Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Engenharia Elétrica

Um método para o gerenciamento da confiança na identificação de recursos e detecção de ataques para a Internet das Coisas

Jean Caminha

Tese de Doutorado apresentada à Coordenadoria do Programa de Pós-Gradução em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para obtenção do título de Doutor em Ciências no Domínio da Engenharia Elérica.

Área de Concentração: Processamento da Informação

Angelo Perkusich, Dr.
Orientador

Campina Grande, Paraíba, Brasil

"UM MÉTODO PARA O GERENCIAMENTO DA CONFIANÇA NA IDENTIFICAÇÃO DE RECURSOS E DETECÇÃO DE ATAQUES PARA A INTERNET DAS COISAS"

JEAN CAMINHA

TESE APROVADA EM 25/05/2018

ANGELO PERKUSICH, D.Sc., UFCG
Orientador(a)

ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFCG
Examinador(a)

JOSÉ SÉRGIO DA ROCHA NETO, D.Sc., UFCG
Examinador(a)

EVANDRO DE BARROS COSTA, D.Sc., UFAL
Examinador(a)

HYGGO OLIVEIRA DE ALMEIDA, D.Sc., UFCG
Examinador(a)

CAMPINA GRANDE - PB

C183m Caminha, Jean.
Um método para o gerenciamento da confiança na identificação de recursos e detecção de ataques para a internet das coisas / Jean Caminha. - Campina Grande, 2018.
102 f. : il. color.

Tese (Doutorado em Engenharia Elétrica) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2018.
"Orientação: Prof. Dr. Angelo Perkusich".
Referências.

1. Internet das Coisas. 2. Técnica de Janela Elástica Dinâmica (JED). 3. Recursos Inteligentes e Cooperativos. I. Perkusich, Angelo. II. Título.

CDU 004.738.5(043)

FICHA CATALOGRÁFICA ELABORADA PELO BIBLIOTECÁRIO GUSTAVO DINIZ DO NASCIMENTO CRB - 15/515

Dedicatória

À minha filha Júlia Liz.

Aquela que acredita que todo professor de verdade precisa buscar seu doutorado.

Agradecimentos

Ao meu orientador, Angelo Perkusich, D.Sc., por proporcionar as melhores condições possíveis para o desenvolvimento deste trabalho e minha formação como pesquisador;

À minha esposa, Silane Caminha, D.Sc., pelo companheirismo, paciência e apoio durante esta árdua caminhada;

Aos membros da minha banca de avaliação, Antonio Marcus Nogueira Lima, Dr., Evandro de Barros Costa, D.Sc., Augusto José Venâncio Neto, Dr., José Sérgio da Rocha Neto, D.Sc. e Hyggo Oliveira de Almeida, D.Sc., pelas importantes contribuições efetuadas, assim como os diversos revisores dos artigos submetidos durante este trabalho;

À Universidade Federal de Mato Grosso pela concessão do afastamento para a realização da minha formação;

À Universidade Federal de Campina Grande pela acolhida em seu Programa de Pós-Graduação;

Aos colegas do Instituto de Computação da UFMT, que assumiram minhas atribuições durante o período de afastamento;

Aos novos amigos que fiz no Laboratório de Sistemas Embarcados e Computação Pervasiva, Lenardo Chaves e Silva, D.Sc., Frederico Moreira Bublitz, D.Sc., Mirko Barbosa Perkusich, D.Sc., Renata Mendonça Saraiva, M.Sc., Felipe Ramos, M.Sc. e Ana Soares, M.Sc., pela troca de experiências e apoio durante todas as minhas passagens por Campina Grande.

À Fundação de Amparo à Pesquisa de Mato Grosso (Fapemat), pela bolsa concedida e financiamento desta pesquisa;

E por último, mas não menos importante, ao povo de Campina Grande, que sempre me fizeram sentir um filho da Paraíba.

Resumo

A Internet das Coisas (IoT) é um conceito relacionado à conexão de objetos cotidianos à Internet, transformado-os em recursos inteligentes e cooperativos. Apesar de possuir muitas semelhanças com as redes de sensores sem fio, a IoT apresenta requisitos especiais, como a integração de redes, contexto semântico, conexão a sistemas legados e objetos heterogêneos, além da necessidade de proteção contra ataques específicos à segurança. Os recursos da IoT cooperam entre si solicitando e oferecendo serviços. Em ambientes heterogêneos e complexos, esses recursos precisam confiar uns nos outros. O conceito de enxame (*swarm*) em IoT descreve a cooperação de dispositivos altamente independentes e heterogêneos para executar tarefas. Os componentes de um sistema de enxame devem descobrir de forma transparente outros objetos. A descoberta semântica é uma abordagem para executar esta tarefa e pode ser realizada de forma automática ou manual, mas que normalmente não leva em conta requisitos de segurança, como a confiança (*trust*) e potenciais atacantes. Neste trabalho é apresentado um método baseado em aprendizado de máquina em conjunto com uma técnica de janela elástica dinâmica (JED), que realiza o reconhecimento semântico de recursos IoT, auxiliando a integração de infraestruturas e serviços, colaborando para o gerenciamento da confiança e identificando ataques. Com a aplicação do método proposto foi possível reconhecer semanticamente recursos IoT com 96% de precisão em um conjunto de dados real e ataques do tipo *On/Off* em ambiente simulado. Em comparação com outros estudos, o método foi 95% mais rápido na identificação ataques do tipo *On-Off*. Com a execução do método, de maneira inovadora, é possível diferenciar também nós atacantes de nós defeituosos.

Abstract

The Internet of Things (IoT) is a concept which describes how everyday objects are connected to the Internet, transforming them into intelligent and cooperative resources. Although IoT has many similarities with wireless sensor networks, IoT presents special requirements such as network integration, semantic context, connection to legacy systems and heterogeneous objects, and the need to protect against specific security attacks. IoT's resources cooperate with each other by requesting and offering services. In heterogeneous and complex environments, these features need to trust each other. The swarm concept in IoT describes the cooperation of highly independent and heterogeneous devices to perform tasks. Components of a swarm system must seamlessly discover other objects. Semantic discovery is an approach to perform this task and can run automatically or manually, but it does not normally address security requirements, such as trust and potential attackers. In this thesis, a method based on machine learning and a dynamic elastic window technique (JED) is presented, that automatically assesses the IoT resource trust, recognizes IoT semantic attributes, helping the integration of infrastructures and services, evaluating service provider attributes. The proposed method collaborates with trust management and attacks identification. In simulated and real-world data, this method was able to recognize IoT semantic attributes, On-Off attacks and fault nodes with a precision of up to 96%.

