispositivos IoT, que estão presentes na rede. Em seguida é preciso coletar uma amostragem do tráfego dos dispositivos de cada classe, e extrair os atributos utilizados no processo de segmentação (vide Tabela 2). Utilizando os atributos extraídos e o algoritmo K-means, são criados os clusters que representam as classes comportamentais da rede

e seus respectivos valores de *centroids* para cada cluster. Ao final desse processo de construção o *dataset* será composto por: uma identificação para cada uma das classes comportamentais do ambiente IoT, uma lista dos endereços MAC dos seus dispositivos e o valor do *centroid* do cluster que representa a classe da rede IoT.

A próxima etapa é definir o tamanho da janela de tempo utilizada para a captura do tráfego de rede. Esse parâmetro define quanto tempo o tráfego de rede é coletado antes de ser extraído seus atributos para o processo de segmentação. Devido a heterogeneidade do ambiente IoT, o tamanho da janela de tempo para coleta de tráfego é variável para cada ambiente IoT e deve ser definida a partir da análise do administrador da rede.

Estabelecido os parâmetros de treinamento, classificação e coleta, tal como representado nas linhas 2 a 4 do Algoritmo 1, inicia-se o processo de captura do tráfego, esse processo (linhas 5 a 22 do Algoritmo 1) é executado enquanto existir tráfego da rede monitorada. Como exemplificado nas linhas 6 e 7 do Algoritmo 1, ao completar o período de cada janela de tempo, é gerado um arquivo temporário com o conteúdo do tráfego capturado. Dessa forma é possível analisar o tráfego já capturado e, simultaneamente, coletar uma nova amostragem de tráfego. Antes de iniciar o processo de segmentação e classificação do tráfego capturado, verifica-se a existência de endereços MAC de dispositivos desconhecidos no ambiente IoT, previamente classificado, ou seja, de dispositivos que não estão atrelados a nenhuma classe descrita no *dataset* de treinamento (linha 8 do Algoritmo 1). Caso exista dispositivos desconhecidos, inicia-se o processo de classificação do mesmo, com o objetivo de identificar a qual classe esse dispositivo pertence, conforme descrito no bloco das linhas 9 a 14 do Algoritmo 1.

Para classificar um dispositivo desconhecido, inicialmente é extraído o conjunto de atributos de segmentação do tráfego (vide Tabela 2). Esses atributos são armazenados e posteriormente submetidos ao processo de clusterização utilizando o algoritmo K-means. Os valores dos *centroids* resultantes da clusterização são comparados com os valores dos *centroids* das classes comportamentais registradas no *dataset* de treinamento. O dispositivo será atrelado a classe que tiver o valor de *centroid* mais próximo ou igual ao valor do *centroid* do dispositivo. Após identificado a classe comportamental (periódico, *streaming* ou eventual) que o dispositivo pertence, o mesmo é registrado no *dataset*.

Salienta-se que realizar a identificação da classe de um dispositivo antes de iniciar o processo de segmentação do tráfego e extração dos atributos para AQR, possibilita que o tráfego gerado por esse dispositivo seja analisado mais precisamente pelo Módulo AQR, potenciali-

zando uma detecção mais precisa.

Uma vez que todos dispositivos no tráfego coletado estejam atrelados a uma classe conhecida, a próxima etapa (represetada ) consiste na extração dos atributos do tráfego para o processo de AQR (vide Tabela 2). Para cada classe conhecida, a partir dos dispositivos, serão extraídos os atributos do tráfego e organizados em uma série de dados. Essas séries de dados serão utilizadas pelo segundo módulo do método AIDA, e o tamanho da série de dados pode variar para cada classe de dispositivos da rede IoT, uma vez que essas séries devem conter dados suficientes que possibilitem ao módulo de AQR gerar os gráficos da recorrência (comportamento recorrente) de cada classe.

### 3.2.2 Extração de Atributos

Dispositivos IoT, tem uma matriz de endereços de comunicação origem/destino e modos de operação limitados, de forma que quando infectados por *malwares* ou ingressam em uma *botnet*, esses dispositivos adotam comportamentos não executados anteriormente, e passam a gerar comunicações com endereços não utilizados em seu modo de operação original. No AIDA, cada janela de tempo de tráfego capturado contém todo o fluxo de pacotes da rede e é armazenada em um arquivo (pcap). Esses arquivos são utilizados na segunda etapa do módulo de coleta, aonde todos os atributos de rede necessários para o funcionamento do método AIDA, são extraídos. Uma vez extraídos, esses atributos permanecem armazenados em tempo de execução, até serem totalmente analisados pelo módulo AQR do método AIDA.

Como discutido, o método AIDA opera com a análise de dois conjuntos distintos de atributos de rede, um primeiro conjunto de atributos é utilizado para a segmentação do tráfego (vide Seção 3.2.1.3), e um segundo conjunto de atributos que é utilizado pelo módulo de AQR para a extração das medidas quantitativas da recorrência.

O conjunto de atributos utilizado pelo método AIDA para a extração das MQR's (medidas quantitativas da recorrência) visa fornecer dados para o detector de anomalias qualificar uma determinada situação como anômala. A escolha desses atributos está fora do escopo deste trabalho e ocorreu com base nos atributos elencados nos trabalhos presentes na literatura (SINANOVIC; MRDOVIC, 2017) (KOLIAS CONSTANTINOS KAMBOURAKIS, 2017) (SPOGNARDI ANGELO DONNO, 2017), aonde os autores identificam que a ação de *malwares* ou agentes maliciosos em dispositivos IoT refletem diretamente no padrão desses atributos de tráfego. Os atributos coletados são: Dst_ip , Open_ip, Dst_port , Packet_size e Protocol_Type,

Open_src_ports, Other_ports_req, Source_bytes Destination_bytes e Service e Protocol_Type. Esses atributos estão listados na Tabela 2, com indicação de emprego na AQR (análise da quantificação da recorrência). A Tabela 2 inclui também os atributos empregados no processo de segmentação, a fim de sintetizar os atributos empregados pelo método AIDA.

Definido os atributos, a função do Módulo Coleta e Segmentação é gerar as séries de dados para cada dispositivo conhecido (pertencente a uma classe) e fornecer as séries ao Módulo AQR.

| Atributo | Descrição | Função |
|---|---|---|
| Dst_port | Portas de destino | Segmentação |
| Open_src_ports | Portas de origem | Segmentação |
| Packet_size | Tamanho médio de pacotes | Segmentação / AQR |
| Protocol_Type | Protocolos de conexão | Segmentação |
| Source_bytes | Número de bytes de dados enviados pelo endereço IP de origem | AQR |
| Destination_bytes | Número de bytes de dados enviados pelo endereço IP de destino | AQR |
| Tx_pac_send | Taxa de pacotes enviados | Segmentação / AQR |
| Tx_pac_rec | Taxa de pacotes recebidos | Segmentação / AQR |

Tabela 2 – Atributos empregados no método AIDA

## 3.3 MÓDULO AQR

O segundo módulo do método AIDA é o módulo de AQR ilustrado na Figura 5. Esse módulo é responsável por gerar os gráficos da recorrência, dos quais são extraídas as medidas quantitativas da recorrência (MQR). É preciso fornecer uma série de parâmetros para o algoritmo responsável por analisar e gerar um gráfico da recorrência, como Taxa de raio da vizinhança, Dimensão da Imersão e Tempo de atraso (vide Seção 2.3). Esses parâmetros são definidos a partir da análise de série de dados, obtidos de cada classe de dispositivos presente no *dataset* de treinamento (definido no primeiro módulo). Os valores para cada um desses parâmetros pode variar para cada uma das classes de dispositivos.

Definidos os parâmetros para criação dos gráficos da recorrência (Função AQR na Tabela 2), refletidos nas séries temporais de dados criados pelo módulo de coleta, o Módulo AQR gera um gráfico da recorrência para cada um dos atributos e, posteriormente, para cada gráfico, são extraídas as seguintes características dinâmicas (MQR): Razão de Recorrência (RR), Entropia de Shannon (ENTR) e Determinismo (DET). Esse processo se repete para cada uma das

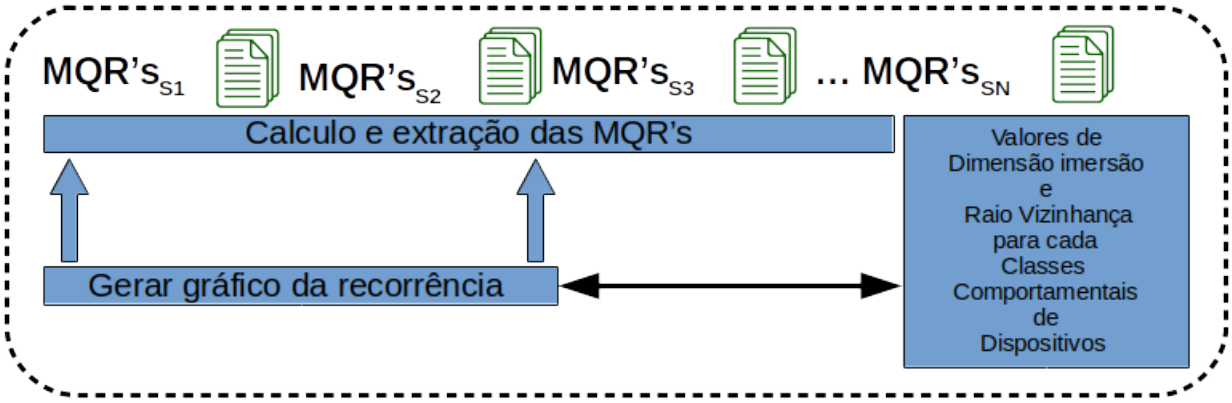


Figura 5 – Segundo módulo da arquitetura do método AIDA para detecção de anomalias em ambientes IoT

Fonte: o próprio autor

classes de dispositivos presentes na rede IoT.

Este método de análise via RR, ENTR e DET foi empregado por (RIGHI; NUNES, 2016) para análise do tráfego de rede não IoT e detecção de ataques DDoS e é reutilizado como base do mecanismo de detecção no método AIDA. No AIDA os cálculos de quantificação de cada MQR, efetuados sobre o gráfico da recorrência, são executados conforme as Equações 3.7, 3.8 e 3.9.

- **Razão de Recorrência (RR)** - determina o valor da densidade dos pontos de recorrência presentes no Gráfico da Recorrência, onde $N$ é o número total de pontos de recorrência possíveis e $R_{i,j}$ é a equação da recorrência.

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j} \tag{3.7}$$

- **Determinismo (DET)** - divisão entre o total de pontos de recorrência que formam estruturas diagonais e o conjunto de pontos de recorrência. Na Equação, $P(l)$ representa a probabilidade de existência de linhas diagonais com comprimentos $l$.

$$DET = \frac{\sum_{l=l_min}^{N} lP(l)}{\sum_{i,j=1}^{N} R_{i,j}} \tag{3.8}$$

- **Entropia de Shannon (ENTR)** - corresponde a distribuição de frequências dos comprimentos das linhas diagonais acima de um dado comprimento mínimo $l_min$ e demonstra o

grau de complexidade da estrutura determinística exposta no sistema.

$$ENT = \sum_{l=l_min}^{N} p(l) \log_2 p(l)$$
$$p(l) = \frac{P(l)}{\sum_{l=l_min}^{N} P(l)} \tag{3.9}$$

- **Comprimento médio das linhas diagonais (L)** - corresponde a media do comprimento das linhas diagonais de um gráfico da recorrência, sendo $l$ o comprimento das linhas diagonais, $P(l)$ o histograma do gráfico da recorrência recorrência e $lmin$o comprimento minimo das linhas diagonais que formam o GR.

$$L = \frac{\sum_{l=lmin}^{N} lP\,(l)}{\sum_{l=lmin}^{N} lP\,(l)} \tag{3.10}$$

- **Comprimento máximo das linhas diagonais (Lmax)** - corresponde a media da linha diagonal mais longa do gráfico da recorrência, sendo $N_l$ o número total de linhas diagonais geradas para o GR.

$$L_{max} = max\left(\{l_i; i = 1, ..., N_l\}\right) \tag{3.11}$$

- **Laminaridade (LAM)** - corresponde a media da porcentagem de pontos recorrentes compreendendo estruturas de linhas verticais do gráfico da recorrência, onde $v$ é referente ao tamanho da estrutura vertical do GR, $P\,(v)$ corresponde a probabilidade das estruturas verticais geradas dentro de um GR e $vmim$ é referente ao número mínimo de estruturas verticais contabilizadas.

$$LAM = \frac{\sum_{v=vmin}^{N} vP\,(v)}{\sum_{i,j}^{N} R_{ij}}) \tag{3.12}$$

- **Comprimento médio das estruturas verticais (TT)** - corresponde a media doo comprimento médio das estruturas de linhas verticais do gráfico da recorrência.

$$TT = \frac{\sum_{v=vmin}^{N} vP\,(v)}{\sum_{v=vmin}^{N} P\,(l)} \tag{3.13}$$

Ao final do processo de extração das medidas dinâmicas (MQRs) coletadas de cada série de dados analisada, as MQRs coletadas são organizadas e armazenadas temporariamente em tempo de execução e serão utilizadas pelo módulo de detecção do método AIDA.

Com as MQRs computadas por esse módulo, mesmo diante de variabilidade dos valores dos atributos das séries temporais, é possível manter a constância dos seus resultados diante de séries com tráfego sem anomalias. Esse fator inibe o surgimento de falso positivos diante da presença de *outliers* nos valores dos atributos utilizados (RIGHI; NUNES, 2016).

### 3.3.1 Técnica para determinar taxa de Raio da Vizinhança

A definição do valor para o Raio da Vizinhança é uma etapa de suma importância, uma vez que esse parâmetro esta diretamente ligado ao processo de criação dos Gráficos da Recorrência. Como apresentado na Seção 2.3.2, dado um espaço de fase reconstruído, o raio da vizinhança define, com base na distância euclidiana, quais pontos são tidos como vizinhos recorrentes. A literatura apresenta diversas formas para determinar o valor para o Raio da Vizinhança, seguindo a premissa sempre que um maior ou menor valor de Raio da vizinhança refletirá diretamente no número de pontos recorrentes do Gráfico gerado.

No AIDA, a definição da taxa do raio da vizinhança utiliza a técnica desenvolvida por (WEBBER; ZBILUT, 2005). Nessa técnica o autor define três diretrizes, que devem ser seguidas e contempladas em ordem, para encontrar o valor ideal para o raio da vizinhança de uma determinada série de dados em análise.

A primeira diretriz consiste na análise de um gráfico de escala linear, formado pelos valores de taxa de recorrência coletados e dispostos no eixo vertical e os valores do raio da vizinhança em um intervalo variante de 1 a 100%, dispostos na horizontal, aonde deve-se observar o ponto do gráfico que apresente um aumento visual no valor da taxa da recorrência.

A segunda diretriz parte da análise de um gráfico de escala logarítmica, formado pelos valores de taxa de recorrência coletados e dispostos no eixo vertical e os valores do raio da vizinhança em um intervalo variante de 1 a 100% dispostos na horizontal, aonde deve-se observar que o valor do raio da vizinhança deve estar entre 0,1 até 2,0% da taxa da recorrência.

Por fim a terceira diretriz parte da análise de um gráfico formado pelos valores de determinismo coletados e dispostos no eixo vertical e os valores do raio da vizinhança em um intervalo variante de 1 a 100% dispostos na horizontal, aonde deve-se observar o primeiro ponto de inflexão e verificar se o valor do raio da recorrência nesse ponto coincide com o valor encontrado nas duas primeiras diretrizes. Caso haja coincidência, esse valor é adotado como ideal para o raio da recorrência da série analisada,

Em síntese, o valor ideal do raio da vizinhança é aquele que atende as três diretrizes em ordem. No entanto, existe a possibilidade de algumas séries não cumprirem a terceira diretriz. Nesse caso somente as duas primeiras diretrizes são utilizadas para definir o valor do raio da vizinhança.

### 3.3.2 Técnica para determinação do tempo de atraso (Delay - $\tau$)

O parâmetro de *delay* define o intervalo temporal entre as análises do valor de uma variável ao longo da reconstrução do espaço de fase. Ao definir o intervalo deve-se levar em conta que: um valor pequeno de *delay* resulta em uma elevada correlação entre os valores da série temporal analisada e um espaço de fases altamente alinhado com a direção da bissetriz do plano, e por consequência a alta recorrência dos dados; um tempo de *delay* com valores muito alto resulta em uma fraca correlação entre os valores da série temporal analisada e pois o espaço de fases tende a estar afastado da direção da bissetriz do plano.

No AIDA, a determinação do tempo de atraso utiliza a técnica de Informação Mútua Média (IMM) proposta por (FRASER; SWINNEY, 1986). Essa técnica utiliza a Entropia de Shannon para calcular a quantidade de informação que uma variável aleatória (de uma série temporal) contém sobre outra variável (da mesma série temporal). O valor ideal para o *delay* é aquele que possibilita extrair o máximo de informações em uma série temporal. Esse valor é obtido observando-se no gráfico o primeiro ponto de inflexão, ponto tido como ideal para reconstrução do espaço de fases de uma série temporal.

### 3.3.3 Dimensão de Imersão

A dimensão de imersão *dim* determina, dentro de um espaço de fase de um sistema, quantas dimensões do espaço são necessárias para representar corretamente a dinamicidade de um sistema. Esse parâmetro influencia diretamente na eficácia do método AIDA, pois influencia na representação do sistema pelas MQRs.

No AIDA a técnica utilizada para definir a dimensão de imersão é a técnica dos Falsos Vizinhos (B.; BROWN, 1992). A escolha dessa técnica se baseou em dois fatos: 1 - possui uma maior documentação e praticidade para sua implementação; 2 - foi empregada por (RIGHI; NUNES, 2016) no cenário de tráfego de rede e se mostrou eficiente.