Lista de símbolos e abreviaturas

API	Interface de Programação de Aplicativos	16
IoT	Internet das Coisas	12
JED	Janela Elástica Dinâmica	47
M2M	Máquina-a-máquina	31
OA	Ataques On-Off	20
REST	Trânsferência de Estado Representacional	37

Lista de Tabelas

Lista de Figuras

Sumário

Capítulo 1

Introdução

O trabalho descrito nesta tese envolve a concepção e o desenvolvimento de um método para a identificação de recursos e detecção de ataques para a Internet das Coisas (IoT), viabilizando a integração de objetos IoT por intermédio do reconhecimento semântico de atributos, fornecendo informações para o gerenciamento da confiança (*trust*). Este método ainda possibilita diferenciar nós atacantes de nós defeituosos.

Este capítulo apresenta uma visão geral da tese. Inicia-se por uma contextualização sobre IoT e uma breve explanação sobre os problemas em aberto para reconhecimento semântico para integração e segurança. Prossegue-se com a delimitação do problema de pesquisa, apresentação das hipóteses consideradas, das delimitações e dos objetivos da tese e, finalizando com a discussão sobre a relevância do problema e da organização geral deste documento.

1.1 Contextualização e Motivação

A Internet das Coisas (IoT) é um conceito computacional que utiliza uma infraestrutura de abrangência escalável, auto-configurável, baseada em padrões e protocolos de comunicação interoperáveis onde objetos físicos e virtuais possuem identidades, atributos, personalidades e interfaces inteligentes, perfeitamente integradas em uma rede de informação. A IoT descreve a transformação de objetos comuns em recursos inteligentes, com capacidades de sensoriamento, atuação e comunicação. A IoT possui potencial para aplicações em domótica, auxílio à saúde, agricultura de precisão, logística, energia, varejo, cidades inteligentes, entre outras [1]. A IoT

Figura 1.1: O conceito de swarm IoT aplicado em uma cidade inteligente.

Fonte: Elaborada pelo autor.

define como outros objetos e sistemas serão conectados, promovendo uma infraestrutura distribuída e produzindo um ambiente ou contexto inteligente pela transformação do fluxo de dados em informações relevantes para serviços e usuários [2].

Em um contexto de *Smart City*, uma infraestrutura IoT pode ser demandada a suportar interações heterogêneas, de forma a permitir a cooperação independente entre uma grande quantidade dispositivos. Este tipo de abordagem é denominada como um sistema *Swarm* (enxame) [3]. Os componentes de um sistema *Swarm* são disponibilizados de forma heterogênea e dinâmica em uma área de larga abrangência geográfica. Os dados (temperatura, iluminação, ruído, etc.) são comumente gerados de forma contínua e multi-modal (texto, numérico, multimídia, etc.), não permitindo facilmente identificar todos os dados dos recursos para responder às solicitações de usuários neste tipo de ambiente distribuído [4].

Pesquisadores ainda divergem sobre as definições relacionadas às arquiteturas de IoT. Os trabalhos de Khan et. al. [5] e Yang et. al. [6] defendem que a arquitetura IoT é composta de cinco diferentes camadas: Sensoriamento, Acesso, Rede, *Middleware* e Aplicação. Os estudos de Khan et. al. [5] e Wang et. al. [7] sustentam três camadas principais: Camada de Percepção (sensoriamento); Camada de Rede e Camada de Serviços (ou aplicação), ou como quatro camadas, como defendido por Tan et. al.[8]: Camada física; Camada de Transporte; Camada de *Middleware* e Camada de aplicações. Assim como essa definição, outras questões relacionadas à disponibilização de serviços (aplicações) e requisitos de segurança [9] encontram-se em aberto.

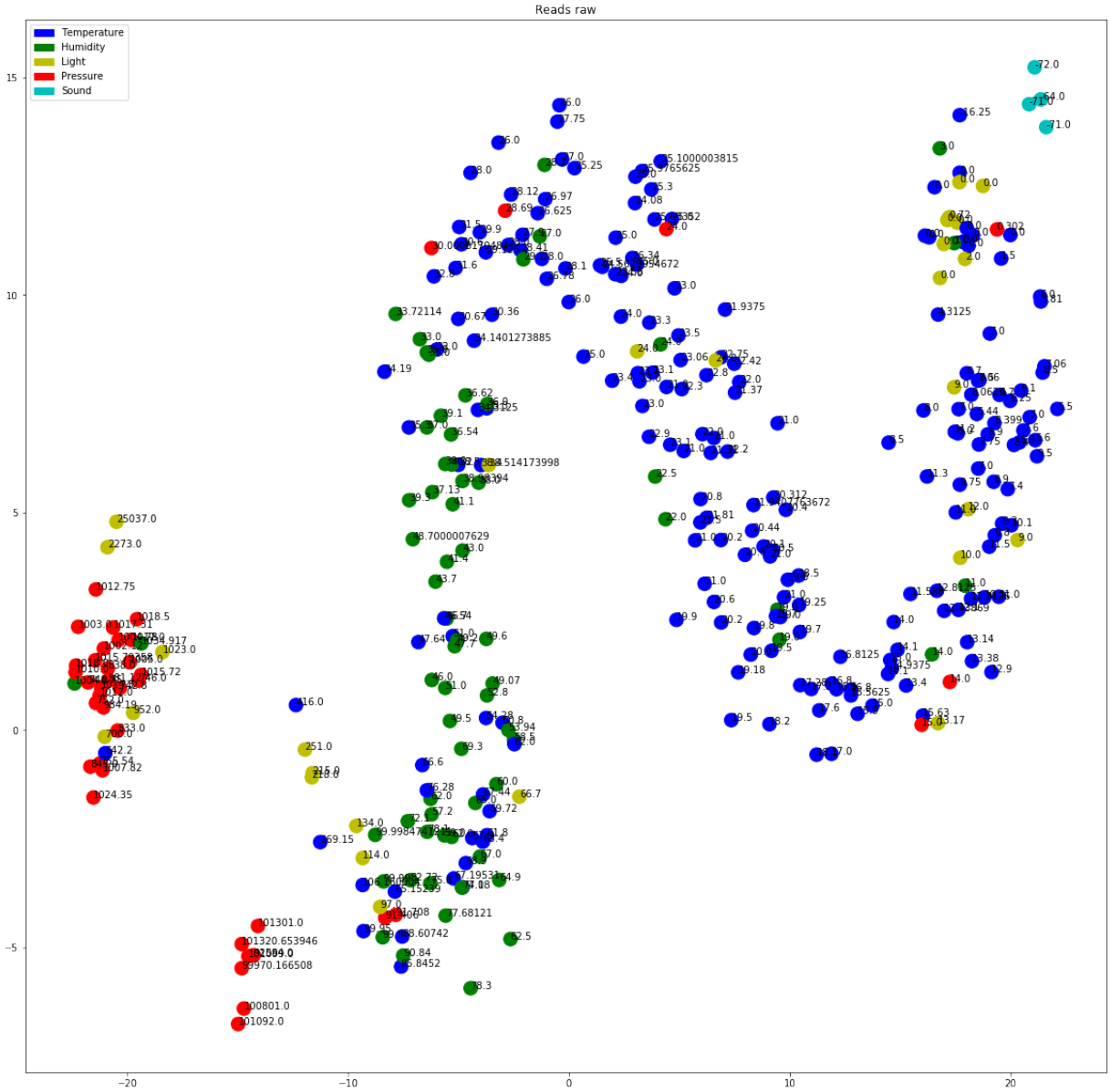
Arquiteturas de IoT podem ser demandadas a conectar diversos dispositivos e sub-redes altamente heterogêneas em larga escala, com propriedades desconhecidas e utilizando diferentes métodos de configuração, o que gera várias lógicas de processos aplicados ao mesmo tipo de solução [10]. Ainda, múltiplos níveis de intraoperabilidade requerida pela IoT devem tratar problemas relacionados ao gerenciamento da convergência entre a Internet tradicional com dispositivos heterogêneos e restritos no que tange a capacidade de processamento, armazenamento e comunicação [11]. Assim, as alternativas de protocolos e abordagens leves de IoT devem elencar alternativas em face à complexidade de utilização de protocolos mais robustos como o HTTP, TCP/IP em equipamentos com baixa capacidade computacional.