Na técnica dos falsos vizinhos, o cálculo da dimensão de imersão é realizado incrementando o parâmetro *dim* de um em um até que a distância entre os pontos no espaço de fase da dimensão sob teste e da dimensão inicial pare de mudar. Em outras palavras, é um processo repetitivo que implica em: 1 - determinação da dimensão; 2 - reconstrução do espaço de fase; 3 - cálculo de distância entre pontos do espaço de fase da dimensão inicial e da dimensão sob teste; e 4 - decisão se continua busca retornando a fase 1 ou pára, caso a distância estabilize. Na

técnica, com a variação do valor da dimensão de imersão é possível a identificação dos falsos vizinhos, uma vez que os pontos que tendem a se manter próximos são validos para a reconstrução confiável do espaço de fase e os pontos que se distanciam são encarados como falsos vizinhos. Segundo (B.; BROWN, 1992), o valor propicio para a dimensão de imersão é aquele que apresenta o menor número de falsos vizinhos.

A identificação ajustada do valor da dimensão de imersão resulta na diminuição da ocorrência de falsos positivos durante o processo de classificação do tráfego IoT. O processo para determinar os valores adequados *dim* para cada atributo de rede está detalhado na Seção 4.4.

### 3.3.4 Módulo Detecção

O último módulo do método AIDA é o Módulo Detecção. Esse módulo tem o objetivo de detectar possíveis anomalias no tráfego da rede IoT, através da análise das características dinâmicas coletadas e armazenadas no módulo AQR. Esse módulo é ilustrado na Figura 6 e seu funcionamento é detalhado no Algoritmo 2.

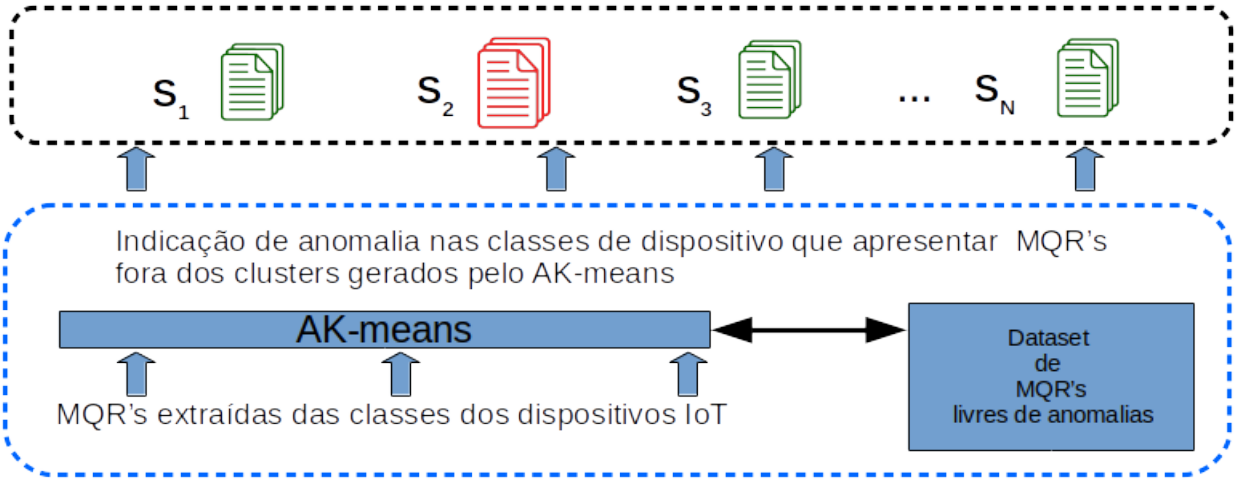


Figura 6 – Arquitetura do Módulo Deteção do método AIDA

Fonte: o próprio autor

De acordo com a Figura 6, o Módulo Detecção tem como entrada o conjunto de MQRs, extraídas de cada uma das classes de dispositivos da rede IoT pelo Módulo AQR. As MQRs são submetidas para o algoritmo de classificação adaptativa *Adaptive K-Means* (BHATIA, 2004), neste trabalho rotulado *A-Kmeans*. Diferentemente do Módulo Coleta e Segmentação, aonde é empregado o algoritmo K-means para classificar um dispositivo a uma classes de dispositivos presente na rede IoT, o Módulo Detecção considera a dinamicidade das séries de dados das

classes analisadas e emprega um algoritmo adaptativo como ferramenta de clusterização. O algoritmo A-Kmeans define automaticamente o número de clusters para cada série analisada e traz o benefício de poder operar com a diversidade de comportamento dentro de uma classe de dispositivos, possibilitando obter melhor qualidade de serviço no detector.

Para a execução do A-Kmeans é preciso um conjunto de dados com comportamento normal, o qual é utilizado como base de treinamento do algoritmo (*dataset* de treinamento). No AIDA, é assumido que os dados utilizados para formar essa base de treinamento são disponíveis. Neste trabalho, os dados (MQRs de treinamento) foram extraídos a partir da mesma base de dados utilizada para determinar as classes de dispositivos consideradas para determinação das classes usadas pelo Módulo Detecção e Segmentação. Na Figura 6 isto é ilustrado pelo uso do bloco que indica o Dataset de MQRs livres de anomalias.

Para a construção do *dataset* o classificador A-Kmeans inicialmente recebe e opera um conjunto de 33 características dinâmicas (MQRs), correspondente aos valores da Razão de Recorrência (RR), Determinismo (DTE) e Entropia (ENTR) para cada um dos 11 atributos do tráfego (vide Tabela 2). Esse mesmo processo é realizado para cada uma das classes de dispositivos da rede IoT. O número de clusters (k), que é definido automaticamente pelo algoritmo e corresponde aos seus respectivos centroids, são organizados e armazenados por classes de dispositivos.

Após a etapa de treinamento, as séries de dados vindas do Módulo AQR são processadas pelo algoritmo A-Kmeans para fins de identificação de anomalias. Ao final da execução, os números dos clusters gerados, bem como os valores dos seus centroids, são comparados com os valores registrados no na fase de treinamento. Se a maior parte dos clusters apresentar desvio significativo (anomalia), a série analisada que está em análise é indicada como anômala e inicia-se o processo para a identificação do dispositivo causador da possível anomalia.

A combinação do método AQR e do o algoritmo A-kmeans é outra contribuição do método AIDA no contexto de redes IoT. O AIDA busca através desta combinação melhor eficiência na detecção de anomalias em sistemas com comportamento dinâmicos e sem estacionaridade.

As características dinâmicas analisadas nesse processo do AIDA inviabilizam a identificação dos elementos responsáveis por essas anomalias. Dessa forma, faz-se necessário a ação de um segundo módulo de análise, o qual deve ter como objetivo a identificação dos elementos com comportamentos suspeitos por classe. Isto pode ser realizado se, diante da indicação de anomalias detectadas pelo Módulo Decisão, os conjuntos de dados de tráfego organizados pelo

**Algoritmo 2:** Algoritmo para detecção de anomalias no Módulo Detecção

**Entrada:** Série Temporal de MQRs por classe de dispositivo
**Saída:** Indicação de anomalia ou normalidade no fluxo de tráfego de uma classe de dispositivo IoT

```
1  início
2      para cada classe de dispositivo IoT faça
3          Leia dataset de treinamento de MQR's
4          Leia conjunto de MQR's obtidas no Módulo AQR
5          centroids_MQRs ← A-Kmeans(MQRs, thresholds_por_MQR)
6          Compara centroids_MQRs e centroids indicados no dataset de MQR's
7          if variação significativa then
8              Indicação de anomalia no tráfego da classe
9      fim
10 fim
11 retorna Indicação comportamental por classe de dispositivos
```

Módulo Coleta e Segmentação forem resubmetidos para o Módulo Detecção com o objetivo de identificar qual dispositivo apresenta anomalias. Porém, esta etapa está fora do escopo do método AIDA, que visa explorar a segmentação dos dispositivos por classe e a aplicação da AQR associada à clusterização adaptativa na identificação de anomalias em redes IoT.

## 3.4 CONSIDERAÇÕES PARCIAIS

Ao longo deste capítulo foi descrito o método AIDA e todos os seus módulos. A dinamicidade de redes IoT é um desafio para a detecção de anomalias nesses ambientes e o método AIDA apresenta uma abordagem inovadora para contornar tal problema. O método explorou a segmentação de uma rede IoT em classes comportamentais de dispositivos e, para cada classe, o emprego da análise da quantificação da recorrência. O capítulo salienta que análise de medidas de quantificação da recorrência é chave nesta abordagem, dado que são elas que efetivamente são capazes de representar o comportamento dinâmico do tráfego de um ambiente IoT.

Em relação aos métodos para detecção de anomalias para redes IoT presentes na literatura, salienta-se que as principais contribuições do método AIDA são: primeiro, considerar a heterogeneidade dos ambientes IoT, pois o método AIDA opera a nível de rede e com a modelagem e identificação de classes comportamentais de dispositivos, fornecendo assim um tratamento direcionado e adequado para cada tipo de comportamento presente na rede, independente das tecnologias ou aplicações presentes, ou que possam ser adicionadas no ambiente IoT; segundo, considerar a dinamicidade característica do tráfego de redes IoT, pois o método

AIDA explora o uso de medidas quantitativas da recorrência como ferramenta de análise do comportamento não estacionário do tráfego IoT.

# 4 CALIBRAGEM DO MÉTODO AIDA

O presente capítulo apresenta o processo para determinar e ajustar os parâmetros de funcionamento do método AIDA. O processo de calibragem consiste em uma série de testes com bases de dados controladas contendo tráfegos de redes IoT normais e com anomalias conhecidas, a fim de identificar parâmetros para segmentação em classes, bem como as MQRs para cada classe de dispositivo IoT.

O restante desse capítulo está disposto da seguinte forma: a Seção 4.1 descreve a composição e a organização da base de dados utilizada; a Seção 4.2 apresenta os resultados obtidos pelo método de segmentação de classes de dispositivos; a Seção 4.3 apresenta o método empregado para a escolha das séries temporais para cada classe de dispositivos IoT; na Seção 4.4 é apresentado o processo para determinar a Dimensão da Imersão utilizada no método AIDA; na Seção 4.5 é expresso a escolha do tempo de *delay* utilizado como intervalo para verificar o próximo estado de uma fase; na Seção 4.6 é apresentado o método e o valor elencado para determinar o Raio da Vizinhança; finalmente, a Seção 4.7 apresenta os valores obtidos para cada uma das 7 MQRs utilizadas no método proposto e seus limites utilizados na versão final calibrada do método AIDA.

## 4.1 BASES DE DADOS

Para o processo de experimentação do método AIDA foi utilizado uma base de dados fornecida pela *School of Electrical Engineering and Telecommunications da UNSW Sydney*[3] (A. SIVANATHAN H. HABIBI GHARAKHEILI; SIVARAMAN, 2018). Essa base foi criada com o intuito de fornecer meios e fomentos para pesquisa na área de análise de redes de IoT, e é amplamente utilizada e aceita na literatura (J. PINHEIRO ANTONIO; R., 2018)(SANTOS M., 2017)(SHAIKH; E. BOU-HARB J. CRICHIGNO, 2018)(BAI; L. YAO S. KANHERE S., 2018)(MEIDAN; M. BOHADANA A. SHABTAI1, 2017).

A base de dados fornecida pela UNSW consiste em dados brutos coletados em tráfego de rede IP de um ambiente IoT. O ambiente utilizado para gerar essa base de dados é composto por 28 dispositivos (vide Tabela 3) conectados a um roteador TPLink (*gateway*), via rede *wireless* ou cabo, conectado a Internet. Os dispositivos presentes nessa base de dados são câme-

[3] Disponível em: https://iotanalytics.unsw.edu.au/iottraces.html

ras de monitoramento, tomadas inteligentes, sensores de qualidade de ar, monitores de saúde, lâmpadas inteligentes, dentre outros, conforme detalha a Tabela 3.

Essa base de dados corresponde a um período de vinte dias de monitoramento da comunicação de uma rede IoT, resultando em um conjunto de dezenove arquivos de captura de tráfego (pcap). Cada arquivo contém aproximadamente vinte e quatro horas de monitoramento e variam de 234.491 à 4.948.806 pacotes capturados, resultando um total de 20.887.637 pacotes capturados em um período de 458 horas de monitoramento.

| Funcionalidade | Dispositivo | Comunicação | Quantidade |
|---|---|---|---|
| Hubs | Smart Things | Cabeada | 1 |
| | Amazon Echo | Wireless | 1 |
| Cameras | Netatmo Welcome | Wireless | 1 |
| | TP-Link Day Night Cloud camera | Wireless | 1 |
| | Samsung SmartCam | Wireless | 1 |
| | Dropcam | Wireless | 1 |
| | Insteon Camera | Cabeada/ Wireless | 2 |
| | Withings Smart Baby Monitor | Wireless | 1 |
| Tomadas e Monitores | Belkin Wemo switch | Wireless | 1 |
| | TP-Link Smart plug | Wireless | 1 |
| | iHome | Wireless | 1 |
| | Belkin wemo motion sensor | Wireless | 1 |
| Monitores de qualidade de Ar | NEST Protect smoke alarm | Wireless | 1 |
| | Netatmo weather station | Wireless | 1 |
| Monitores de saúde | Withings Smart scale | Wireless | 1 |
| | Blipcare Blood Pressure meter | Wireless | 1 |
| | Withings Aura smart sleep sensor | Wireless | 1 |
| Lampadas Inteligentes | LiFX Smart Bulb | Wireless | 1 |
| Outros | Triby Speaker | Wireless | 1 |
| | PIX-STAR Photo-frame | Wireless | 1 |
| | Smartphone Android | Wireless | 2 |
| | Smartphone Iphone | Wireless | 1 |
| | Laptop/MacBook | Wireless | 2 |
| | TPLink Router Bridge LAN (Gateway) | Cabeada/ Wireless | 1 |
| | Impressora HP | Wireless | 1 |

Tabela 3 – Dispositivos IoT presentes na base e dados UNSW

### 4.1.1 Geração de Base de Dados com Traços Anômalos

Da base de dados bruta da UNSW, foi realizado inicialmente um processo de remoção de dispositivos que não possuem uma taxa mínima de tráfego que permita a análise de recorrên-

cia. Os dispositivos removidos foram: Blipcare Blood Pressure meter, Smartphones Android, Laptos e MacBooks e os Smartphones IPhone. Os dispositivos removidos correspondem a um total de 2.772.917 pacotes. Após a remoção, os demais dispositivos utilizados resultam em uma base com 18.114.720 pacotes úteis para manipulação e análise.

Com o tráfego de interesse estratificado e ciente de que a base fornecida pela UNSW contém apenas tráfego livre de anomalias, para fins experimentais foi necessário a injeção de dados anômalos na base de dados. Os dados injetados visam simular ações maliciosas na comunicação dos dispositivos IoT. O processo de manipulação da base de dados está exemplificado na Figura 7. Para gerar o tráfego anômalo foi utilizado a biblioteca Python Scapy[4] da linguagem Python, a qual fornece uma série de funções para manipulação de arquivos pcap.

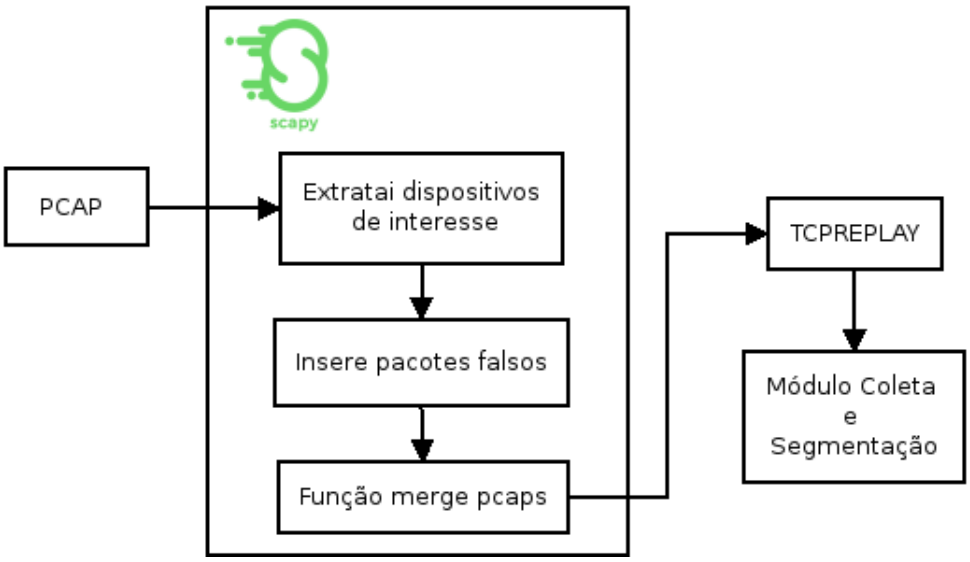


Figura 7 – Processo de manipulação da base de dados UNSW

Fonte: o próprio autor, via software VRA

No pré-processamento ilustrado na Figura 7, em um primeiro momento é extraído da base um grupo de dispositivos e seus pacotes. Em seguida são injetados ataques através da injeção de pacotes que representam ameaças e ações de agentes maliciosos conhecidos de ambientes IoT, tal como pacotes correspondentes ao ataque Mirai (maior ataque de DDoS já registrado) e pacotes que simulam um ataque de DDoS SYN Flood.

Para simular a ação do *malware* Mirai, foram criados fluxos de tráfego que representam a comunicação entre um mestre (dispositivo controlador) e um escravo (infectado). Os pacotes injetados nesse tráfego foram construídos manipulando características específicas

[4] Scapy é uma biblioteca escrita em Python que provê funções para manipulação de pacotes de rede: https://scapy.net/

de pacotes. A escolha das características manipuladas foi baseada nos trabalhos de (PROKOFIEV; Y. S. SMIRNOVA, 2018)(ANGRISHI, 2017)(FERRANDO; STACEY, 2017)(BERTINO; ISLAM, 2017)(DOSHI; N. APTHORPE, 2018)(SINANOVIC; MRDOVIC, 2017)(KOLIAS CONSTANTINOS KAMBOURAKIS, 2017). Os pacotes injetados correspondem a um fragmento de tráfego [5] capturado da comunicação de um dispositivo infectado com o *malware* Mirai. Como resultado, a injeção de um ataque Mirai corresponde a um conjunto de 20 pacotes de protocolos TELNET [6], com tamanhos que variam entre 66 e 78 bytes e com porta de comunicação de números 23 ou 54650.

Os trechos de tráfego que simulam um ataque de DDoS SYN Flood estão organizados em dois grupos. O primeiro grupo contém tráfego no qual os dispositivos IoT são autores do ataque de SYN Flood. O segundo grupo representa um fluxo de tráfego no qual alguns dispositivos da base são alvos e sofrem um ataque de SYN Flood. Os pacotes utilizados para gerar ambos tráfegos simulados com ataques SYN Flood possuem um tamanho de 100 bytes, utilizam protocolo TCP e Flag de controle de conexão SYN e porta de comunicação de destino de número 80 para os dispositivos atacantes e número 80 como origem para dispositivos alvo do ataque. O volume de pacotes é gerado randomicamente, variando de 3 a 10 pacotes por interação do dispositivo.

Após a inserção de pacotes no trace extraído, o trace de dados com ataques foi reinserido na base original através da utilização da função Merge packets da biblioteca Scapy. Essa função provê meios para a adição e ordenação de novos pacotes de rede em um arquivo pcap já existente. O resultado do processo de pré-processamento é uma nova base de dados contendo tráfego anômalo característico de ambientes de IoT, chamada TCPREPLAY na Figura 7. Os dados TCPREPLAY são usados como entrada do Módulo Coleta e Segmentação do método AIDA nos experimentos deste trabalho.

Os ataques que compõe a base de tráfego com anomalias estão dispostos de acordo com a classe comportamental dos dispositivos. Para a classe *Streaming*, tráfego original dos dispositivos TP-Link Day Night Cloud e Samsung SmartCam, foi inserido na forma intercalada traços de tráfego correspondente a ação do *malware* Mirai, totalizando 329.528 pacotes adicionais de tráfego malicioso. Na classe de dispositivos com comportamento de eventual, foram inseridos traços de ataque DDoS SYN Flood de forma intercalada ao tráfego original dos dispositivos TP-Link Smart plug e NEST Protect smoke alarm, totalizando 2.244.000 pacotes adicionais de

[5] Tráfego Mirai: https://github.com/ixiacom/ATI/blob/master/PCAPS/Mirai_command_and_control.pcap
[6] TELNET RFC: https://tools.ietf.org/html/rfc854

tráfego malicioso. Para a classe de comportamento eventual, o tráfego anômalo foi construído de forma que os dispositivos atuem como autores de ataque DDoS. Por fim, na classe de comportamento periódico foram inseridos traços de ataque DDoS SYN Flood de forma intercalada ao tráfego original dos dispositivos Smart Things e Amazon Echo, totalizando 1.688.359 pacotes adicionais de tráfego malicioso. Para essa classe, o tráfego anômalo foi construído de forma que os dispositivos atuem como vítimas de um ataque DDoS.

### 4.1.2 Divisão e Organização Base de Dados

Após gerar uma base de dados com traços anômalos (correspondentes à ataques), a base de dados foi divida em seis partes como expressa a Tabela 4.

A base de dados original é composta por vinte dias de tráfego de um ambiente IoT. Desse total, três dias com tráfego normal foram separados para treinamento do módulo de segmentação de classes, três dias com tráfego normal foram destinados para o teste do módulo de segmentação, três dias de tráfego normal foram utilizados para treinamento e coleta dos valores das MQRs e três dias de tráfego normal combinado com traços de anomalias foram usados para testar os valores das MQRs e sua classificação. Finalmente, outros seis dias foram usados para validação do método AIDA, três dias de tráfego normal e três dias com tráfego com anomalias. Desta forma, tanto os processos de treinamento e testes quanto o de validação, utilizam dados distintos.

| Quantidade de Dias | Tráfego | Função |
|---|---|---|
| 3 | Normal | Treinamento segmentação de classes |
| 3 | Normal | Teste de segmentação de classes |
| 3 | Normal | Treinamento MQR |
| 3 | Normal + Traços Anômalo | Teste MQR |
| 3 | Normal | Validação método |
| 3 | Normal + Traços Anômalos | Validação método |

Tabela 4 – Divisão da base e dados UNSW

## 4.2 SEGMENTAÇÃO DE CLASSES

O processo de segmentação adotado no método AIDA, segue o modelo descrito na Seção 3.2.1. Para o funcionamento desse processo é preciso a determinação de uma janela de tempo para coleta, extração e classificação das características do tráfego de rede que melhor descrevem

o comportamento de cada classe de dispositivos presente no ambiente IoT.

Os experimentos realizados nessa Seção visam determinar o tamanho apropriado da janela de tempo e os limiares de classificação de cada uma das características do tráfego de rede (vide Seção 3.2.2) para cada classe de dispositivos adotada nesse trabalho. Tais experimentos estão organizados em duas etapas: 1) extração e classificação das características do tráfego de rede de cada classe de dispositivo (Seção 4.2.1); e 2) determinação do tamanho das janelas de tempo para classificação de dispositivos (Seção 4.3).

### 4.2.1 Extração e Modelagem de Características para Classificação de Dispositivos

Em métodos supervisionados, para cada elemento a ser classificado é necessário utilizar um conjunto de dados previamente conhecidos para treinamento do algoritmo. Neste sentido, para a classificação de dispositivos IoT é preciso manter um conjunto de treinamento individual para cada um dos dispositivos, bem como para cada novo dispositivo ingressante. A utilização de um algoritmo de aprendizado não supervisionado, tal como o K-means, em conjunto com a abordagem de classes comportamentais, possibilita ao método de classificação contornar a questão do volume e diversidade de dispositivos em uma rede IoT, visto que sua classificação se baseia na proximidade de valores (medidas) semelhantes.

Mesmo sem a necessidade de um conjunto de treinamento especifico para cada dispositivo, o método AIDA precisa identificar e modelar os limiares para cada classe de dispositivos. Esse processo inicialmente é feito de forma empírica, aonde é preciso identificar manualmente um grupo de dispositivos da rede que representam cada uma das classes comportamentais.

Logo após a classificação são extraídas as características do tráfego (vide 4.2.1) de cada dispositivo. Para este processo de extração é utilizado um algoritmo escrito em Python que utiliza as bibliotecas Scapy e TShark [7], especificas para a manipulação de pacotes de rede. As características coletadas são organizadas em um *dataset* e processadas pelo algoritmo K-means. Os centroids resultantes para cada *clusters* (que representam cada classe comportamental) são utilizados como limiares para o processo de classificação e segmentação de dispositivos.

Para o processo de classificação, testes realizados previamente demonstram que características de endereços de origem e destino e de portas e protocolos podem ser características que trazem pontos de imprecisão e *overfiting* dos parâmetros de classificação do comportamento de

[7] TShark é uma interface CLI que permite a captura e analise de pacotes de rede: https://www.wireshark.org/docs/man-pages/tshark.html

um dado dispositivo. Já características estatísticas sobre o fluxo de rede apresentam um melhor resultado para o processo de classificação. Neste sentido, foi utilizado em todos os testes um conjunto formado pelas 5 características consideradas mais relevantes, haja vista que o uso de um número maior não acarreta em um aumento relevante nos valores de acurácia. Para a identificação dos dispositivos por características estatísticas do fluxo foram selecionados o tamanho médio dos pacotes utilizados na comunicação observada, a porta de origem e a porta de destino mais utilizadas e também a taxa de pacotes enviados e recebidos durante a amostra de tráfego observada. Para a classificação do tráfego de rede foram selecionados a taxa de pacotes enviados e recebidos, o total de bytes transmitidos e recebidos e o tamanho médio dos pacotes.

### 4.2.2 Determinação das janelas de tempo para classificação de dispositivos

A escolha do tamanho da janela de tempo para análise do tráfego do ambiente IoT é um parâmetro importante. Se a janela de tempo for muito grande, podem ocorrer distorções, pois as características tendem a se tornar similares entre os grupos de dispositivos. Já se a janela de tempo for muito pequena, as informações coletadas no intervalo observado podem não ser suficiente para caracterizar o real comportamento de um dispositivo ou classe de dispositivos.

Para a determinação do tamanho da janela de tempo com melhor acurácia para classificação de dispositivos, foram realizados experimentos com janelas de diferentes tamanhos de espaço amostral (60, 120, 360, 720, 1080 e 1440 minutos).

Para cada uma das seis janelas de tempo testadas, foi extraído e classificado os valores das quatro características do tráfego de rede e em seguida mensurada sua acurácia. Esse processo foi repetido para as classes de dispositivos (*Streaming*, Eventual e Periódico).

O conjunto de dados utilizado para essa etapa, corresponde a um total de três dias de tráfego livre de anomalias (vide Tabela 4), aonde foi identificado os dispositivos pertencentes a cada classe comportamental e suas respectivas funções, conforme representado na Tabela 5. A classe de comportamento *Streaming* contém nove dispositivos; a classe Eventual possuí oito dispositivos; e a classe Periódico possui cinco dispositivos. Desse total, uma parcela dos dispositivos de cada classe foi separada e utilizada como conjunto de treinamento, sendo quatro dispositivos para classe *Streaming*, quatro dispositivos para classe Eventual e 3 dispositivos para a classe Periódico.

Os resultados de acurácia obtidos em diferentes janelas de tempo para as classes comportamentais *Streaming*, Eventual e Periódico estão representados na Figura 8. O gráfico repre-

| Classe | Dispositivo |
|---|---|
| *Streaming* | TP-Link Day Night Cloud camera<br>Samsung SmartCam<br>Triby Speaker |
| Eventual | Insteon Camera<br>Belkin_Wemo_switch<br>TP-Link Smart plug<br>iHome<br>Belkin wemo motion sensor<br>NEST Protect smoke alarm<br>Netatmo weather station<br>Withings Smart scale<br>Withings Aura smart sleep sensor<br>Light Bulbs LiFX Smart Bulb<br>PIX-STAR Photo-frame |
| Periódico | Smart Things<br>Amazon Echo<br>Netatmo Welcome<br>Dropcam<br>Withings_Smart_Baby_Monitor<br>Belkin_Wemo_switch<br>Belkin wemo motion sensor<br>Light Bulbs LiFX Smart Bulb<br>Triby Speaker |

Tabela 5 – Organização das classes de dispositivos na base de dados da UNSW

senta o percentual de acurácia para cada uma das classes de dispositivos em relação a janela de tempo que vária de 60 a 1440 minutos. A linha com marcação estrela representa a classe de comportamento periódico, a linha com marcação circular representa a classe Eventual e por fim a linha com marcação quadrada representa a classe *Streaming*.

Ao longo do processo de experimentação identificou-se que para as classes de dispositivos adotadas nesse trabalho, a janela de tempo de maior acurácia é de 720 minutos, sendo atingido para a classe eventual uma acurácia de 89.6%, para a classe Periódico um percentual de 99,73% de acurácia, e para a classe *Streaming* uma acurácia de 80,73%.

## 4.3 DETERMINAÇÃO DAS JANELAS DE TEMPO PARA MQR

De maneira similar ao processo de classificação dos dispositivos, a determinação da janela de tempo, a ser utilizada para analisar as séries temporais que devem resultar nas MQRs que descrevem o comportamento dinâmico dos dispositivos, é um parâmetro de suma importância

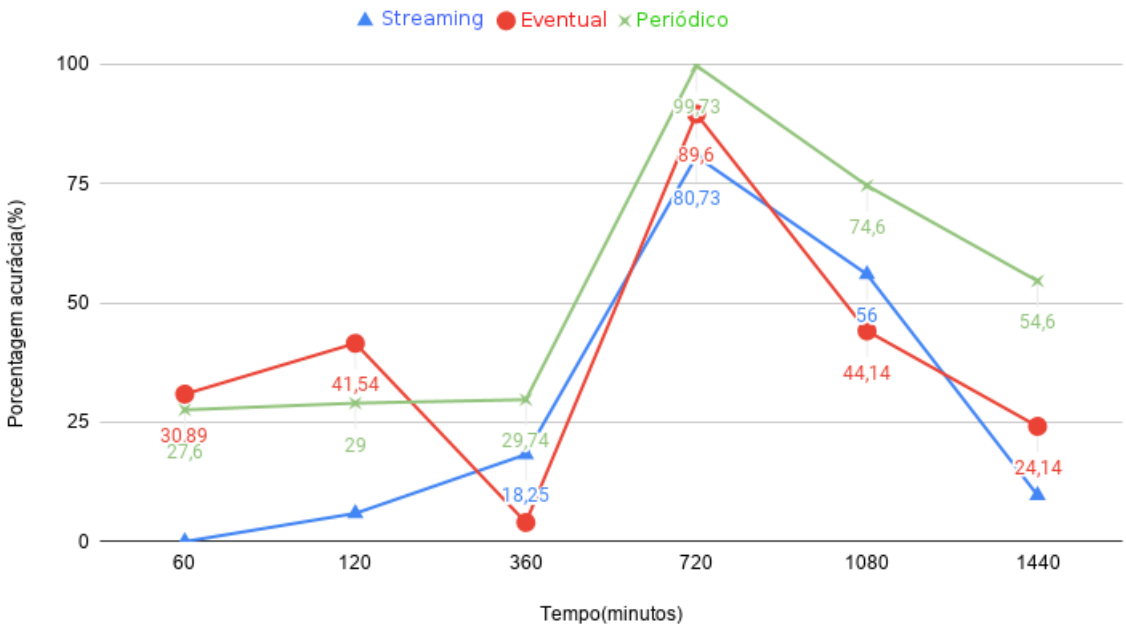


Figura 8 – Acurácia para classificação de dispositivos em classes de segmentação.

Fonte: o próprio autor

para o funcionamento do Módulo AQR do método AIDA. O tamanho adequado de uma janela de tempo, deve corresponder a um intervalo de observação de uma série temporal que contenha um número suficiente de amostras das caraterísticas do tráfego de rede, capaz de descrever a recorrência comportamental da classe de cada grupo de dispositivos IoT.

Para uma melhor seleção da janela de tempo, assume-se por hipótese que os grupos de dispositivos do ambiente IoT tenham um fluxo de comunicação periódica por segundo, dessa forma eliminando alguns dispositivos da base de dados (descrita na seção 4.1). Os experimentos realizados para seleção da janela de tempo utilizada no método AIDA para a extração das MQRs seguem como descrito. Para cada classe de comportamento (vide Seção 3.2.1.1) foram criadas séries temporais com durações de 5, 10, 30, 60, 90 e 120 minutos.

Cabe salientar que o tamanho de uma janela de tempo para AQR pode variar para diferentes ambientes IoT. Ambientes com maior fluxo de rede tendem a ter janelas de tempo menores, uma vez que seu comportamento recorrente pode ser capturado em um menor espaço temporal.

As janelas de tempo são formadas por dados extraídos de tráfego normal e com anomalias (vide Tabela 4) e submetidas ao algoritmo A-kmeans. O resultado obtido do processo de clusterização é coletado e aplicado o cálculo de acurácia. Esse processo é repetido para as três

classes comportamentais adotadas nesse trabalho. O resultado obtido em cada janela de tempo está ilustrado na Figura 9.

Os resultados expressos na Figura 9 estão organizados da seguinte forma: a linha com marcação estrela representa a classe de comportamento periódico; a linha com marcação circular representa a classe Eventual; e a linha com marcação quadrada representa a classe *Streaming*. Ao analisar a classe *Streaming* observa-se que a janela de maior acurácia é de 30 minutos, a qual atinge acurácia de 90,3%. Para a classe eventual a janela de tempo com melhor acurácia é a de 120 minutos, com uma acurácia de 91,05%. Para a classe Periódico a janela de maior acurácia é a de 60 minutos, com acurácia de 88,73%.