O nível de serviços em IoT também enfrenta alguns desafios técnicos que necessitam serem tratados [12]. O desenvolvimento de uma arquitetura orientada à serviços (SOA) específicos para IoT é complexo, devido às limitações de performance e custos e problemas de escalabilidade, dado ao grande número de objetos a serem conectados. A heterogeneidade das redes de conexão é outro ponto de atenção, imposta pelos vários tipos de tecnologias e a carência de uma plataforma comumente aceita, bem como um serviço transparente e padronizado para a nomeação das aplicações, endereçamento, identificação e otimização, em nível de arquiteturas e protocolos. Da mesma forma, a ausência de uma linguagem de descrição globalmente aceita, dificulta o desenvolvimento e a integração de serviços, somadas pela complexidade de integrar sistemas legados em uma arquitetura unificada de informação, devendo suportar o crescente número de dispositivos conectados e o aumento do fluxo de dados.

Dados obtidos por objetos IoT podem não possuir muito valor para os sistemas e usuários a menos que seu significado e contexto sejam compreendidos. A construção de aplicações no qual dados heterogêneos relacionados a IoT, combinando com informações tradicionais é uma tarefa desafiadora para várias indústrias [9]. A IoT expande os conceitos de computação ubíqua habilitando objetos inteligentes a cooperarem em seu ambiente (contexto) e executarem tarefas coletivas, nas dimensões objeto-objeto, humano-objeto, ambiente-objeto e negócio-objeto [13]. Entretanto, para oferecer todo seu potencial de serviços, a IoT demanda que os recursos sejam localizados, acessados, gerenciados e interajam com outros objetos (Figura 1.1). Para isso, a interoperabilidade possui um papel importante para a realização da visão da IoT [10].

Em um conjunto de provedores de serviços, alguns recursos IoT podem ter atributos semânticos idênticos (por exemplo, identificação, localização, fornecedor), bem como seus valores ob-

Figura 1.2: Exemplos de leituras extraídos dos canais públicos da plataforma *Thingspeak* [14].



Fonte: Elaborada pelo autor.

servados. Na Figura 1.2, apresenta-se um exemplo de como algumas leituras de sensores de temperatura (pontos azuis) se sobrepõem às leituras de sensores de umidade (pontos verdes). Além disso, outros problemas de interoperabilidade da IoT estão relacionados à volatilidade dos recursos, usuários e dispositivos, dependência do contexto, limitação dos recursos de processamento e armazenamento dos dispositivos, além das necessidades de capacidades autônomas [3].

A interoperabilidade é um dos principais desafios para pesquisadores e fabricantes no desenvolvimento em IoT, pela crescente quantidade de dispositivos heterogêneos, demanda de serviços,

aplicações e níveis de integração necessárias, desde a conexão com as redes, redes entre redes e entendimento das informações (semântica). No nível semântico, problemas relacionados à atributos semanticamente similares possuírem nomes diferentes, outros semanticamente diferentes mas com nomes iguais, conflitos de nomes e problemas de generalização precisam ser superados e evoluídos [13]. Além disso, as tratativas de interoperabilidade em IoT esbarram em dificuldades relacionadas à volatilidade de recursos, usuários e dispositivos; na dependência do contexto da aplicação e restrições de processamento, armazenamento e comunicação, ampliadas pelas exigências de capacidades autonômicas, descoberta, gerenciamento e transparência pelos usuários [15].

Implementações comuns em IoT são realizadas no modo unimodal fechado, como em aplicações de sensores de temperatura em aplicações de HAVC (heating, ventilation, and air conditioning) ou sensores de movimento em serviços de segurança. A integração entre esses dois tipos de soluções, em um mesmo contexto (edifício) facilitaria o desenvolvimento de novos serviços, como o gerenciamento eficiente de energia [16].