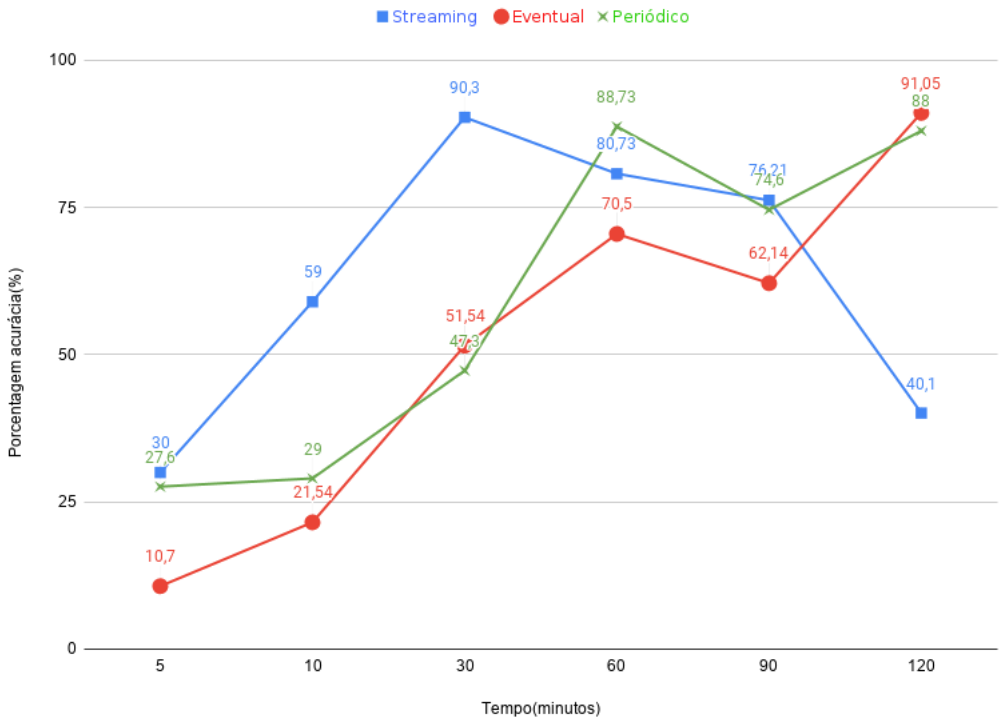


Figura 9 – Acurácia da classificação AIDA por classe comportamental.

Fonte: o próprio autor

## 4.4 DETERMINAÇÃO DAS DIMENSÕES DE IMERSÃO (DIM)

A dimensão de imersão estabelece o número de coordenadas para a reconstrução de um espaço de fase de um sistema. A partir do espaço de fase é possível representar a dinâmica de funcionamento de um dado sistema analisado (vide Seção 2.3.1.1). Dessa forma a dimensão imersão é um parâmetro de grande importância para o funcionamento e precisão do método

AIDA e a definição adequada desse parâmetro minimiza a ocorrência de falsos alertas de um comportamento recorrente. Nessa Seção são apresentados os experimentos para a definição da dimensão de imersão ($dim$) para os 5 atributos de rede adotados para o processo de AQR, para cada uma das três classes comportamentais.

Para identificar o valor da dimensão de imersão ideal que gera o menor número de falsos vizinhos, foi gerado um gráfico para cada um dos 5 atributos de redes utilizado nas três classes comportamentais, utilizando o software *Visual Recurrence Analysis (VRA)*. Tais gráficos são ilustrados nas Figuras 10, 11 e 12. Os resultados obtidos para dimensão de imersão ideal estão organizados na Tabela 6.

A Figura 10 apresenta o gráfico de falsos vizinhos gerado para o atributo Packet_size da classe *Streaming*, onde os valores dispostos no eixo vertical correspondem ao valor da porcentagem de falsos vizinhos, e variam de pouco menos de 45% até 52% para esse atributo. O eixo horizontal corresponde ao valor da dimensão de imersão utilizada em um determinado ponto do espaço de fase. Para esse atributo os valores variam de 1 até 10. Seguindo a diretriz de que o ponto com menor percentual de falsos vizinhos é o valor ideal da dimensão de imersão, pode-se observar que para o atributo Packet_size da classe *Streaming* o valor ideal para dimensão de imersão é 2.

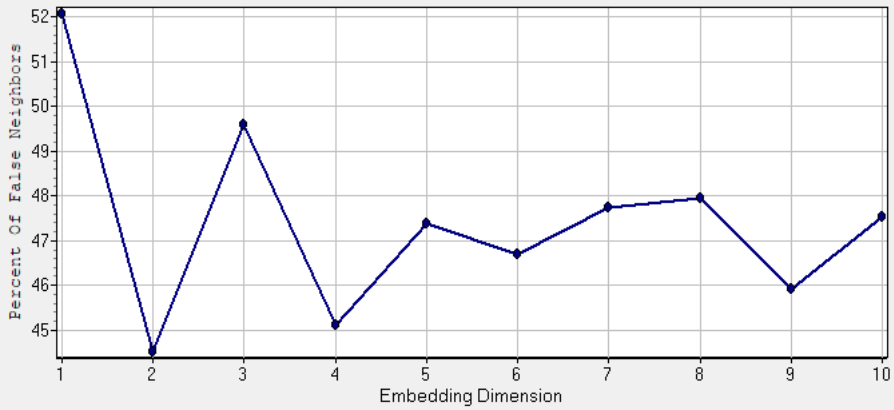


Figura 10 – Percentual de Falsos Vizinhos para o atributo Packet_size - Classe *Streaming*.

Fonte: o próprio autor, via software VRA

A Figura 11 apresenta o gráfico de falsos vizinhos gerado para o atributo Packet_size da classe Periódico, onde os valores dispostos no eixo vertical correspondem ao valor da porcentagem de falsos vizinhos, e variam de 47% até 51% para esse atributo. O eixo horizontal correspondente ao valor da dimensão de imersão os valores variam de 1 a 10 para esse atributo. Seguindo a diretriz para identificar o valor ideal da dimensão de imersão, para o atributo

Packet_size da classe Periódico pode-se observar que o valor ideal para dimensão de imersão é 7.

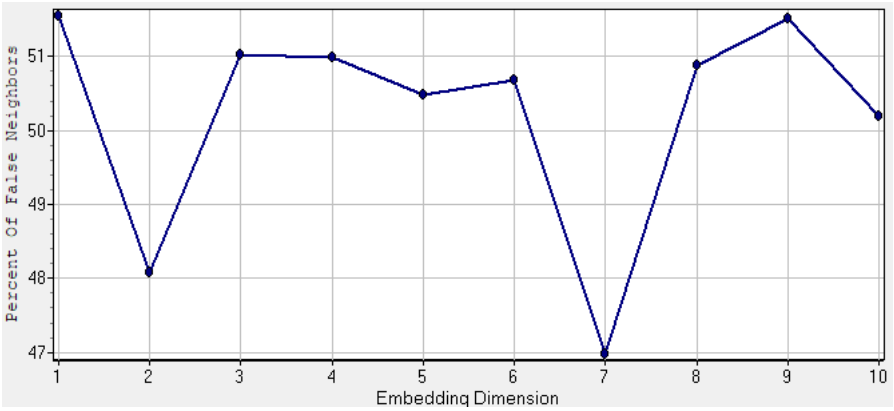


Figura 11 – Percentual de Falsos Vizinhos para o atributo Packet_size - Classe Periódico.

Fonte: o próprio autor, via software VRA

A Figura 12 apresenta o gráfico de falsos vizinhos gerado para o atributo Packet_size da classe Eventual, onde os valores dispostos no eixo vertical correspondem ao valor da porcentagem de falsos vizinhos, e variam de 47% até 51% para esse atributo. O eixo horizontal correspondente ao valor da dimensão de imersão os valores variam de 1 a 10 para esse atributo. Seguindo a diretriz de que o ponto com menor percentual de falsos vizinhos é o valor ideal da dimensão de imersão, para o atributo Packet_size pode-se observar que o valor ideal para dimensão de imersão é 7, ao fim do processo de analise, observou-se que o gráfico de falsos zinhos do atributo Packet_size apresenta um mesmo comportamento para as classes periódico e eventual. Ao investigar o motivo da semelhança dos resultados para o gráfico dos falsos vizinhos, observou-se que os dispositivos classificados para as classes de periódico e de *streaming* tem pacotes com valores que variam de 42 a 1481 bytes, o que difere as duas classes são os vetores de comunicação, sendo uma classe com os dispositivos comunicando após algum estimulo do ambiente e a outra classe mantendo uma comunicação regular em um tempo definido. O fato da igualdade dos gráficos ocorreu em virtude de que o calculo efetuado para o percentual de falsos vizinhos, baseia-se na distancia entre os valores próximos em uma série, dado uma dimensão de imersão adotada, uma vez que os dispositivos possuem semelhança nos tamanhos de pacote e volume gerado em suas comunicações, a distância e contabilidade dos falsos vizinhos, ao passo que incrementa a dimensão de imersão, tendem a gerar gráficos iguais par ambas as classes.

A mesma metodologia foi aplicada aos outros atributos para determinar o valor da dimensão de imersão $dim$. A Tabela 6 ilustra o resultado para todos os atributos e classes. A

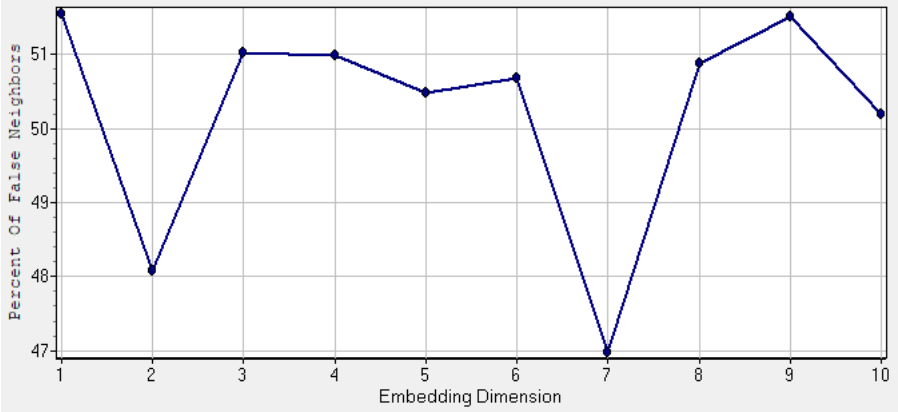


Figura 12 – Percentual de Falsos Vizinhos para o atributo Packet_size - Classe Eventual.

Fonte: o próprio autor, via software VRA

definição direcionada e ajustada do valor de dimensão de imersão impacta diretamente no cálculo das MQRs e por sua vez no desempenho do método AIDA, pois uma dimensão de imersão ajustada é aquela que possui o menor número de falsos vizinhos e por consequência terá uma baixa ocorrência de falsos alarmes. Os valores encontrados na Tabela 6 demonstram a existência de diferentes valores para a dimensão de imersão de um mesmo atributo em diferentes classes como, por exemplo, o atributo Packet_size, que possui 2 possíveis valores para 3 diferentes classes.

| Atributo | *Streaming* (m) | Periódico (m) | Eventual (m) |
|---|---|---|---|
| Packet_size | 2 | 7 | 7 |
| Tx_pac_send | 2 | 2 | 2 |
| Tx_pac_rec | 2 | 2 | 2 |
| Source_bytes | 2 | 2 | 2 |
| Destination_bytes | 5 | 2 | 2 |

Tabela 6 – Organização para valores da dimensão de imersão das classes de dispositivos.

## 4.5 DETERMINAÇÃO DO TEMPO DE ATRASO (DELAY)

O parâmetro de tempo de atraso ($delay$ - $\tau$) define o intervalo temporal entre as análises do valor de uma variável ao longo da reconstrução do espaço de fase. Para determinar o tempo de atraso utilizado para os 5 atributos de rede em cada uma das três classes comportamentais, foi utilizado a técnica de Informação Mútua Média (IMM) proposta por (FRASER; SWINNEY, 1986).

A Figura 13 apresenta o gráfico de Informação Mútua Média (IMM) gerado para o

atributo Packet_size da classe *Streaming*. Os valores dispostos no eixo vertical correspondem ao valor da média de informação mutua em um dado ponto, e variam de 0 até 2,6 para esse atributo. O eixo horizontal correspondente ao valor de $delay$ e os valores variam de 0 a 100 para esse atributo. Seguindo a diretriz de que o primeiro ponto de inflexão no gráfico é o valor ideal $delay$, para o atributo Packet_size pode-se observar que o valor ideal é 4.

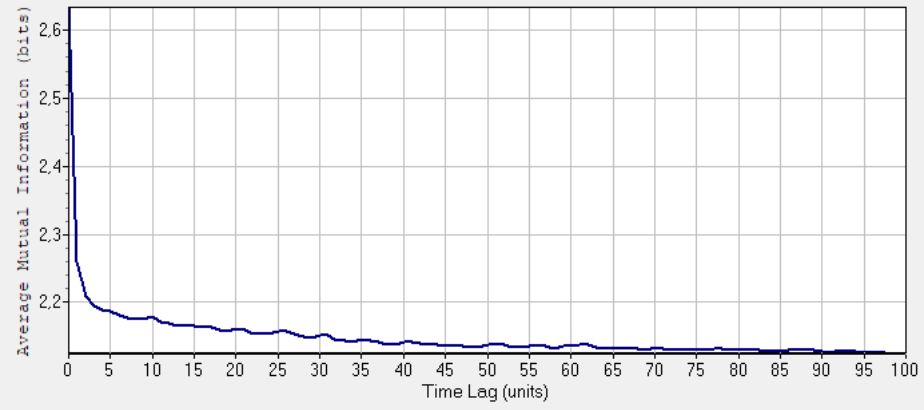


Figura 13 – IMM para o atributo Packet_size - Classe *Streaming*.

Fonte: o próprio autor, via software VRA

De maneira similar, as Figuras 14 e 15 apresentam os gráficos de IMM gerados para o atributo Packet_size da classe Periódico e Eventual, respectivamente. Os gráficos indicam um valor ideal de $delay$ igual 4 para o atributo Packet_size tanto na classe Periódico quanto na classe Eventual.

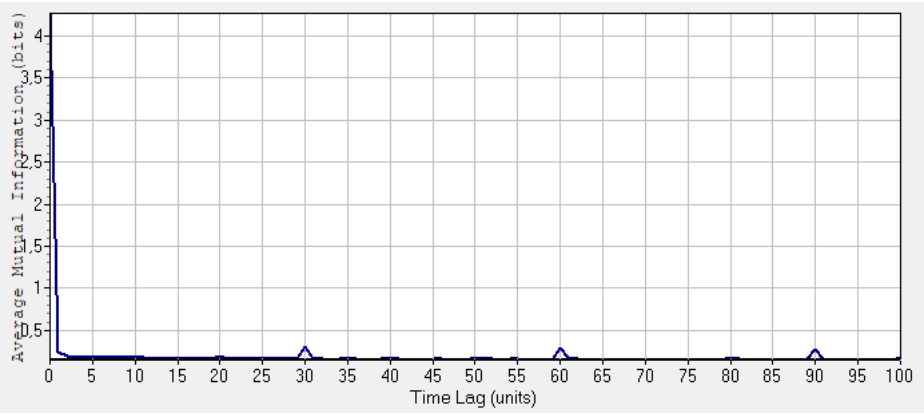


Figura 14 – IMM para o atributo Packet_size - Classe Periódico.

Fonte: o próprio autor, via software VRA

Os resultados de $delay$ ideal obtidos para os 5 atributos de redes a serem utilizados nas três classes comportamentais do método AIDA estão organizados na Tabela 7. Esses valores

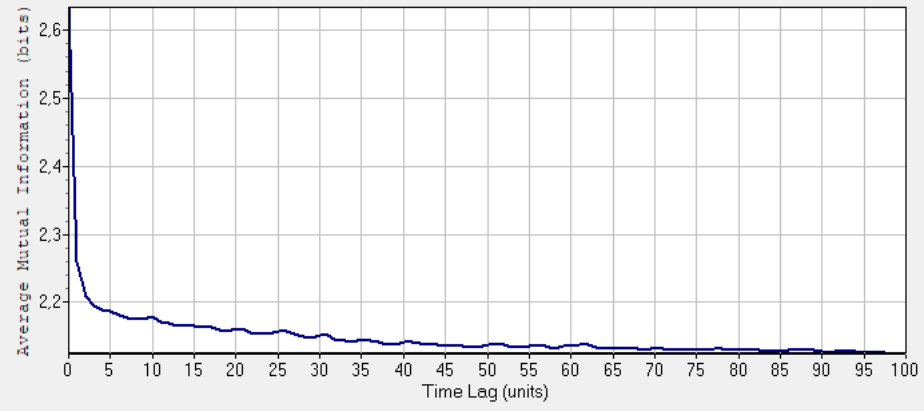


Figura 15 – IMM para o atributo Packet_size - Classe Eventual.

Fonte: o próprio autor, via software VRA

são obtidos utilizando a ferramenta VRA. Os valores de $delay$ são de suma importância para a criação dos GRs, uma vez que apontam a região do sistema dinâmico que possui maior informação para ser extraída. Na Tabela 7 observa-se que os valores de $delay$ encontrados diferem entre os atributos de uma mesma classe. Por exemplo, na classe Periódico há 3 possíveis valores para 5 atributos de rede e estes valores diferem entre as diferentes classes para um mesmo atributo. Observe o atributo Packet_size, que possui 2 possíveis valores para 3 diferentes classes. Deste modo, neste trabalho são utilizados valores específicos de $delay$ para cada atributo e sua respectiva classe, tal como expresso na Tabela 7.

| Atributo | *Streaming* ( $\tau$ ) | Periódico ($\tau$) | Eventual ($\tau$) |
|---|---|---|---|
| Packet_size | 8 | 4 | 4 |
| Tx_pac_send | 8 | 2 | 2 |
| Tx_pac_rec | 8 | 2 | 2 |
| Source_bytes | 5 | 5 | 5 |
| Destination_bytes | 5 | 5 | 5 |

Tabela 7 – Organização para valores de delay das classes de dispositivos

## 4.6 DETERMINAÇÃO DO RAIO DA VIZINHANÇA

O raio da vizinhança ( $\epsilon$ ) é um parâmetro importante para gerar o gráfico da recorrência (GRs) dos quais são extraídas as medidas da quantificação da recorrência(MQRs). Esse parâmetro irá definir a dispersão dos pontos de recorrência que compõem um GR. GRs com pontos muito ou pouco dispersos impactam nos valores MQRs. Seguindo o que é indicado na literatura (YUAN; R. YUAN, 2014), a taxa de raio da vizinhança adotada nesse trabalho é de 10%. Esse

valor é indicado como próximo ao ideal para a análise de anomalias de rede.