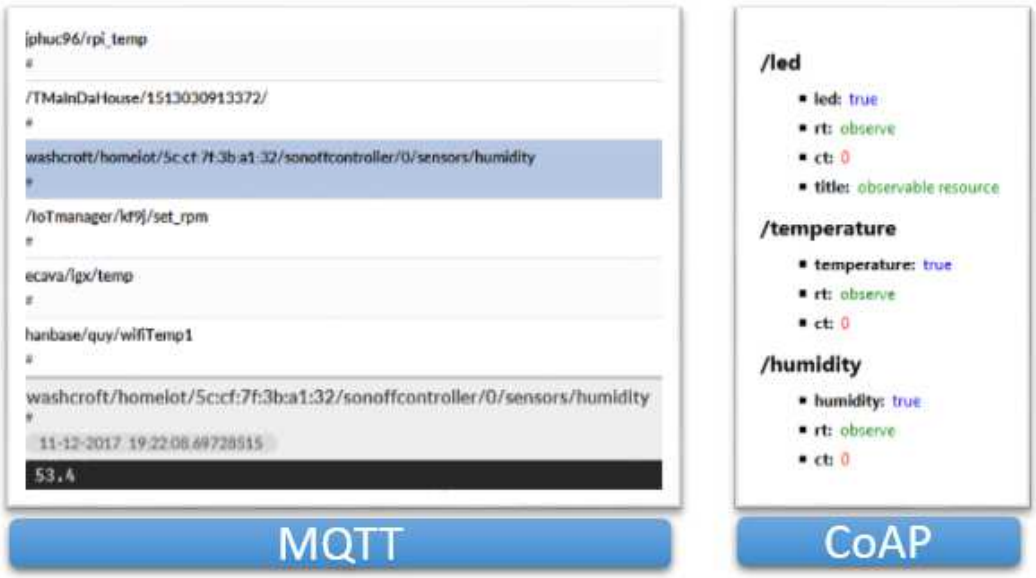
Engenheiros de empresas envolvidas com o desenvolvimento da IoT defendem a visão de integração por meio de plataformas como serviço, como a IBM/Bluemix [1] e ThingsSpeak [2]. O problema encontrado nessa abordagem é devido à diferença entre suas *Application Programming Interface* (APIs) e modelos de dados, que dificultam para estas plataformas se integrarem, impedindo o desenvolvimento de aplicações convergentes [17].

Outra abordagem para a interoperabilidade em IoT é realizada pela padronização através de projetos regionais de especificações e aplicações em larga escala, como o SENSEI, SensorWeb, Open Geospatial Consortium (OGC), SemSorGrid4Env, Exalted, Kno.e.sis Center, CSIRO, Spanish Meteorological Agency e o Sensor Web Enablement (SWE) [2] [16]. A padronização utilizada como maneira de resolver os problemas de interoperabilidade aplicada de modo prematuro eleva o risco de sufocar a inovação. Iniciativas comerciais ou a impossibilidade de atuação da comunidade podem criar ilhas de soluções IoT não convergentes [18].

Os protocolos IoT amplamente utilizados, como o Message Queue Telemetry Transport (MQTT) e o Constrained Application Protocol (CoAP), são flexíveis para registrar os atributos de recursos, mas não oferecem descrições semânticas ricas ao publicar seus dados. Os desenvolvedores deci-

[1] https://www.ibm.com/cloud-computing/bluemix/internet-of-things
[2] https://thingspeak.com/

Figura 1.3: Exemplos de saída da listagem de recursos via MQTT e CoAP.



Fonte: Elaborada pelo autor.

dem quais atributos um determinado recurso possuirá e como serão referenciados. Os usuários no entanto, podem enfrentar dificuldades para encontrar um serviço ou recurso desejado. Como exemplo, uma breve consulta em um *broker* público do MQTT [3] (Figura 1.3) retorna vários objetos IoT (inclusive alguns do mesmo tipo) rotulados de maneiras diferentes.

Outros sistemas de suporte a IoT, como o Core e o mDNS-SD [19], estão focados em recursos de descoberta em uma rede, sem descrição do serviço semântico ou inferência a qualquer informação útil aos assinantes. O protocolo Universal Plug and Play (uPNP) fornece algumas informações relevantes sobre atributos de recursos, mas os dispositivos uPNP são sujeitos a um processo formal de padronização, o que pode dificultar o desenvolvimento de novos serviços.

Uma possível solução para promover a integração semântica é o uso de ontologias (forma de representação de conhecimento). A combinação de diferentes descrições ontológicas possibilita a criação de uma representação integrada a partir de várias outras em um mesmo domínio do conhecimento. Entretanto, a tarefa de refletir na ontologia destino o confronto de todas as similaridades e diferenças observados nas ontologias de origem é complexa, sendo observado que em vários métodos de combinações publicados, pouco melhoram em relação a indicadores de eficiência (*precision/recall*) [20]. Pesquisas para o cálculo da similaridade entre conceitos na construção de ontologias e os modelos carecem de um estudo profundo da similaridade estrutural.

[3] broker.hivemq.com

Este problema faz com que as ontologias reflitam apenas conceitos linguísticos em detrimento de definições semânticas [21].

Como diferentes abordagens podem utilizar ontologias heterogêneas, a anotação ou mapeamento semântico é usado com sucesso, evoluindo ontologias existentes para o entendimento de novos recursos ou outras arquiteturas. O mapeamento semântico utiliza tecnologias para identificar textos, imagens, vídeos, programas, serviços em informações compreendidas pelas máquinas e usuários, podendo ser realizada por métodos manuais ou (semi) automatizados. Anotações manuais demandam conhecimento específico e alto custo para realização em larga escala [22]. Métodos automáticos são divididos em duas categorias: reconhecimento de padrões e aprendizado de máquina. Normalmente os métodos baseados em aprendizado de máquina funcionam relativamente melhor [23].

Ao mesmo tempo, a conexão de várias tecnologias heterogêneas para a disponibilização de serviços para IoT em vários cenários de aplicação, geram requisitos especiais de segurança das informações e privacidade [24]. As novas abordagens de aplicações suportados pela IoT como a Internet de Veículos, Internet da Energia, entre outros, produzirá bilhões de interações entre dispositivos e pessoas [4]. A flexibilidade da infraestrutura requerida, os diferentes padrões de comunicação e controle, como também a quantidade e volatilidade de objetos conectados, aumentam os riscos à segurança. Tratativas de proteção tradicionais não atendem em muitos casos, às necessidades de segurança para a IoT [25]. Para garantir a segurança em serviços IoT, requisitos de autenticação, confidencialidade, controle de acesso, privacidade, reforço (*enforcement*), confiança (*trust*), *middlewares* seguros e segurança móvel devem ser desenvolvidos. Requisitos tradicionais para um sistema de autenticação, autorização, controle de acesso e de não repúdio são elevados a um nível maior de complexidade em ambientes IoT, pela existência de um ambiente compartilhado entre objetos e usuários, como também a utilização de equipamentos com poucos recursos computacionais, impedindo-se de implementar métodos avançados de criptografia [26].

O European Research Cluster on the Internet of Things (IERC) AC4 lançou em março de 2015 um conjunto de melhores práticas e recomendações para interoperabilidade semântica para IoT. Salvo melhor conhecimento, não é conhecida nenhuma outra abordagem semelhante para a aplicação de metodologias web semânticas e melhores práticas para IoT. O grupo menciona a necessidade de superar os seguintes desafios: (1) um modelo unificado para anotar semanticamente os dados da IoT, (2) mecanismos de *reasoning* , (3) abordagem de dados interligados, (4)

integração com aplicações existentes, (5) design leve e versões para ambientes restritos, além do (6) alinhamento entre diferentes vocabulários. Entretanto, o IERC AC4 não faz referência a ferramentas concretas que incentivam as melhores práticas de web semântica, o uso de metodologias assegurar a interoperabilidade entre os aplicativos de IoT e a reutilização do conhecimento de um domínio semântico já estruturados [27].