## 4.7 DETERMINAÇÃO DOS LIMITES DAS MQRS

Como demonstrado nas seções anteriores, as MQRs são descritores do comportamento dinâmico de um dado sistema (comportamento dinâmico do dispositivo). Porém, mesmo sendo capaz de descriminar a dinamicidade de um sistema, é preciso determinar os limites das MQRs para uma real compreensão do comportamento a partir das MQRs. Nessa Seção são definidos os valores para os limites superiores e inferiores para as 7 MQRs em cada uma das três classes comportamentais utilizadas no método AIDA.

A definição dos limites superior e inferior de uma MQR tem como objetivo delimitar os valores de um tráfego tido como normal e os valores excedentes a esses limites são tidos como indicação de um tráfego anômalo. Para elucidar as alterações nos valores limites que uma MQR pode sofrer, foi gerado um gráfico para cada classe comportamental, contendo uma série com tráfego normal e uma com tráfego anômalo. Os dados utilizados foram os mesmos da base de treinamento (vide Tabela 4) e os resultados para o atributo Packet_size são ilustrados nas Figuras 16 18 17. Ainda, para cada MQR coletada da série de tráfego sem anomalia, foi realizado o teste de Controle Estatísticos de Processo (CEP-Média Móvel) e os resultados obtidos por classe estão organizados nas Tabelas 8, 10 e 9. O teste CEP-Média Móvel é descrito em (CALZADA; SCARIANO, 2003) e já foi empregado em (RIGHI; NUNES, 2016) para a definição de limites de MQRs.

Ao analisar a Figuras 16 pode-se observar que para a classe *Streaming* uma série normal (linha com pontos círculos) mantém o valor da taxa da recorrência entre 59,17% e 92,78% e uma série anômala (linha com pontos em quadrados) apresenta uma média menor para a taxa da recorrência, aonde os valores se mantém na faixa de 47%. O resultado é uma diferença média de 40% em relação a série normal e anômala. Os resultados totais dos limites obtidos para a classe *Streaming* estão organizados na Tabela 8.

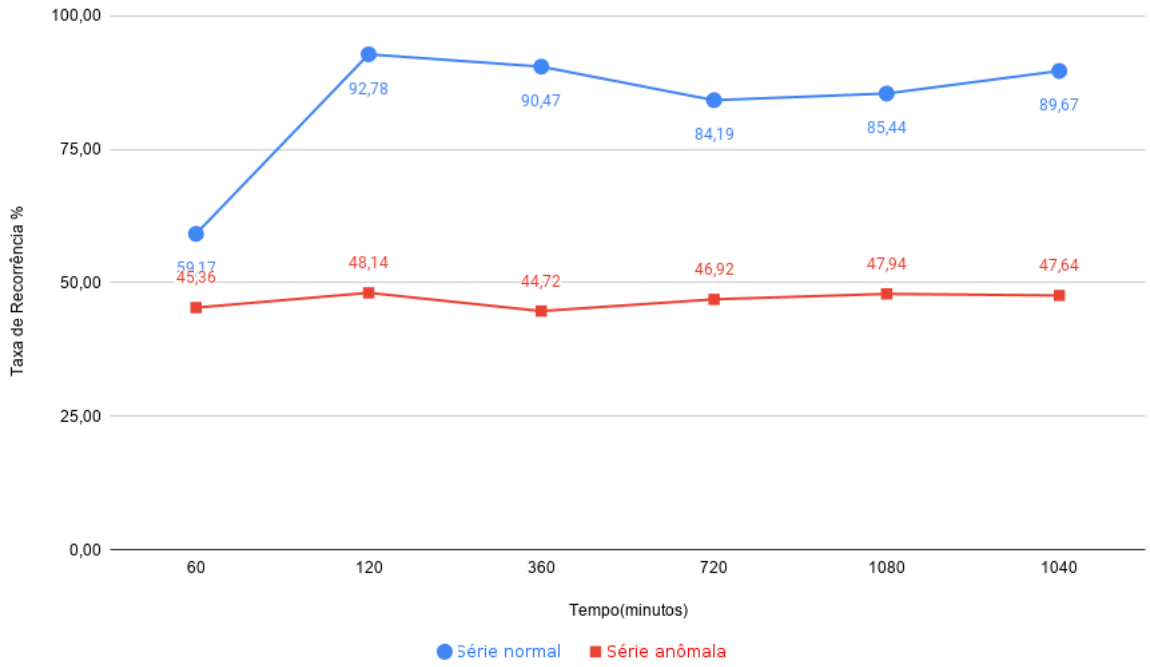


Figura 16 – Taxa de recorrência para o atributo Packet_size - Classe *Streaming*.

Fonte: o próprio autor

| Atributo | Item | RR | DET | Lmax | L | Entr | Lam | TT |
|---|---|---|---|---|---|---|---|---|
| Packet_size | Limite Superior | 102.02 | 112.83 | 34.53 | 10.95 | 2.96 | 116.24 | 23.62 |
| | Limite Inferior | 102.02 | 112.83 | 34.53 | 10.95 | 2.96 | 116.24 | 23.62 |
| | Linha Central | 51.60 | 73.23 | 12.22 | 3.63 | 1.43 | 79.12 | 5.25 |
| | Desvio Padrão | 16.81 | 13.20 | 7.44 | 2.44 | 0.51 | 12.37 | 6.12 |
| Source_bytes | Limite Superior | 4.09 | 132.89 | 1.35 | 0.97 | 2.90 | 87.67 | 13.67 |
| | Limite Inferior | -3.92 | -65.83 | -1.30 | -0.93 | -2.07 | -66.18 | -11.12 |
| | Linha Central | 0.09 | 33.53 | 0.02 | 0.02 | 0.41 | 10.74 | 1.28 |
| | Desvio Padrão | 1.33 | 33.12 | 0.44 | 0.32 | 0.83 | 25.64 | 4.13 |
| Destination_bytes | Limite Superior | 4.47 | 139.96 | 1.52 | 0.73 | 2.77 | 100.54 | 4.95 |
| | Limite Inferior | -4.26 | -39.62 | -1.45 | -0.69 | -1.27 | -75.44 | -3.69 |
| | Linha Central | 0.10 | 50.17 | 0.03 | 0.02 | 0.75 | 12.55 | 0.63 |
| | Desvio Padrão | 1.45 | 29.93 | 0.49 | 0.24 | 0.67 | 29.33 | 1.44 |
| Tx_pac_rec | Limite Superior | 118.84 | 131.91 | 41.89 | 14.65 | 3.20 | 132.67 | 21.32 |
| | Limite Inferior | -3.33 | 30.96 | -14.63 | -5.45 | -0.35 | 33.21 | -10.77 |
| | Linha Central | 57.75 | 81.44 | 13.63 | 4.60 | 1.43 | 82.94 | 5.27 |
| | Desvio Padrão | 20.36 | 16.83 | 9.42 | 3.35 | 0.59 | 16.58 | 5.35 |
| Tx_pac_send | Limite Superior | -30.45 | -26.55 | -15.64 | -3.46 | -0.88 | -8.32 | -13.70 |
| | Limite Inferior | 97.54 | 133.85 | 32.28 | 9.10 | 2.44 | 1.05 | 22.80 |
| | Linha Central | 33.54 | 53.65 | 8.32 | 2.82 | 0.78 | 70.46 | 4.55 |
| | Desvio Padrão | 21.33 | 26.73 | 7.99 | 2.09 | 0.55 | 26.26 | 6.08 |

Tabela 8 – Valores limites das MQRs - Classe *Streaming*.

A Tabela 8 demonstra que para a classe *Streaming*, cada atributo de rede apresenta diferentes limites de valores que suas MQRs podem assumir para representar um tráfego normal. Para esse trabalho os valores apresentados na Tabela 8 são utilizados como limiares de classificação do tráfego da classe *Streaming* no método AIDA.

Para a classe Periódico, os resultados relativos à taxa de recorrência para o atributo Packet_size são ilustrados na Figuras 17. Pode-se observar que a série normal (linha com pontos círculos) mantém o valor da taxa da recorrência entre 59,17% e 92,78% e os valores são semelhantes aos valores coletados para a classe de dispositivos de *Streaming*. A mesma semelhança acontece com os valores da série anômala (linha com pontos em quadrados) que apresentam uma taxa 40% menor em relação a série normal. Coincidentemente os valores da taxa de recorrência para as classes Periódico e *Streaming* são iguais, porém as demais MQRs obtidas diferem (vide Tabela 9). Os resultados totais dos limites obtidos para a classe Peroódico estão organizados na Tabela 9

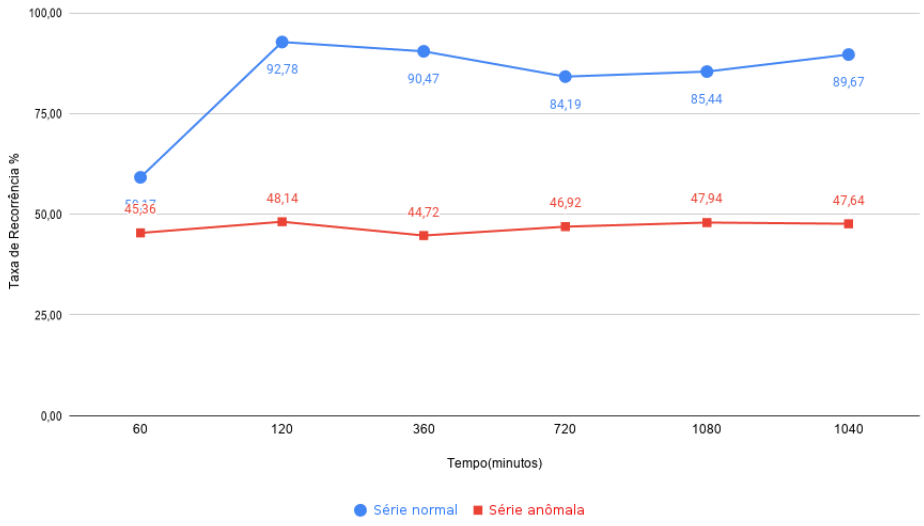


Figura 17 – Taxa de recorrência para o atributo Packet_size - Classe Periódico.

Fonte: o próprio autor, via software VRA

| Atributo | Item | RR | DET | Lmax | L | Entr | Lam | TT |
|---|---|---|---|---|---|---|---|---|
| Packet_size | Limite Superior | 102.02 | 112.83 | 34.53 | 10.95 | 2.96 | 116.24 | 23.62 |
| | Limite Inferior | 1.19 | 33.63 | -10.09 | -3.68 | -0.11 | 42.00 | -13.12 |
| | Linha Central | 51.60 | 73.23 | 12.22 | 3.63 | 1.43 | 79.12 | 5.25 |
| | Desvio Padrão | 16.81 | 13.20 | 7.44 | 2.44 | 0.51 | 12.37 | 6.12 |
| Source_bytes | Limite Superior | 4.09 | 132.89 | 1.35 | 0.97 | 2.90 | 87.67 | 13.67 |
| | Limite Inferior | -3.92 | -65.83 | -1.30 | -0.93 | -2.07 | -66.18 | -11.12 |
| | Linha Central | 0.09 | 33.53 | 0.02 | 0.02 | 0.41 | 10.74 | 1.28 |
| | Desvio Padrão | 1.33 | 33.12 | 0.44 | 0.32 | 0.83 | 25.64 | 4.13 |
| Destination_bytes | Limite Superior | 4.47 | 139.96 | 1.52 | 0.73 | 2.77 | 100.54 | 4.95 |
| | Limite Inferior | -4.26 | -39.62 | -1.45 | -0.69 | -1.27 | -75.44 | -3.69 |
| | Linha Central | 0.10 | 50.17 | 0.03 | 0.02 | 0.75 | 12.55 | 0.63 |
| | Desvio Padrão | 1.45 | 29.93 | 0.49 | 0.24 | 0.67 | 29.33 | 1.44 |
| Tx_pac_rec | Limite Superior | 118.84 | 131.91 | 41.89 | 14.65 | 3.20 | 132.67 | 21.32 |
| | Limite Inferior | -3.33 | 30.96 | -14.63 | -5.45 | -0.35 | 33.21 | -10.77 |
| | Linha Central | 57.75 | 81.44 | 13.63 | 4.60 | 1.43 | 82.94 | 5.27 |
| | Desvio Padrão | 20.36 | 16.83 | 9.42 | 3.35 | 0.59 | 16.58 | 5.35 |
| Tx_pac_send | Limite Superior | 97.54 | 133.85 | 32.28 | 9.10 | 2.44 | 149.25 | 22.80 |
| | Limite Inferior | -30.45 | -26.55 | -15.64 | -3.46 | -0.88 | -8.32 | -13.70 |
| | Linha Central | 33.54 | 53.65 | 8.32 | 2.82 | 0.78 | 70.46 | 4.55 |
| | Desvio Padrão | 21.33 | 26.73 | 7.99 | 2.09 | 0.55 | 26.26 | 6.68 |

Tabela 9 – Valores limites das MQRs - Classe Periódico.

A Tabela 9 demonstra os valores que suas MQRs podem assumir para representar um tráfego normal da classe Periódico. Os valores apresentados na Tabela 9 são utilizados como limiares de classificação do tráfego da classe Periódico no método AIDA.

Diferentemente, para o mesmo atributo, na classe Eventual (vide Figura 18) nota-se que a série normal (linha com pontos círculos) mantém o valor da taxa da recorrência entre 38,1% a 54,58%, ou seja, bem menores do que para as outras duas classes. Os valores da série anômala (linha com pontos em quadrados), com valores entre 45,3% a 47,6%, se mantiveram similares aos das outras classes, mas com uma média levemente inferior ao da série normal. Os resultados totais dos limites obtidos para a classe Eventual estão organizados na Tabela 10.

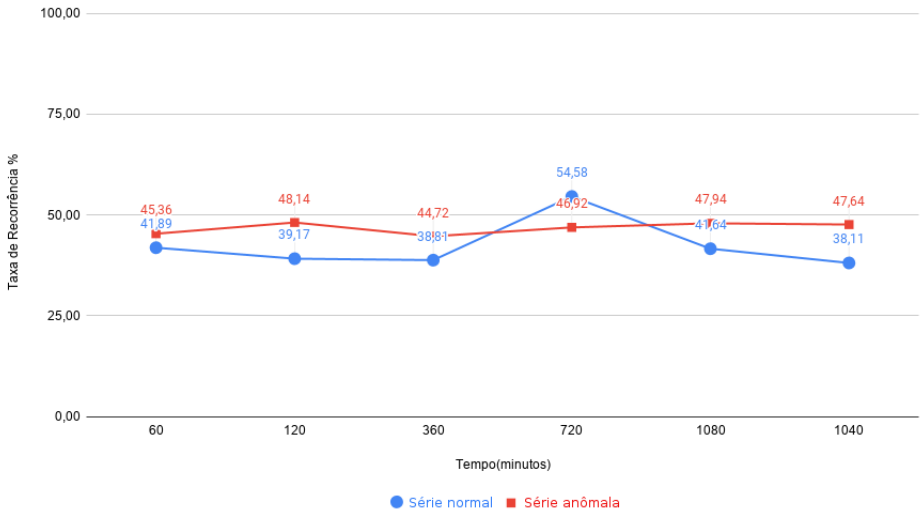


Figura 18 – Taxa de recorrência para o atributo Packet_size - Classe Eventual.

Fonte: o próprio autor, via software VRA

| Atributo | Item | RR | DET | Lmax | L | Entr | Lam | TT |
|---|---|---|---|---|---|---|---|---|
| Packet_size | Limite Superior | 103.64 | 112.75 | 36.76 | 11.98 | 3.05 | 117.21 | 27.33 |
| | Limite Inferior | -3.96 | 30.33 | -12.49 | -4.66 | -0.26 | 37.49 | -16.48 |
| | Linha Central | 49.84 | 71.54 | 12.14 | 3.66 | 1.39 | 77.35 | 5.42 |
| | Desvio Padrão | 17.93 | 13.74 | 8.21 | 2.77 | 0.55 | 13.29 | 7.30 |
| Source_bytes | Limite Superior | 0.44 | 30.84 | 0.24 | 0.24 | 0.00 | 23.08 | 1.31 |
| | Limite Inferior | -3.92 | -65.83 | -1.30 | -0.93 | -2.07 | -66.18 | -11.12 |
| | Linha Central | 0.09 | 33.53 | 0.02 | 0.02 | 0.41 | 10.74 | 1.28 |
| | Desvio Padrão | 0.14 | 8.32 | 0.08 | 0.08 | 0.00 | 0.00 | 7.20 |
| Destination_bytes | Limite Superior | 3.48 | 240.59 | 1.33 | 0.40 | 3.44 | 64.38 | 2.42 |
| | Limite Inferior | -3.35 | -148.81 | -1.29 | -0.39 | -2.13 | -55.83 | -2.10 |
| | Linha Central | 0.07 | 45.89 | 0.02 | 0.01 | 0.66 | 4.27 | 0.168 |
| | Desvio Padrão | 1.14 | 64.90 | 0.44 | 0.13 | 0.93 | 20.04 | 0.75 |
| Tx_pac_rec | Limite Superior | 128.40 | 139.31 | 45.42 | 15.28 | 3.35 | 138.69 | 21.95 |
| | Limite Inferior | -11.30 | 21.97 | -17.27 | -5.64 | -0.59 | 25.57 | -11.02 |
| | Linha Central | 58.55 | 80.64 | 14.08 | 4.82 | 1.38 | 82.13 | 5.46 |
| | Desvio Padrão | 23.28 | 19.56 | 10.45 | 3.49 | 0.66 | 18.85 | 5.50 |
| Tx_pac_send | Limite Superior | 106.78 | 140.36 | 36.62 | 8.84 | 2.51 | 154.59 | 22.61 |
| | Limite Inferior | -28.61 | -18.98 | -16.33 | -3.07 | -0.76 | -3.17 | -13.06 |
| | Linha Central | 33.54 | 53.65 | 8.32 | 2.82 | 0.78 | 70.46 | 4.55 |
| | Desvio Padrão | 21.33 | 26.73 | 7.99 | 2.09 | 0.55 | 26.26 | 6.08 |

Tabela 10 – Valores limites das MQRs - Classe Eventual.

A Tabela 10 apresenta os valores que suas MQRs podem assumir para representar um tráfego normal da classe Periódico. Os valores apresentados na Tabela 9 são utilizados como limiares de classificação do tráfego da classe Periódico no método AIDA.

## 4.8 CONSIDERAÇÕES FINAIS

Esse capitulo abordou o processo de calibragem do método AIDA, inicialmente é apresentado a base de dados utilizada como suporte para o treinamento dos parâmetros de classificação e segmentação de dispositivos, e dos parâmetros utilizados para realizar os cálculos das MQRS. Outros parâmetros definidos nesse capitulo são os valores referentes as janelas de tempo utilizadas no método AIDA, sendo a definição de uma janela única para o processo de segmentação de classes e outro conjunto de janelas de tempo utilizados para o calculo de AQR de cada uma das classes comportamentais.

Visto que o desenvolvimento de um método inovador para a classificação dos dispositivos IoT não é o objetivo principal do trabalho, além dos parâmetros definidos nesse capitulo, na seção 4.2.2 foram realizados experimentos para a classificação dos dispositivos IoT em suas respectivas classes comportamentais, o resultado obtido e apresentado na Tabela 5, foi adotado para os demais experimentos realizados para a validação do método AIDA.

## 4.9 CONSIDERAÇÕES PARCIAIS

A adoção da AQR pelo método AIDA requer a calibração do método. Embora a AQR tenha a capacidade de capturar comportamentos dinâmicos através das MQRs, a criação do gráfico da recorrência considera limiares que devem estar adequadamente calibrados aos tipo de dados que se deseja aplicar a AQR. Este capítulo dedicou-se a esta calibragem, tendo sido definido, com base de dados de redes IoT: as classes a serem adotadas na segmentação de dispositivos; o tamanho da janela temporal de amostragem a ser considerada no método AIDA; a dimensão da imersão, o *delay* e o raio da Vizinhança, utilizados no computo do gráfico da recorrência; e finalmente, os limiares para cada MQR utilizada no método IADA.

# 5 EXPERIMENTAÇÃO E VALIDAÇÃO DO MÉTODO AIDA

Este capítulo aborda os experimentos executados para avaliar o funcionamento e desempenho do método AIDA. O método é empregado em diferentes cenários de um ambiente IoT. O restante desse capítulo está organizado conforme segue: a Seção 5.1 apresenta detalhes do ambiente de experimentação e a Seção 5.2 apresenta as métricas adotadas para avaliação e aferição do método AIDA na experimentação; a Seção 5.3 demostra o processo de criação e inclusão de tráfego anômalo (simulação de tráfego malicioso) na base de dados; nas Seções 5.4, 5.5 e 5.6 são descritos, respectivamente, os experimentos 1 (uso do método AIDA em tráfego agregado não segmentado), 2 (uso do método AIDA em tráfego agregado segmentado) e 3 (comparativo com outros métodos de detecção); a Seção 5.7 apresenta uma discussão e análise dos resultados obtidos em cada um dos experimentos realizados. A Seção 5.8 encerra o capítulo apresentando as considerações parciais em relação à experimentação e validação do método.

## 5.1 AMBIENTE DE EXPERIMENTAÇÃO

Para realizar os experimentos e validações do método AIDA foi utilizado um computador com as seguintes configurações: Processador Intel(R) Core(TM) i3-4010U 1.70GHz, cache L2 256K e L3 3072K, 7882 MB de memória RAM e sistema operacional Debian GNU/Linux 9 (stretch) 64 bits. Para a implementação do método AIDA foi utilizada: a linguagem Python em conjunto com as bibliotecas Scapy e TShark que fornecem recursos para a manipulação de tráfego de rede, a linguagem R em conjunto com os pacotes de clusterização adaptativa (akmeans), o pacote que fornece a integração entre R e a linguagem C++ (Rcpp), bem como o pacote para análise de quantificação da recorrência (rqa).

## 5.2 MÉTRICAS DE AVALIAÇÃO

Baseado no número de amostras de tráfego livre de anomalias (N) e no número de amostras de tráfego contendo anomalias previamente conhecidas (A), o método AIDA foi avaliado nesse trabalho através das seguintes métricas:

1. Verdadeiro Positivo (VP): indicativo da classificação correta de uma amostra de tráfego anômalo;

2. Verdadeiro Negativo (VN): indicativo da classificação correta de uma amostra de tráfego livre de anomalias;

3. Falso Positivo (FP): indicativo da classificação equivocada de uma amostra de tráfego normal, indicando que se trata de uma amostra de tráfego com anomalia;

4. Falso Negativo (FN): indicativo da classificação equivocada de uma amostra de tráfego com anomalias, indicando que se trata de uma amostra de tráfego com normal;

5. Acurácia (AC): corresponde a capacidade do método AIDA para detectar o comportamento anômalo de forma correta. Expressa na forma de percentual, essa métrica é calculada através da divisão feita entre o resultado da soma dos valores de verdadeiros positivos e verdadeiros negativos e o resultado da soma dos valores de amostras livres e com anomalias. O cálculo percentual para taxa de acurácia do método AIDA está representado na equação 5.1.

$$AC = \left( \frac{VP + VN}{N + A} \right) * 100 \tag{5.1}$$

6. Precisão (P): Métrica que mensura a porção de classificação correta de uma amostra de tráfego anômalo (VP) em relação as amostras classificados como positivos

$$\frac{VP}{(VP + FP)} \tag{5.2}$$

7. Sensitividade (Recall): mensura a porção de verdadeiros positivos classificados corretamente pelo método, o cálculo para essa métrica está representado na equação 5.3.

$$\frac{VP}{(VP + FN)} \tag{5.3}$$

8. F1-Score: representa um balanço entre as métricas de Precisão e Sensitividade, o cálculo para essa métrica está representado na equação 5.4..

$$2 * \left( \frac{P * Recall}{P + Recall} \right) \tag{5.4}$$

Os elementos utilizados para realizar os cálculos das métricas utilizadas são baseados em uma matriz de confusão, exemplificadas na Tabela 11.

| | Normal | Anomalia |
|---|---|---|
| Normal | VN | FP |
| Anomalia | FN | VP |

Tabela 11 – Matriz de Confusão.

## 5.3 CRIAÇÃO E INCLUSÃO DE TRÁFEGO ANÔMALO

Existe uma escassez de bases de dados de ambientes IoT reais que fornecem dados a nível de rede. Tal escassez se estende a bases com tráfego de dispositivos infectados ou sobre o controle de agentes mal intencionados. Para contornar essa situação foi preciso criar traços de tráfego anômalo de forma sintética e posteriormente inserí-los na base de dados, com o objetivo de criar séries de tráfego anômalo para análise do método AIDA (vide Seção 4.1.1).

O tráfego sintético foi gerado utilizando um pequeno programa escrito em Python e que utiliza a biblioteca Scapy. Os pacotes de rede que compõe esse tráfego malicioso foram criados contendo características de ataques conhecidos, como Mirai Bot e ataques de DDoS SYN Flood. Após criados, as amostras de tráfego malicioso foram inseridas num traço de tráfego do ambiente IoT com 3.600 minutos (60 horas). Cada amostra analisada pelo AIDA corresponde a 1 minuto de observação de tráfego. Cada ponto de inserção de tráfego sintético foi realizado em um tempo conhecido e intercalado com traços de tráfego livre de anomalias. A Tabela 12 ilustra o resultado deste processo de criação e inserção de tráfego anômalo nos dados de experimentação. O valores da coluna Classe correspondem a: T para Transmissão, P para Periódico e E para Eventual.

| Tempo (h) | Amostras | Tráfego | Ataque | Classe | Ambiente IoT |
|---|---|---|---|---|---|
| 1-2 | 120 | Normal | - | T, P, E | Tráfego agregado |
| 2-4 | 120 | Normal | - | T, P, E | |
| 4-6 | 120 | Anomalia | Mirai | T | |
| 6-8 | 120 | Normal | - | T, P, E | |
| 8-10 | 120 | Anomalia | DDoS | P | |
| 10-12 | 120 | Normal | - | T, P, E | |
| 12-14 | 120 | Anomalia | DDoS | E | |
| 14-16 | 120 | Normal | - | T, P, E | |
| 16-18 | 120 | Anomalia | Mirai | T | |
| 18-20 | 120 | Normal | - | T, P, E | |
| 20-22 | 120 | Anomalia | DDoS | P | |
| 22-24 | 120 | Normal | - | T, P, E | |
| 24-26 | 120 | Anomalia | Mirai | T | |
| 26-28 | 120 | Normal | - | T, P, E | |
| 28-30 | 120 | Anomalia | DDoS | E | |
| 30-32 | 120 | Normal | - | T, P, E | |
| 32-34 | 120 | Anomalia | DDoS | P | |
| 34-36 | 120 | Normal | - | T, P, E | |
| 38-40 | 120 | Anomalia | DDoS | P | |
| 40-42 | 120 | Normal | - | T, P, E | |
| 42-44 | 120 | Anomalia | Mirai | T | |
| 44-46 | 120 | Normal | - | T, P, E | |
| 46-48 | 120 | Anomalia | Mirai | T | |
| 48-50 | 120 | Normal | - | T, P, E | |
| 52-54 | 120 | Anomalia | DDoS | E | |
| 56-58 | 120 | Normal | - | T, P, E | |
| 58-60 | 120 | Anomalia | DDoS | E | |

Tabela 12 – Organização do tráfego IoT normal/anômalo para a experimentação.

## 5.4 EXPERIMENTO 1 - MÉTODO AIDA SOBRE TRÁFEGO AGREGADO

O objetivo do Experimento 1 é verificar a capacidade de detecção de anomalias utilizando a análise quantitativa da recorrência via método AIDA sobre um tráfego agregado de internet das coisas.

Nesse Experimento foi utilizado uma série de 1440 minutos de tráfego agregado (sem segmentação por classe) livre de anomalias. O processo para extração e definição dos valores dos parâmetros para o cálculo de MQRs e também seus limites superior e inferior foi realizado conforme indicado no Capítulo 4. Deste modo, o processo para obter os valores de dimensão de imersão e $delay$, seguiram tal como descrito nas seções 4.4 e 4.5, respectivamente. O valor de raio da vizinhança foi fixado em 10%. A Tabela 13 representa os resultados obtidos sobre

o tráfego agregado para os parâmetros a serem utilizados no cálculo das MQRs durante esse Experimento 1. Ao analisar os valores presentes na Tabela 13, observa-se claramente que existe uma diferença desses valores em relação aos calculados para cada uma das classes (vide Tabela 6 e 7). Essa diferença indica que a análise do comportamento dinâmico de um traço agregado difere em relação ao tráfego classificado de forma segmentada por classe de dispositivo. Este resultado aponta que a aplicação da AQR é sensível ao tipo de tráfego, sugerindo a aplicação de segmentação para obtenção de melhores resultados.

| Atributo | Dimensão Imersão ($dim$) | Delay ($\tau$ ) | Raio da Vizinhança ( $\epsilon$ ) |
|---|---|---|---|
| Packet_size | 2 | 7 | 10 |
| Tx_pac_send | 2 | 2 | 10 |
| Tx_pac_rec | 2 | 2 | 10 |
| Source_bytes | 2 | 2 | 10 |
| Destination_bytes | 5 | 2 | 10 |

Tabela 13 – Organização para valores da dimensão de imersão, delay e raio da vizinhança do tráfego IoT agregado.

Dando sequência a análise, o segundo conjunto de valores considerados importantes para uma análise de traços via a AQR são os valores de limites inferior e superior de cada uma MQR. Esses valores definem os limites de variação que os valores de uma MQR pode atingir e ser considerado normal. O processo para a definição desses valores no Experimento 1 foi realizado tal como o procedimento descrito na seção 4.7. Os resultados obtidos estão organizados na Tabela 14.

| Atributo | Item | RR | DET | Lmax | L | Entr | Lam | TT |
|---|---|---|---|---|---|---|---|---|
| Packet_size | Limite Superior | 58.28 | 533.94 | 16.18 | 3.41 | 0.72 | 99.36 | 4.62 |
| | Limite Inferior | 39.60 | 344.80 | 4.77 | 2.42 | 0.51 | 57.60 | 1.86 |
| | Linha Central | 48.94 | 439.37 | 10.48 | 2.92 | 0.61 | 78.48 | 3.24 |
| | Desvio Padrão | 3.11 | 31.52 | 1.90 | 0.16 | 0.04 | 6.96 | 0.46 |
| Source_bytes | Limite Superior | 80.81 | 318.09 | 8.36 | 5.13 | 1.35 | 201.48 | 58.10 |
| | Limite Inferior | -51.25 | -221.22 | -5.59 | -3.23 | -0.11 | -80.79 | -28.92 |
| | Linha Central | 14.78 | 48.44 | 1.38 | 0.95 | 0.62 | 60.34 | 14.59 |
| | Desvio Padrão | 22.01 | 89.88 | 2.32 | 1.39 | 0.24 | 47.04 | 14.50 |
| Destination_bytes | Limite Superior | 80.86 | 147.24 | 8.54 | 5.16 | 1.68 | 200.39 | 58.97 |
| | Limite Inferior | -51.37 | -28.46 | -5.70 | -3.23 | -0.74 | -79.27 | -29.51 |
| | Linha Central | 14.74 | 59.39 | 1.42 | 0.96 | 0.47 | 60.56 | 14.73 |
| | Desvio Padrão | 22.04 | 29.28 | 2.37 | 1.40 | 0.40 | 46.61 | 14.75 |
| Tx_pac_rec | Limite Superior | 111.67 | 795.42 | 38.58 | 17.20 | 3.03 | 110.77 | 21.80 |
| | Limite Inferior | 49.67 | 81.06 | -2.93 | -2.93 | 0.89 | 80.58 | -5.73 |
| | Linha Central | 80.67 | 438.24 | 17.82 | 7.13 | 1.96 | 95.68 | 8.04 |
| | Desvio Padrão | 10.33 | 119.06 | 6.92 | 3.35 | 0.36 | 5.03 | 4.59 |
| Tx_pac_send | Limite Superior | 109.75 | 719.32 | 53.40 | 15.09 | 1.02 | 104.74 | 31.83 |
| | Limite Inferior | 55.83 | 143.42 | 1.85 | -0.72 | 0.57 | 91.10 | -6.08 |
| | Linha Central | 82.79 | 431.37 | 27.62 | 7.18 | 0.80 | 97.92 | 12.87 |
| | Desvio Padrão | 8.99 | 95.98 | 8.59 | 2.63 | 0.08 | 2.27 | 6.32 |

Tabela 14 – Valores limites das MQRs - Tráfego Agregado

Ao comparar os valores dos limites de MQRs para o ambiente IoT agregado, expressos na Tabela 14, e os valores de limites de MQRs coletados de forma ajustada para cada classe comportamental, expressos nas Tabelas 8 , 9 e 10, pode-se observar que existe uma diferença relevante entre os valores. Por exemplo, para o ambiente agregado o limite de classificação superior da MQR RR referente ao atributo de Packet Size é 58.28, já para as demais classe esse valor é superior a 100 (*Streaming* e Periódico -102.02 e para a classe Eventual - 103.03). A diferença entre os valores, quando coletados do tráfego agregado e quando coletados por classe comportamental, reforça a ideia desse trabalho, do tratamento de tráfego segmentado, visto que o valor da taxa de recorrência de pacotes praticamente dobra quando calculada para cada classe.

Definidos os limiares para a classificação do tráfego, foi realizado no Experimento 1 o processo de análise de acurácia do AIDA para o tráfego agregado do ambiente IoT, ou seja, a análise da acurácia sobre dados sem a distinção entre dispositivos ou classes comportamentais. Os resultados obtidos estão sintetizados na Tabela 15. Os resultados de tráfego identificado que possuem marcação com asterisco (*) assinalam a detecção equivocada do tráfego.

| Tempo(h) | Quantidade de amostra | Tráfego Esperado | Tráfego Identificado |
|---|---|---|---|
| 1-2 | 120 | Normal | Normal |
| 2-4 | 120 | Normal | Normal |
| 4-6 | 120 | Anomalia | **Normal*** |
| 6-8 | 120 | Normal | Normal |
| 8-10 | 120 | Anomalia | **Normal*** |
| 10-12 | 120 | Normal | Normal |
| 12-14 | 120 | Anomalia | Anomalia |
| 14-16 | 120 | Normal | Normal |
| 16-18 | 120 | Anomalia | **Normal*** |
| 18-20 | 120 | Normal | **Anomalia*** |
| 20-22 | 120 | Anomalia | **Normal*** |
| 22-24 | 120 | Normal | Normal |
| 24-26 | 120 | Anomalia | Anomalia |
| 26-28 | 120 | Normal | Normal |
| 28-30 | 120 | Anomalia | Anomalia |
| 30-32 | 120 | Normal | **Anomalia*** |
| 32-34 | 120 | Anomalia | Anomalia |
| 34-36 | 120 | Normal | Normal |
| 38-40 | 120 | Anomalia | Anomalia |
| 40-42 | 120 | Normal | Normal |
| 42-44 | 120 | Anomalia | **Normal*** |
| 44-46 | 120 | Normal | Normal |
| 46-48 | 120 | Anomalia | Anomalia |
| 48-50 | 120 | Normal | **Anomalia*** |
| 52-54 | 120 | Anomalia | Anomalia |
| 56-58 | 120 | Normal | Normal |
| 58-60 | 120 | Anomalia | Anomalia |

Tabela 15 – Distribuição do tráfego agregado em traços de tráfego almejados e traços obtidos pós classificação

Ao observar os valores obtidos no processo de classificação do tráfego agregado (Tabela 15), observa-se que de um total de 26 traços de tráfego analisados (13 normais e 13 com anomalias) tem-se: 9 amostras de tráfego normal classificadas corretamente (VN), 8 amostras de tráfego anômalo classificado corretamente (VP), 3 amostras de tráfego normal classificados erroneamente como anômalo (FP) e 5 amostras de tráfego contendo anomalias sendo classificados como tráfego normal (FN). O resultado é uma acurácia de 68% para o tráfego agregado. Este resultado demonstra a dificuldade inerente para identificação de anomalias em dispositivos IoT se o tráfego estiver agregando tráfego de dispositivos de diferentes classes comportamentais. A diversidade de comportamentos no tráfego dificulta de sobre maneira a identificação do tráfego de um dispositivo ou classe de dispositivo.