O desafio geral para a IoT, nos aspecto semântico, é a composição do serviço em ambientes inteligentes devido ao grande número de dispositivos de vários fornecedores e diferentes domínios de aplicativos. As descrições de serviços precisam ser abstratas o suficiente para cobrir os vários domínios das aplicações, permitindo a composição automática do serviço. Além disso, ambientes inteligentes são essencialmente dinâmicos, com dispositivos se juntando, movimentando e falhando temporariamente. Assim, provedores de serviços e consumidores encontram-se fracamente acoplados, sensíveis às mudanças do ambientes [28].

As tecnologias de Web Semântica baseadas no formalismo de representação interpretável por máquina mostraram-se promissoras para descrever objetos, compartilhar e integrar informações e inferir novos conhecimentos em conjunto com outras técnicas inteligentes de processamento. No entanto, a natureza dinâmica e restrita de recursos da IoT requer que considerações especiais de projeto sejam levadas em consideração para aplicar efetivamente as tecnologias semânticas nos dados do mundo real. Dentro da comunidade IoT, cada projeto ou plataforma desenvolve sua própria abordagem semântica dificulta a interoperabilidade. Um grande desafio seria reutilizar tanto quanto possível as definições semânticas existentes para facilitar a interoperabilidade entre projetos internacionais, plataformas, sistemas e serviços. Para não especialistas em web semântica, este é um desafio real [29].

Os recursos de IoT podem interagir entre si requisitando e provendo serviços, ocasionalmente de forma oportunística. Neste contexto, usuários mal intencionados podem utilizar recursos conectados a IoT para realizarem ataques baseados no abuso de confiança. Ataques à confiança podem ser dos tipos *ballot-stuffing*, *bad-mouthing*, *On-Off* e ataques de auto-promoção [30]. Para maximizar a segurança de todo o sistema, é importante avaliar a dos provedores de serviço nos ambientes IoT [31]. Um recurso IoT pode atuar como um provedor ou requisitante de serviços. Um requisitante de serviços necessita escolher e confiar no melhor provedor disponível. Um provedor malicioso, através da disponibilização de informações incorretas, poderá comprometer serviços em IoT. Ataques à confiança (*Trust*) e suas contramedidas são também um problema em

aberto sendo tratado por vários pesquisadores [32] [33].

Os ataques *On-Off* (OA) são considerados um tipo de ataque à confiança. Um recurso malicioso pode prover de forma aleatória um bom ou mal serviço ao sistema, de forma a evitar ser identificado como um nó não confiável. Do mesmo modo, um atacante OA pode interagir de forma diferenciada com seus vizinhos, para alcançar diferentes percepções sobre sua confiabilidade. Este tipo de ataque é considerado de difícil detecção por métodos tradicionais de gerenciamento da confiança [34]. Além disso, nem todos os recursos que se comportam como atacantes OA (gerando informações corretas (confiáveis) e erradas aleatoriamente) são maliciosos. Os dispositivos podem se encontrar em um estado de mal-funcionamento. A correta separação entre atacantes e dispositivos com defeito é útil para o desenvolvimento de soluções de recuperação para serviços IoT [35].

1.2 Cenários de Aplicação

Uma típica aplicação de IoT é o sensoriamento participativo. Este tipo de serviço utiliza sensores disponibilizados pelos usuários como seus *SmartPhones* ou outros dispositivos embarcados para coletar informações em um determinado contexto. Iniciativas como o CitySense [36] na cidade de Málaga na Espanha, utiliza um aplicativo que coleta dados de temperatura, umidade, barulho urbano, entre outros fenômenos, através dos *SmartPhones* dos cidadãos e envia para a infraestrutura da prefeitura para posterior análise, entendimento da cidade e criação de políticas públicas. Este mesmo aplicativo também entende sinais de *beacons* de *bluetooth* espalhados pela cidade, transmitindo informações relevantes da prefeitura em um determinado contexto regional. Projetos como o Smartcitizen.me [37] utilizam um kit de sensores que realizam leituras ambientais e enviam os dados para acesso Web ou atrás de redes sociais. Essas duas fontes de dados poderiam ser integradas, aumentando a capilaridade do sensoriamento e criação de novos serviços para os usuários.

Outra aplicação que exige ao máximo as capacidades adaptativas de IoT é o gerenciamento da cadeia de suprimentos de alimentos (Food-IoT). Por ser um processo extremamente distribuído e complexo, com demandas de abrangência geográficas e temporais, possuindo um grande número de `steakholders` com diferentes objetivos. Diversas iniciativas de IoT buscam atender demandas específicas como rastreabilidade, visibilidade, controle, agricultura de precisão, produção de

alimentos, processamento, armazenamento, distribuição e consumo [12]. Pela existência de várias arquiteturas especificas, muitas vezes utilizando padrões e tecnologias proprietárias, torna-se difícil a tarefa de padronização que possibilitaria o desenvolvimento de serviços integrados. Uma arquitetura inteligente pode oferecer soluções de integração entre estes agentes, reconhecendo, classificando e padronizando recursos (produtos e dispositivos) e informações, integrando-os a cadeia de suprimentos.

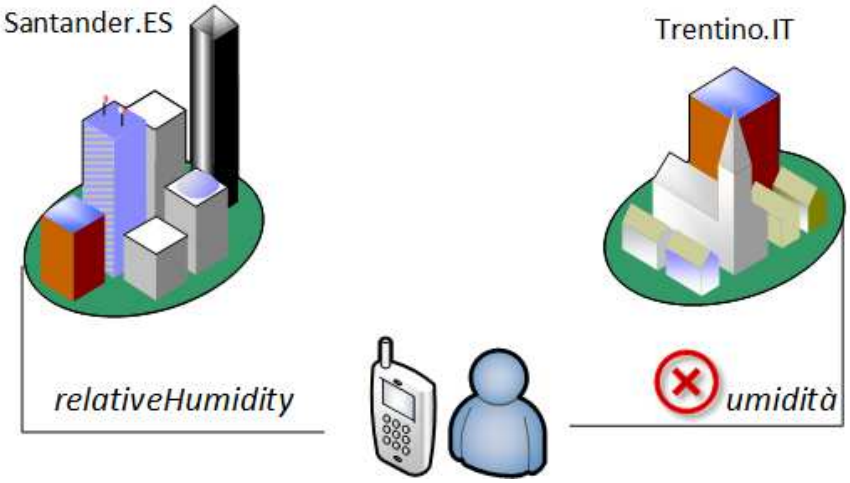
O gerenciamento inteligente do trânsito através da IoT é objeto de estudo em várias *SmartCities* [2]. Um grande número de soluções e infraestruturas são implementadas para a otimização do tráfego, gestão de vagas de estacionamento e auxílio à veículos inteligentes, dentre outras aplicações. Devido a quantidade e heterogeneidade de objetos (veículos, vagas, dispositivos e *gateways*) aliado a dificuldade de provisionar (um veículo pode viajar para outra *SmartCity*) e a diversidade de tecnologias, uma arquitetura não adaptável encontra dificuldades para conectar mais objetos e oferecer outros serviços. Em um contexto mais específico, uma arquitetura IoT inteligente e integradora possibilitaria que um serviço de informação sobre vagas de estacionamento em uma cidade possa ser oferecido, reconhecendo os dados provenientes de várias infraestruturas heterogêneas de estacionamentos privados e públicos.

O gerenciamento de resíduos urbanos é uma outra aplicação de IoT para *SmartCities*. Prefeituras utilizam lixeiras inteligentes conectadas a internet equipadas com sensores para verificar seu preenchimento, que auxiliam o planejamento de rotas de coletas, reduzindo os custos. Em várias implementações, todos os objetos e a infraestrutura pertencem a prefeitura da cidade. Uma arquitetura inteligente facilitaria a conexão de lixeiras dos condomínios e residências, desenvolvidas por diferentes fabricantes à infraestrutura da cidade, padronizando as informações transmitidas. Do mesmo modo, sistemas de gerenciamento de condomínios e residências poderiam consumir informações diretamente das lixeiras públicas sem a necessidade do uso de dicionários de dados ou permissões de acesso especiais.

Outro problema está relacionado a aplicativos ou dispositivos embarcados concebidos para funcionar em uma determinada *SmartCity* ou arquitetura, podendo não atuar adequadamente em uma outra cidade, pelo desconhecimento dos parâmetros de configuração das mesmas (endereços, protocolos, etc.) e recursos, como também pela complexidade enfrentada por uma empresa ou desenvolvedor em criar um mesmo software compatível com todas as cidades.

Para melhor ilustrar esse problema, tomamos com exemplo um aplicativo para *SmartPhone*

Figura 1.4: Aplicação IoT em duas diferentes SmartCities.



Fonte: Elaborada pelo autor.

que auxilia pessoas com problemas respiratórios, que para entregar seu serviço, necessita de informações sobre a qualidade do ar de uma determinada cidade. Ele foi hipoteticamente desenvolvido para da cidade de Santander na Espanha e funciona perfeitamente nesta *SmartCity*. Quando realiza a consulta na infraestrutura sobre a condição da umidade do ar em uma determinada região da cidade, recebe uma leitura com um atributo do tipo ”relativeHumidity” (Figura 1.4), efetuando os cálculos necessários oferecendo o serviço ao usuário.

Todavia, este mesmo aplicativo não funcionaria corretamente na cidade de Trentino na Itália. Ao realizar a mesma requisição a infraestrutura, esta *SmartCity* retornaria o atributo do tipo ”umidità” (Figura 1.4), a qual não seria reconhecido pelo aplicativo e impossibilitando seu funcionamento. Este cenário é agravado no contexto de IoT, em virtude do constante desenvolvimento de novos dispositivos e serviços, bem como seu dinamismo na inclusão e exclusão em arquiteturas distintas, com diferentes capacidades e usos, feito tanto por pessoas ou outras máquinas [38].

1.2.1 Delimitação do Problema

O conceito de IoT demanda a conexão de diversos recursos e arquiteturas, com funções específicas ou originárias de implementações legadas. Por ser um campo novo de pesquisa, a carência de padronização acarreta o desenvolvimento de iniciativas individuais de pesquisadores e fabricantes, criando soluções heterogêneas ou isoladas, o que faz uma aplicação, dispositivo ou serviço para IoT funcionar em um ambiente conhecido ou utilizando equipamentos de um mesmo fornecedor

e ao mesmo tempo, ser incompatível em outro cenário. Por outro lado, a integração de recursos e informações que geraria novos serviços é prejudicada pela falta de conhecimento do contexto e conexão com outras arquiteturas. Cada iniciativa define sua própria abordagem ou linguagem para a descrição destes contextos e dispositivos, muitas vezes não compatíveis com outras arquiteturas, gerando dificuldades para integração. A previsão de cerca de 30 bilhões de dispositivos IoT conectados até 2020 [4], bem como o aumento de suas capacidades e necessidades de novos serviços, irá demandar metodologias eficientes para provisionar os elementos de arquiteturas heterogêneas em serviços integrados no mesmo contexto.

Além da necessidade de atendimento aos requisitos técnicos de comunicação entre dispositivos heterogêneos, o uso e entendimento da semântica das informações e atributos dos recursos deve ser evoluído para a integração e disponibilização dos serviços de IoT. Soluções para o mapeamento ou anotações semânticas são abordagens que colaboram para a criação de aplicações IoT interoperáveis. Entretanto, os métodos de mapeamento semânticos tradicionais possuem deficiências no tratamento do dinamismo exigido pelas arquiteturas IoT, quer seja por utilizar tratativas manuais, tecnologias adaptadas como o *Service-Oriented Architecture* (SOA) ou mesmo pela ausência de padrões. Adicionado a este problema, uma vez que esta integração semântica seja superada, ainda é necessário verificar o nível de confiança das informações utilizadas como também a detecção de anomalias, prevenindo ataques à segurança.

Com relação as ameaças à segurança, nem todos os nós que possuem um comportamento análogo a um atacante são necessariamente nós maliciosos. Um determinado nó podem se encontrar em um estado de mal funcionamento e sua identificação permite aos administradores de serviços IoT executarem as tarefas apropriadas para a proteção ou conserto destes recursos do sistema.

Pelo exposto, faz-se necessário pesquisar formas mais eficientes e dinâmicas de se realizar a integração semântica de recursos e arquiteturas de IoT, ao mesmo tempo verificando a confiança das informações providas pelos objetos e identificando ataques.

1.2.2 Metodologia

Com o propósito de investigar de forma estruturada a solução para o problema apresentado, foi desenvolvido um plano de pesquisa baseado na seguinte metodologia: Esta pesquisa possui

[4] http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

finalidade prática, pois visa propor um método para tratar o problema de integração em IoT, em especial seu contexto semântico, ao mesmo tempo em que aprimora o gerenciamento da confiança, de forma a colaborar com a expansão segura de serviços para a IoT.