## 5.5 EXPERIMENTO 2 - MÉTODO AIDA SOBRE TRÁFEGO SEGMENTADO

Nesse experimento foram realizados testes do método AIDA, aplicando a análise quantitativa da recorrência em conjunto com o algoritmo A-kmeans de forma direcionada para cada classe comportamental de dispositivos IoT. Tal como Experimento 1 (vide Seção 5.5), foram utilizados os dados planilhados na Tabela 12 como fonte de tráfego a ser analisada pelo método AIDA. Os parâmetros $dim$ e $delay$ empregados para a obtenção das MRQ foram obtidos conforme apresentado nas Seções 4.5 e 4.4, respectivamente, bem como os limites das MQR de cada um dos atributos de rede, por classe comportamental, foram obtidos conforme apresentado na Seção 4.7. A distribuição dos traços de tráfego analisados pelo método AIDA nesse Experimento está representa na Tabela 16. Os traços de tráfego estão organizados em duas colunas principais, uma coluna indicando o tráfego esperado (original) e outra indicando o tráfego identificado (resultante da classificação do método AIDA).

| Tempo(h) | Quantidade de amostra | Tráfego Esperado | Tráfego Identificado |
|---|---|---|---|
| 1-2 | 120 | Normal | Normal |
| 2-4 | 120 | Normal | Normal |
| 4-6 | 120 | Anomalia | Anomalia |
| 6-8 | 120 | Normal | Normal |
| 8-10 | 120 | Anomalia | Anomalia |
| 10-12 | 120 | Normal | Normal |
| 12-14 | 120 | Anomalia | Anomalia |
| 14-16 | 120 | Normal | Normal |
| 16-18 | 120 | Anomalia | **Normal*** |
| 18-20 | 120 | Normal | Normal |
| 20-22 | 120 | Anomalia | Anomalia |
| 22-24 | 120 | Normal | Normal |
| 24-26 | 120 | Anomalia | Anomalia |
| 26-28 | 120 | Normal | Normal |
| 28-30 | 120 | Anomalia | Anomalia |
| 30-32 | 120 | Normal | **Anomalia*** |
| 32-34 | 120 | Anomalia | Anomalia |
| 34-36 | 120 | Normal | Normal |
| 38-40 | 120 | Anomalia | Anomalia |
| 40-42 | 120 | Normal | Normal |
| 42-44 | 120 | Anomalia | **Normal*** |
| 44-46 | 120 | Normal | Normal |
| 46-48 | 120 | Anomalia | Anomalia |
| 48-50 | 120 | Normal | Normal |
| 52-54 | 120 | Anomalia | Anomalia |
| 56-58 | 120 | Normal | Normal |
| 58-60 | 120 | Anomalia | Anomalia |

Tabela 16 – Distribuição do tráfego em traços de tráfego almejados e traços obtidos pós classificação do ambiente IoT observado por classes comportamentais

Ao observar os valores obtidos após o processo de classificação do tráfego através do método AIDA, processando o tráfego de acordo com cada classe, nota-se que de um total de 26 traços de tráfego analisados (13 normais e 13 com anomalias) foram obtidos os seguintes resultados: 12 amostras de tráfego normal classificadas corretamente (VN), 10 amostras de tráfego anômalo classificado corretamente (VP), 2 amostra de tráfego normal classificada como anômala (FP) e 2 amostras de tráfego contendo anomalias sendo classificadas como tráfego normal (FN). O resultado mostra uma acurácia de 91,66% para o método AIDA. Este resultado é bem superior à acurácia de 68% obtida para o tráfego agregado, demonstrando que a segmentação de tráfego por classe comportamental resulta em grande benefício para a capacidade classificatória da composição AQR + A-kmeans.

A Tabela 17 apresenta uma síntese dos resultados obtidos nos dois experimentos realizados, aonde pode-se observar a acurácia do emprego da análise quantitativa da recorrência sobre ambientes IoT, principalmente quando aplicado de forma direcionada e ajustada para diferentes classes comportamentais. Ao relacionar os resultados dos dois experimentos, identifica-se uma diminuição na ocorrência de falsos positivos e falsos negativos e consequentemente um acréscimo no número de verdadeiros positivos e negativos, aonde o Experimento 2 obtêm uma diferença de 3,88% de falsos positivos e 11,54% de falsos negativos, resultando em um ganho de 23,66% na acurácia do AIDA. Nota-se que um tratamento de forma individual e direcionada para cada classe presente em um ambiente IoT apresenta um significativo ganho na taxa de detecção, pois uma vez que os parâmetros para os cálculos das MQRs são definidos de forma ajustada para cada classe comportamental, o número de falsos negativos e falsos positivos tendem a serem menor.

| Experimento | VP(%) | FP(%) | VN(%) | FN(%) | AC(%) |
|---|---|---|---|---|---|
| 1 | 30,70 | 11,54 | 34,62 | 19,23 | 68 |
| 2 | 46,15 | 7,69 | 38,46 | 7,69 | 91,66 |

Tabela 17 – Síntese dos resultados dos Experimento 1, Experimento 2

## 5.6 EXPERIMENTO 3 - COMPARATIVO DO MÉTODO AIDA E OUTROS MÉTODOS DE DETECÇÃO

Com o objetivo de comparar o desempenho do método AIDA, frente a outros métodos de detecção de anomalia em ambientes IoT, foram realizados experimentos com outros dois métodos de classificação. Os métodos executados para esse experimento são o método DDoSbyAQR e o método de classificação através de regressão logística (vide seção 2.4). As Seções 5.6.1 e 5.6.2 descrevem, respectivamente, os experimentos realizados com os métodos DDoSbyAQR e Regressão logística.

### 5.6.1 DDoSbyAQR

Para esse experimento foi implementado o método DDoSbyAQR apresentado em (RIGHI; NUNES, 2016). O método DDoSbyAQR foi executado em dois cenários. O primeiro cenário analisa o ambiente IoT via tráfego agregado e o segundo cenário analisa o ambiente IoT via tráfego segmentado de acordo com as classes comportamentais. Em ambos os cenários a execução do método DDoSbyAQR utilizou os atributos de rede, parâmetros de *dim* e *delay* e os limiares

de classificação indicados pelos autores. Os dados utilizados para experimentação seguem tal como organizados na Tabela 12.

Os resultados obtidos para o primeiro cenário (DDoSbyAQR processando o tráfego agregado) pode-se observar que de um total de 26 traços de tráfego analisados (13 normais e 13 com anomalias) foram obtidos os seguintes resultados: 0 amostras de tráfego normal classificadas corretamente (VN), 8 amostras de tráfego anômalo classificado corretamente (VP), 2 amostra de tráfego normal classificada como anômala (FP) e 16 amostras de tráfego contendo anomalias sendo classificadas como tráfego normal (FN), alcançando uma acurácia de 30.76%. Para o segundo cenário (DDoSbyAQR processando o tráfego segmentado) pode-se observar que de um total de 26 traços de tráfego analisados (13 normais e 13 com anomalias) foram obtidos os seguintes resultados: 1 amostra de tráfego normal classificadas corretamente (VN), 13 amostras de tráfego anômalo classificado corretamente (VP), 5 amostra de tráfego normal classificada como anômala (FP) e 1 amostra de tráfego contendo anomalias sendo classificadas como tráfego normal (FN), totalizando uma acurácia de 53.84%.

Os resultados totais obtidos nesse experimento estão organizados na Tabela 18. Cada linha da tabela descreve o percentual de verdadeiro positivo (VP), falso negativo (FP), verdadeiro negativo (VN), falso negativo (FN) e acurácia (AC), obtidos nas análises dos traços de trafego do ambiente IoT agregado e segmentado, quando analisados pelo método DDoSbyAQR. Dos resultados é possível observar que para o tráfego segmentado, em relação ao não segmentado, há um pequeno aumento no número de falsos positivos (11,54%) e um aumento significativo dos verdadeiros positivos, demonstrando que o tráfego segmentado também melhora a condição de detecção no método DDoSbyAQR. É possível observar também uma queda de 34,62% no número de falsos negativos. Finalmente, a acurácia aumenta de 30,76% para 53,84%, sedimentando o benefício da análise de tráfego agregado.

| DDoSbyAQR | VP(%) | FP(%) | VN(%) | FN(%) | AC(%) |
|---|---|---|---|---|---|
| Agregado | 30,77 | 7,69 | 0 | 61,54 | 30.76 |
| Segmentado | 50 | 19,23 | 3,85 | 26,92 | 53,84 |

Tabela 18 – Síntese dos resultados do experimento com o método DDoSbyAQR

Analisando comparativamente o AIDA com o DDoSbyAQR (vide Tabela 19), observa-se que o AIDA apresenta acurácia muito superior ao obtido pelo método DDoSbyAQR. O AIDA consegue alcançar 91,66% de acurácia, contra apenas 53,84% do DDoSbyAQR.

| | VP(%) | FP(%) | VN(%) | FN(%) | AC(%) |
|---|---|---|---|---|---|
| AIDA | 46,15 | 7,69 | 38,46 | 7,69 | 91,66 |
| DDoSbyAQR | 50 | 19,23 | 3,85 | 26,92 | 53,84 |

Tabela 19 – Síntese comparativa do AIDA com o DDoSbyAQR (com tráfego segmentado)

### 5.6.2 Regressão Logística

Para esse experimento foi implementado e executado o método de regressão logística, processando o tráfego agregado e segmentado de acordo com as classes comportamentais mencionadas nesse trabalho. O método de regressão logística, diferentemente dos métodos DDoSbyAQR e AIDA, é uma técnica estatística que se baseia em um determinado modelo probabilístico para determinar de forma categórica a probabilidade de um evento acontecer. O modelo probabilístico é criado a partir da observação de um grupo de valores (traços de entrada). Para criar o modelo de classificação, foi utilizado como conjunto de treinamento os dados descritos tal como na Seção 4.1. Para o processo de experimentação os dados utilizados seguem tal como organizados na Tabela 12. Diferentemente dos outros métodos experimentados, a técnica de regressão logística analisa cada amostra de trafego individualmente, ou seja, para esse experimento foram analisados um total de 3.120 amostras (o equivalente aos 26 traços de tráfego para análise apresentados na Tabela 12).

Os resultados obtidos para o primeiro cenário (regressão logística processando o tráfego agregado) permitem observar os seguintes resultados: 1.099 amostras de tráfego normal classificadas corretamente (VN), 1.048 amostras de tráfego anômalo classificado corretamente (VP), 420 amostras de tráfego normal classificadas como anômala (FP) e 561 amostras de tráfego contendo anomalias sendo classificadas como tráfego normal (FN), alcançando uma acurácia de 30,76%. No segundo cenário (regressão logística processando o tráfego segmentado) pôde-se observar os seguintes resultados: 748 amostras de tráfego normal classificadas corretamente (VN), 1.511 amostras de tráfego anômalo classificado corretamente (VP), 458 amostras de tráfego normal classificadas como anômala (FP) e 403 amostras de tráfego contendo anomalias sendo classificadas como tráfego normal (FN), alcançando uma acurácia de 68,50%.

Os resultados totais obtidos nesse experimento estão organizados na Tabela 20. A tabela descreve o percentual de VP, FP, VN, FN e AC, dos traços de tráfego do ambiente IoT agregado e segmentado, quando analisados pelo método regressão logística. Ao observar os valores obtidos é possível observar novamente observa-se um acrescimo da acurácia quando o tráfego está

segmentado em classes, o que corrobora com os resultados obtidos com os outros métodos.

| Regressão Logística | VP(%) | FP(%) | VN(%) | FN(%) | AC(%) |
|---|---|---|---|---|---|
| Agregado | 30,33 | 13,46 | 35,22 | 0,16 | 68,55 |
| Segmentado | 48,43 | 14,68 | 23,97 | 1,28 | 72,40 |

Tabela 20 – Síntese dos resultados dos experimento com o método Regressão Logística

Analisando o desempenho do AIDA com o método de Regressão Logística para o tráfego segmentado (vide Tabela 21), observa-se que o AIDA apresenta acurácia superior. O AIDA consegue alcançar 91,66% de acurácia, contra 72,40% do método de Regressão Logística.

| | VP(%) | FP(%) | VN(%) | FN(%) | AC(%) |
|---|---|---|---|---|---|
| AIDA | 46,15 | 7,69 | 38,46 | 7,69 | 91,66 |
| Reg Log | 48,43 | 14,68 | 23,97 | 1,28 | 72,40 |

Tabela 21 – Síntese comparativa do AIDA com o método de Regressão Logística (com tráfego segmentado)

## 5.7 ANÁLISE DOS RESULTADOS

Esta seção avalia os resultados dos experimentos considerando as métricas de Precisão, Sensitividade (Recall), Acurácia e o balanço entre Precisão e Sensitividade (F1-Score). Os resultados alcançados para as mesmas configurações dos experimentos anteriores estão sintetizados na Tabela 22.

Dos resultados observa-se que para os métodos experimentados a maior precisão é de 85,71%, alcançada pelo método AIDA atuando sobre tráfego segmentado. Sobre este mesmo tipo de tráfego, a diferença nos valores de precisão, em comparação com os demais métodos, é ganho de 13,49% em relação ao DDoSbyAQR e 8,98% em relação ao de Regressão Logística. Para o tráfego agregado, o método DDoSbyAQR alcançou a melhor precisão (80%), 7,28% melhor do que o AIDA. Para a métrica de sensitividade (Recall), a qual indica a capacidade de um método de classificar corretamente uma classe de interesse, observa-se que para tráfego segmentado o método AIDA é novamente o melhor. Ele obteve um Recall de 85.71%, frente aos 78,94% do de regressão logística e 53,84% do DDoSbyAQR. Ao analisar os resultados obtidos para a métrica de F1-Score observa-se que o método AIDA, quando aplicado num tráfego segmentado, alcança o melhor resultado (85,71%), e relação aos demais métodos experimentados. A métrica de F1-Score, que representa um balanço entre as métricas de Precisão e Sensitividade,

indica que o método AIDA apresenta uma boa capacidade de classificação correta, apontando que o método proposto tem capacidade de alcançar boa qualidade de detecção.

O bom resultado do método AIDA também pode ser observado em termos de acurácia, uma métrica que analisa o percentual de acertos em relação ao total de classificações e que já foi explorada nas análises por experimento. Comparativamente, tal como ilustrado na coluna Acurácia da Tabela 22, o AIDA também atingiu o maior percentual de acurácia (91,66%), em relação aos demais métodos experimentados. Para o mesmo tipo de tráfego (segmentado) o método Regressão Logística apresentou a segunda maior acurácia, atingindo 72,40%.

| Modelo | Precisão(%) | Recall(%) | Acurácia(%) | F1-Score(%) |
|---|---|---|---|---|
| Regressão Logística Agregado | 71,23 | 64,95 | 68,55 | 67,95 |
| Regressão Logística Segmentado | 76,73 | 78,94 | 72,40 | 77,82 |
| DDoSbyAQR Agregado | 80 | 33,33 | 30,76 | 47,05 |
| DDoSbyAQR Segmentado | 72,22 | 65 | 53,84 | 68,42 |
| AIDA Agregado | 72,72 | 61,53 | 68 | 66,66 |
| AIDA Segmentado | 85,71 | 85,71 | 91,66 | 85,71 |

Tabela 22 – Síntese dos resultados dos Experimento 1, Experimento 2 e Experimento 3

Os resultados ilustram que o método proposto nessa dissertação, que utiliza AQR + A-kmeans em conjunto com a segmentação de classes comportamentais, permite obter uma boa qualidade de serviço na detecção de anomalias em ambientes IoT.

## 5.8 CONSIDERAÇÕES PARCIAIS

Neste capítulo foram desenvolvidos experimentos com o objetivo de avaliar o desempenho do método proposto neste trabalho. Para tal foram realizados três experimentos. O primeiro experimento realizado, executa o método AIDA para o tratamento do tráfego IoT de forma agregada, aonde os resultados obtidos demonstram que uma baixa capacidade de detecção de anomalias, com uma taxa de 11,54% de falso negativos e 19,23% de falsos positivos resultando em uma acurácia de 68%. O segundo experimento executa o método AIDA para o tratamento do tráfego IoT aplicando o conceito de segmentação, oferecendo um tratamento direcionado para cada classe comportamental. O resultado desse experimento demonstra uma capacidade satisfatória para a classificação de anomalias, apresentando uma redução na taxa de falsos positivos (7,69%) e falsos negativos (7,69) e um aumento da acurácia para 91,66%.