Este trabalho utilizou a abordagem quantitativa para evidenciar a eficiência do método proposto, medindo sua precisão na identificação semântica de objetos IoT e ataques à confiança, de forma a compará-lo com outras abordagens disponíveis na literatura. As hipóteses estudadas foram constituídas através do método hipotético-dedutivo, observando as necessidades de integração e confiança das arquiteturas IoT.

Os procedimentos de pesquisa bibliográfica e pesquisa-ação foram adotados para o entendimento da problemática e situação atual dos problemas investigados, como também para a validação do método proposto. O levantamento do estado da arte foi realizado através de uma profunda revisão sistemática dos estudos publicados, utilizando a metodologia de *Snowballing*, onde referências e citações também são verificadas. A pesquisa-ação foi empregada, utilizando cenários reais e simulados, para desenvolver o método e avaliar os resultados apresentados.

Com o objetivo de aplicação prática dos resultados desta pesquisa, os conjuntos de dados utilizados estão relacionados a projetos de *Smartcities* publicamente disponíveis ou plataformas abertas de publicação de dados de sensores. Os dados relacionados à ataques foram gerados através de simulação, de forma a manter a coerência e compatibilidade com as informações reais oriundas das *Smartcities* usadas.

O método proposto também utiliza em parte, técnicas de aprendizado de máquina para a classificação das leituras. Visando a robustez dos resultados, comparações de desempenho, como também a possibilidade de replicação do estudo por outros pesquisadores, optou-se pelo uso de bibliotecas já testadas e aceitas no meio acadêmico à uma implementação direta dos classificadores.

E por fim, as medições realizadas para validar a solução desenvolvida estão baseadas nas métricas comumente utilizadas para este tipo de investigação e comparadas a outros estudos, quando disponíveis.

1.3 Relevância da Tese

Os projetos de arquiteturas e soluções para IoT são normalmente baseados em modelos unimodais, que descrevem todas as camadas de trabalho interagindo de forma conhecida e previsível. Os requisitos de integração com outras infraestruturas demandam tarefas técnicas e acordos administrativos para a disponibilização das informações. Além disso, muitas iniciativas de IoT são realizadas por empresas ou projetos regionais, dificultando a integração, agravada pela falta de padronização [39].

O entendimento do contexto semântico de IoT pode colaborar de modo eficaz em tarefas de integração. Através do reconhecimento semântico automatizado de recursos, é possível compatibilizar recursos, aplicações e usuários, novos ou provenientes de sistemas legados em um contexto cooperativo. Muitas metodologias de reconhecimento e anotação semântica são utilizadas em IoT, porém com limitações relacionadas ao uso de ferramentas manuais ou semi-automatizadas, que necessitam de conhecimento e trabalho dos administradores de sistemas, arquiteturas centralizadas ou fortemente acopladas. Além disso, há uma carência de soluções integradas e equilibradas que ofereçam serviços a aplicativos e usuários atendendo simultaneamente a requisitos de segurança [30].

Neste contexto, o método proposto neste trabalho tem por objetivo promover a integração de recursos IoT através do reconhecimento semântico automatizado de atributos, ao mesmo tempo que viabiliza o gerenciamento da confiança (*Trust*) e detecta ataques do tipo OA. Este método combina a aprendizagem de máquina com uma análise de dados temporal para entender corretamente as variações das informações relacionadas aos recursos. Compatível com outras infraestruturas e padrões abertos, este método colabora com usuários, dispositivos, aplicações e serviços de IoT identificando, classificando e padronizando recursos que podem ser conectados, reconhecendo seus atributos semânticos, ao mesmo tempo em que analisa a confiança destas interações, identificando ataques e nós defeituosos.

As contribuições desta pesquisa são:

(i) A proposição de método para a realização da integração de sistemas IoT, através do reconhecimento semântico de atributos relacionados aos recursos em seu contexto semântico, com o mínimo ou nenhuma intervenção administrativa, permitindo a padronização e uso por sistemas e serviços;

(ii) A aplicação deste método no gerenciamento da confiança (Trust) de IoT para a análise e identificação de recursos a serem selecionados pelos sistemas e usuários;

(iii) Evoluir a segurança de IoT na identificação de atacantes ao gerenciamento da confiança, em especial, ataques do tipo On-Off;

(iv) A utilização deste método para a diferenciação entre atacantes OA e recursos em estado de mal funcionamento;

(v) Design e execução de uma prova de conceito usando dados simulados e reais de um projeto de cidade inteligente, demonstrando a eficiência do método.

1.4 Hipótese

Para realizar a integração flexível e confiável de objetos IoT, considerando o problema abordado neste trabalho, é necessário o reconhecimento e análise semântica confiável dos atributos e dados dos recursos IoT. Desta forma, para o problema abordado neste trabalho é considerada a hipótese de que: "Um método baseado em aprendizado de máquina e análise temporal das informações de atributos relacionados a objetos IOT pode habilitar a integração de sistemas, ao mesmo tempo em que colabora para o gerenciamento da confiança e diferencia nós atacantes de defeituosos".

1.5 Delimitação da Tese

O presente trabalho situa-se no contexto de integração semântica de recursos para a IoT e segurança, entretanto, não é escopo desta tese:

- propor ou formalizar métodos de aprendizado de máquina, visto que é o objetivo deste é a ampla integração com infraestruturas existentes e compatibilidade, por conseguinte, a utilização de metodologias conhecidas para facilitar este objetivo;
- a comunicação (troca de dados) entre dispositivos não é discutido neste trabalho, visto que a solução proposta atua no escopo de integração semântica e independência de arquiteturas;

- outros requisitos de segurança, como a autenticação e confidencialidade não são tratados. Este trabalho foca em tratar os riscos relacionados à confiança das informações utilizadas pelos recursos e proteção de ataques do tipo *On-Off*. Outros tipos de ataques relacionados à confiança da IoT, como ataques oportunista à serviços oportunistas, *ballot-stuffing*, *bad-mouthing* e ataques de auto-promoção não são investigados nesse trabalho.

1.6 Objetivos da Tese

O objetivo principal deste trabalho é desenvolver um método para IoT que possibilite o reconhecimento de novos recursos conectados, integrando serviços heterogêneos, pelo reconhecimento semântico automatizado de atributos, ao mesmo tempo que promove segurança, com a detecção de nós atacantes e defeituosos, colaborando com o gerenciamento da confiança.