O terceiro experimento realizado neste capítulo consiste na execução de outros dois métodos de classificação de tráfego e o objetivo desse experimento é avaliar e comparar os

resultados obtidos desses métodos em relação ao método AIDA. Foram executados os métodos DDoSbyAQR e Regressão Logística, ambos processando o tráfego IoT de forma agregada e segmentada. Dos resultado obtidos, percebeu-se que o método Regressão Logística Segmentado apresentou uma acurácia de 72,40%, sendo superior ao DDoSbyAQR, mas não ao AIDA, o qual apresenta uma acurácia 19,26% maior.

# 6 CONSIDERAÇÕES FINAIS

Ao longo dessa dissertação foi construído um método para detecção de anomalias em ambientes de Internet das Coisas (IoT). O funcionamento desse método baseia-se na combinação de duas abordagens, a identificação e segmentação de classes comportamentais de dispositivos IoT combinada com a análise da quantificação da recorrência e a clusterização adaptativa. O uso da técnica de análise da quantificação da recorrência em ambientes de redes é pouco explorado, tendo sido empregada inicialmente com sucesso para a detecção de ataques DDoS em redes convencionais.

Os experimentos realizados nesse trabalho demonstraram que a análise de tráfego agregado de redes IoT pode induzir classificações errôneas e pouco precisas, reforçando assim a necessidade de uso de uma abordagem de segmentação do tráfego nesses ambiente, tal como o método AIDA propõe. Porém o processo de segmentação exige um grau de conhecimento do ambiente IoT, a fim de possibilitar que o método de segmentação possa aferir se os atributos utilizados modelam corretamente as classes de dispositivos presentes.

No desenvolvimento desse trabalho fomos confrontados com algumas dificuldades. No contexto da experimentação para a validação do método proposto, a seleção adequada de atributos para a segmentação e classificação dos dispositivos IoT é um desafio. A escolha deve levar em conta atributos capazes de modelar corretamente o comportamento dos dispositivos presentes na rede. A seleção de atributos errados pode levar a classificações equivocadas. Por outro lado, modelagens com classes muito especializadas podem levar a *overfiting* e tornar o modelo de segmentação e classificação pouco eficiente para operar com a presença de novos dispositivos no ambiente IoT.

Outra dificuldade encontrada no desenvolvimento desse trabalho foi a escassez de bases de dados com traços de redes IoTs reais. Em sua maioria as bases públicas encontradas são bases geradas em laboratório e com pouca diversidade de dispositivos ou tecnologias. Atrelado a falta de bases reais, também existe uma escassez de bases que apresentam tráfego anômalo de dispositivos infectados. Dessa forma, nesse trabalho foi preciso o desenvolvimento e inserção de tráfegos anômalos sintéticos para formar uma base composta por anomalias.

Por fim, os experimentos realizados neste trabalho demonstram que a utilização da análise quantitativa da recorrência em conjunto com a clusterização adaptativa, quando direcionada ao tratamento de classes comportamentais de dispositivos IoT, tal como proposto no método

AIDA, é uma nova abordagem para a detecção de anomalia em ambientes IoT que apresenta qualidade de serviço competitiva.

## 6.1 TRABALHOS FUTUROS

A exploração do uso de segmentação e AQR para classificação comportamental de dispositivos IoT é um desafio que ainda merece de mais atenção. Um ponto a ser melhor explorado na utilização de AQR em ambientes IoT como um elemento chave para a detecção de anomalias está relacionado à maior experimentação e avaliação de quais MQRs possuem melhor desempenho para cada atributo de rede. Outro ponto relevante é realizar a seleção e experimentação de novos atributos de rede que permitam avaliar métodos eficazes para a segmentação de classes comportamentais de dispositivos IoT. Finalmente, outro ponto a ser explorado é a experimentação da abordagem proposta pelo método AIDA com outras bases de dados e com outros tipos de anomalias.

# REFERÊNCIAS

A. SIVANATHAN H. HABIBI GHARAKHEILI, F. L. A. R. C. W. A. V.; SIVARAMAN, V. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. **IEEE Transactions on Mobile Computing**, [S.l.], 2018.

AHSAN, U.; BAIS, A. A Review on Big Data Analysis and Internet of Things. In: IEEE 13TH INTERNATIONAL CONFERENCE ON MOBILE AD HOC AND SENSOR SYSTEMS, 2016. **Anais...** [S.l.: s.n.], 2016.

ALABA, F. A.; M. OTHMAN IBRAHIM ABAKER T. HASHEM, F. A. Internet of Things security: a survey. In: JOURNAL OF NETWORK AND COMPUTER APPLICATIONS 88. **Anais...** [S.l.: s.n.], 2017. p.10–28.

ANGRISHI, K. Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : iot botnets. , [S.l.], 2017.

B., M.; BROWN, R. Determining embedding dimension for phase- space reconstruction using a geometrical construction. **PHYSICAL REVIEW A**, [S.l.], p.3403, 1992.

BAI, L.; L. YAO S. KANHERE S., X. W. Z. Y. Automatic Device Classification from Network Traffic Streams of Internet of Things. **IEEE 43rd Conference on Local Computer Networks (LCN**, [S.l.], 2018.

BAPTISTA, M. . **Gráficos de recorrência e de poincaré na análise da quantidade de internações por diferentes grupos nosológicos, ocorridas ao longo de uma década, em um hospital de ensino**. 2011. Dissertação (Mestrado em Ciência da Computação) — Programa de Pós-Graduação em Ciências da Saúde - Faculdade de Medicina de São José do Rio Preto, São José do Rio Preto/SP, Brasil.

BERTINO, E.; ISLAM, N. Botnets and Internet of Things Security. In: COMPUTER. **Anais...** [S.l.: s.n.], 2017.

BHATIA, S. K. Adaptive K-Means Clustering. **The AAAI Conference on**, [S.l.], p.695–699, 2004.

BIKMUKHAMEDOV R. F., N. A. F. Lightweight Machine Learning Classifiers of IoT Traffic Flows. **Systems of Signal Synchronization, Generating and Processing in Telecommunications, SYNCHROINFO 2019**, [S.l.], p.1–5, 2019.

CAMINHA, J.; A. PERKUSICH, M. P. A Smart Trust Management Method to Detect On-Off Attacks in the Internet of Things. **Security and Communication Networks**, [S.l.], v.2018, 2018.

CISCO, T. **Cisco 2017 Midyear Cybersecurity Report**. Acessado em setembro de 2017.

DOSHI, R.; N. APTHORPE, N. F. Machine Learning DDoS Detection for Consumer Internet of Things Devices. **Cryptography and Security**, [S.l.], 2018.

E. NTHI L. WILLIAMS, M. S. G. T. P. B. A Supervised Intrusion Detection System for Smart Home IoT Devices. **IEEE Internet of Things Journal**, [S.l.], 2019.

ECKMANN, J. P.; S. OLIFFSON KAMPHORST, D. R. Recurrence Plots of Dynamical Systems. **Europhys**, [S.l.], p.973–977, 1987.

FERRANDO, R.; STACEY, P. Classification of Device Behaviour in Internet of Things Infrastructures: towards distinguishing the abnormal from security threats. **International Conference on Internet of Things and Machine Learning**, [S.l.], 2017.

FRASER, A. M.; SWINNEY, H. L. Independent coordinates for strange attractors from mutual information. **Physical Review Journals**, [S.l.], v.33, 1986.

HAIBO, S.; Z. KECHEN, Z. H. A trust evaluation method for improving nodes utilization for wireless sensor networks. **KSII Transactions on Internet and Information Systems**, [S.l.], v.12, n.3, 2018.

HOANG H. DANG, N. D. H. Detecting Anomalous Network Traffic in IoT Networks. **International Conference on Advanced Communication Technology (ICACT)**, [S.l.], v.5, 2019.

HODO, E.; X. BELLEKENS, A. H. Threat analysis of IoT networks using artificial neural network intrusion detection system. , [S.l.], p.579–584, 2016.

INDRE, I.; LEMNARU, C. Detection and prevention system against cyber attacks and botnet malware for information systems and Internet of Things. **Proceedings - 2016 IEEE 12th In-**

**ternational Conference on Intelligent Computer Communication and Processing, ICCP 2016**, [S.l.], 2016.

J. PINHEIRO ANTONIO, B. J. M.; R., C. D. Packet Padding for Improving Privacy in Consumer IoT. **Symposium on Computers and Communications (ISCC)**, [S.l.], 2018.

J.CANEDO; SKJELLUM, A. Using machine learning to secure IoT systems. In: PRIVACY, SECURITY AND TRUST (PST), 2016 14TH ANNUAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2016.

JEYANTHI, N.; R. THANDEESWARAN, J. V. Rqa based approach to detect and prevent ddos attacks in voip networks. **Journal Cybernetics and Information Technologies**, [S.l.], v.14, p.11–24, 2014.

JEYANTHIM, N.; J. VINITHRA SNEHA, R. T. N. C. S. N. I. A Recurrence Quantification Analytical Approach to Detect DDoS Attacks. **2011 International Conference on Computational Intelligence and Communication Networks**, [S.l.], 2011.

JING, Q.; ATHANASIOS V. VASILAKOS J. WAN, J. L. D. Q. Security of the internet of things: perspectives and challenges. **Wireless Networks**, [S.l.], v.20, n.8, p.2481–2501, 2014.

KOLIAS CONSTANTINOS KAMBOURAKIS, G. S. A. V. J. DDoS in the IoT: mirai and other botnets. In: COMPUTER. **Anais...** [S.l.: s.n.], 2017. v.50, p.80–84.

KUMAR, C. A.; K. BHARGAVI, G. J. A Note on Implementing Recurrence Quantification Analysis for Network Anomaly Detection. **Defence Science Journal**, [S.l.], v.162, p.112–116, 2012.

LIN, Q.; REN, D. Quantitative trust assessment method based on Bayesian network. **2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)**, [S.l.], 2016.

LINDQVIST, P. G. N. The Future of the Internet of Things. **Communications of the ACM**, [S.l.], p.26–30, 2017.

MARWAN, N.; M.CARMEN ROMANO MARCOTHIEL, J. Recurrence plots for the analysis of complex systems. **Physics Reports**, [S.l.], v.438, p.237–329, 2007.

MEIDAN, Y.; M. BOHADANA A. SHABTAI1, J. G. D. M. O. N. T. O. Y. E. ProfilIoT: a machine learning approach for iot device identification based on network traffic analysis. **SAC '17 Proceedings of the Symposium on Applied Computing**, [S.l.], 2017.

MIETTINEN, M.; S. MARCHAL I. HAFEEZ, N. A. A. S. S. T. IoT SENTINEL: automated device-type identification for security enforcement in iot. **2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)**, [S.l.], 2017.

M.MINDLIN, G.; R.GILMORE. Topological analysis and synthesis of chaotic time seri. **Physica D: Nonlinear Phenomena**, [S.l.], v.58, p.229–242, 1992.

MOSENIA ARSALAN JHA, N. K. A comprehensive study of security of internet-of-things. In: IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING. **Anais...** [S.l.: s.n.], 2017. p.586–602.

NGUYEN, T. T.; ARMITAGE, G. A survey of techniques for internet traffic classification using machine learning. **International Journal Of Computers Communications & Control**, [S.l.], v.10, p.56 –76, 2008.

NIE, Y.; MA, Y. A First Look at AMI Traffic Patterns and Traffic Surge for Future Large Scale Smart Grid Deployments. **The Second International Conference on Advanced Communications and Computation INFOCOMP 2012**, [S.l.], 2012.

OTT, E.; SAUER, T. **Coping with Chaos**: analysis of chaotic data and the exploitation of chaotic systems (wiley series in nonlinear science). [S.l.]: Wiley VCH, 1994.

P, S. M. R.; C., C. A. An Architecture Proposal for Network Traffic Monitoring with IoT Traffic Classification Support. **2017 IEEE First Summer School on Smart Cities**, [S.l.], 2017.

PACHECO A. B. LUIS GONDIM J. C. JOAO, B. A. S. P.; E., A. Evaluation of Distributed Denial of Service threat in the Internet of Things. **International Symposium on Network Computing and Applications**, [S.l.], 2016.

PACHECO, J.; HARIRI, S. IoT Security Framework for Smart Cyber Infrastructures. **2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)**, [S.l.], 2016.

POINCAR'E; HENRI. Sur le problème des trois corps et les équations de la dynamique. **Acta Mathematicas**, [S.l.], v.13, p.1–270, 1890.

PROKOFIEV, A. O.; Y. S. SMIRNOVA, V. A. S. A Method to Detect Internet of Things Botnets. **2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)**, [S.l.], 2018.

Q. ASHRAFN, M. H. H. Autonomic schemes for threat mitigation in Internet of Things. **Journal of Network and Computer Applications**, [S.l.], p.112–127, 2015.

QIAN ZHU RUICONG WANG, Q. C. Y. L. W. Q. IOT Gateway: bridging wireless sensor networks into internet of things. **IEEE/IFIP International Conference on Embedded and Ubiquitous Computing**, [S.l.], 2010.

RAUT S. ABHINAV, S. R. K. Anomaly Based Intrusion Detection-A Review. **International Journal Network Security**, [S.l.], v.5, 2014.

RAZAA, S.; L. WALLGRENA, T. V. SVELTE: real-time intrusion detection in the internet of things. **Ad Hoc Networks**, [S.l.], p.2661–2674, 2013.

R.FU; K. ZHENG D. ZHANG, Y. Y. An intrusion detection scheme based on anomaly mining in internet of things. **4th IET International Conference on Wireless, Mobile & Multimedia Networks (ICWMMN 2011)**, [S.l.], 2011.

RIGHI, M. A.; NUNES, R. C. Detecção de DDoS Através da Análise da Quantificação da Recorrência Baseada na Extração de Características Dinâmicas e Clusterização Adaptativa. **XVI Simpósio Brasileiro em Segurança da Informação e Sistemas Computacionais - SBSEg 2016**, [S.l.], 2016.

RIGHI M. A.; NUNES, R. C. Combining Recurrence Quantification Analysis and Adaptive Clustering to Detect DDoS Attacks. **The Cyber Defense Review**, [S.l.], v.1, p.15–28, 2019.

RIGHIM, M. A.; NUNES, R. C. Um Modelo de Sistema de Detecção de Anomalias em Redes de Computadores Baseado na Extração de Características Dinâmicas. **Anais do EATI - Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação**, [S.l.], v.4, p.183–190, 2014.

S. BABAR A. PRASAD, N. S. J. P. Proposed embedded security framework for Internet of Things (IoT). **2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless VITAE 2011**, [S.l.], p.1–5, 2011.

SANTOS M., C. A. An Architecture Proposal for Network Traffic Monitoring with IoT Traffic Classification Support. **First Summer School on Smart Cities**, [S.l.], 2017.

SHAIKH, F.; E. BOU-HARB J. CRICHIGNO, N. G. J. A Machine Learning Model for Classifying Unsolicited IoT Devices by Observing Network Telescopes. **IEEE International Wireless Communications and Mobile Computing Conference (IWCMC 2018)**, [S.l.], 2018.

SHEIKHAN, M.; BOSTANI, H. A Hybrid Intrusion Detection Architecture for Internet of Things. In: INTERNATIONAL SYMPOSIUM ON TELECOMMUNICATIONS (IST'2016) A, 2016. **Anais...** [S.l.: s.n.], 2016.

SINANOVIC, H.; MRDOVIC, S. Analysis of Mirai malicious software. **2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)**, [S.l.], 2017.

SIVANATHAN A. HASSAN L., F. R. A. W. C. V. A. S. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. **IEEE Transactions on Mobile Computing**, [S.l.], v.7, 2018.

SIVANATHAN A. SHERRATT D., G. H. H. R. A. W. C. V. A.; V., S. Characterizing and classifying IoT traffic in smart cities and campuses. **IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2017**, [S.l.], p.559–564, 2017.

SOUSA, B. F. L. M.; Z. ABDELOUAHAB D. CICERO P. LOPES, N. C. S. W. F. R. An Intrusion Detection System for Denial of Service Attack Detection in Internet of Things. **ICC '17 Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing**, [S.l.], 2017.

SPOGNARDI ANGELO DONNO, M. D. D. N. G. A. Analysis of DDoS-Capable IoT Malwares. In: COMPUTER SCIENCE AND INFORMATION SYSTEMS (FEDCSIS), 2017 FEDERATED CONFERENCE ON. **Anais...** [S.l.: s.n.], 2017. p.807–816.

T. GARRETT S. DUSTDAR, E. B. L. P. D. E. Traffic Differentiation on Internet of Things. **2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)**, [S.l.], 2018.

TAKENS, F. Modeling the internet of things: a hybrid modeling approach using complex networks and agent-based models. **Proceedings of a Symposium Held at the University of Warwick 1979/80**, [S.l.], 1980.

THIEL, M.; M. CARMEN ROMANO J. KURTHS, R. M. E. A. T. F. A. Influence of observational noise on the recurrence quantification analysis. **Physics Letters A**, [S.l.], v.171, p.199–203, 1992.

WEBBER, C. L.; MARWAN, J. N. **Recurrence Quantification Analysis**: theory and best practices. springer series: understanding complex systems. springer international publishing, cham switzerland. [S.l.]: Springer, 2015.

WEBBER, C. L.; ZBILUT, J. P. Dynamical assessment of physiological systems and states using recurrence plot strategies. **Journal of Applied Physiology**, [S.l.], v.76, p.965–973, 1994.

YUAN, J.; R. YUAN, X. C. Network Anomaly Detection based on Multi-scale Dynamic Characteristics of Traffic. **International Journal of Computers, Communications & Control (IJCCC)**, [S.l.], p.102–112, 2014.