1.6.1 Objetivos Específicos

Para alcançar os objetivos propostos, as seguintes atividades são consideradas:

- compreender os modelos utilizados pelas arquiteturas de IoT atuais disponíveis, ao que concerne a integração semântica, suas limitações e soluções;
- identificar tecnologias apropriadas para o desenvolvimento de uma arquitetura inteligente e auto-configurável, capaz de reconhecer semanticamente recursos e realizar anotações em ontologias;
- propor uma método para identificar objetos IoT, para o desenvolvimento de novos serviços integrados semanticamente, bem como garantir a confiança das informações;
- realizar a implementação e realização de provas de conceito, utilizando dados reais obtidos de várias *SmartCities*;
- avaliar os resultados buscando evidenciar a eficiência e a eficácia da solução desenhada, calculando as métricas de precisão e *recall*.

1.7 Organização do Texto

O presente documento está estruturado da seguinte forma: o capítulo 2 contém os trabalhos relacionados ao estado da arte sobre dos métodos de reconhecimento semântico de recursos IoT como também ao gerenciamento da confiança e detecção de ataques;

No capítulo 3 é apresentado o método que realiza o reconhecimento semântico de recursos IoT e colabora para o gerenciamento da confiança, avaliando os metadados dos objetos e identificando ataques. Inicia-se com as considerações preliminares sobre o método, seguindo com suas características de implantação, baseadas em aprendizado de máquina e no conceito de janela elástica dinâmica.

O capítulo 4 contém as descrições das metodologias utilizadas para validar o método proposto. As capacidades de reconhecimento semântico de novos recursos IoT e identificação de atacantes foram verificadas utilizando dados reais provenientes de *SmartCities* e também simulações. Da mesma forma, foram realizados ensaios para aferir o desempenho do método com outros estudos.

Os resultados da validação do método proposto são apresentados no capítulo 5. Os resultados para as capacidades do método em reconhecer semanticamente os atributos dos objetos e identificar ataques são mostrados separadamente. O capítulo inicia-se com a os resultados referentes a identificação semântica de recursos e finaliza com o detalhamento da performance na identificação de atacantes e nós defeituosos. As ameaças à validação deste trabalho também estão enumeradas neste capítulo.

E por último, o capítulo 6 contém as discussões finais e conclusões deste trabalho.

Capítulo 2

Trabalhos Relacionados

Nesta seção é descrito o estado da arte relacionado aos métodos de reconhecimento semântico de recursos IoT como também ao gerenciamento da confiança e detecção de ataques. Os trabalhos foram selecionados através de uma profunda revisão sistemática na bibliografia recentemente publicada e buscas nas bases indexadoras. Os estudos foram avaliados considerando o problema enfrentado nesta pesquisa. A primeira seção detalha o método de revisão sistemática utilizado. A análise dos trabalhos selecionados acerca da integração semântica e gerenciamento da confiança encontram-se me seções específicas.

2.1 Revisão Sistemática

A revisão sistemática é uma metodologia de pesquisa estruturada, baseada em evidências, que tem como objetivo identificar o estado da arte, a prática e as lacunas relevantes em um determinado tópico em publicações, como fundamentos para futuras pesquisas [40]. O método de revisão sistemática proposto por Webster e Watson [41] estabelece um processo de três etapas para o levantamento: (a) objetivos e definição de questões de pesquisa; (B) seleção de palavras-chave e consultas de bancos de dados; e (c) identificação e análise de artigos relevantes. Esta análise baseia-se em métodos qualitativos relacionados à contribuição para o tema, conceitos-chave, proposições consistentes e resultados concretos.

Neste Tese, no entanto, utilizamos o método de revisão sistemática conhecido como Snowballing [42], como uma evolução ao método proposto por Webster e Watson [41] e Kitchenham [40].

As citações feitas por estudos selecionados (*backward*), bem como as citações para eles (*forward*) são levados em consideração para análises, aumentando o escopo e precisão de seleção.

2.1.1 Questões de Pesquisa

Para validar a revisão sistemática, o método exige a elaboração de questões de pesquisa que serão usadas como referência para a geração de palavras-chave para a busca [40] [42], com a finalidade de identificar os estudos primários. Duas questões de pesquisa relevantes foram propostas:

QP1 Quais métodos são apropriados para a integração semântica de arquiteturas heterogêneas em IoT?

QP2 Quais métodos de integração semântica levam em consideração requisitos de confiança?

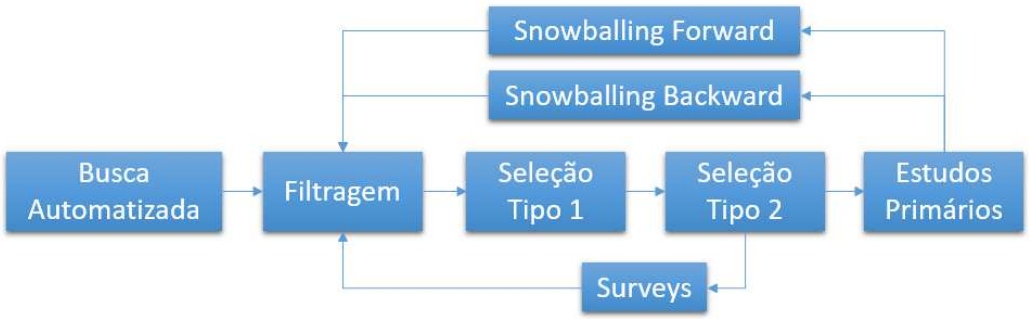
O objetivo dessas questões de pesquisa era entender os métodos de integração semântica em IoT e suas aplicações em IoT, ao mesmo tempo em que se avaliados requisitos de segurança relacionados a confiança (*Trust*). Esses dois fatores são importantes para o desenvolvimento da IoT, dado o crescente número de objetos heterogêneos atuando em diferentes contextos para gerar aplicações cooperativas.

2.1.2 Método de Busca

O processo de *Snowballing* utilizado neste trabalho está ilustrado na Figura 2.1. A partir de questões de pesquisa determinadas, uma *string* de busca 2.1 foi desenvolvida com palavras-chave, abreviaturas e sinônimos, conforme os procedimentos propostos por Kitchenham [40], para ser usada na busca automatizada nas principais bases de dados de indexação de publicações científicas (IEEEXPlore, ACM Digital Biblioteca, ScienceDirect.com, Scopus, ISI Web of Science e Google Scholar). Os resultados (Tabela 2.2) seguiram para a fase de filtragem, onde estudos indexados por mais de uma fonte foram removidos. Os demais estudos foram submetidos às fases de seleção tipo 1 e 2. Na fase tipo 1, os critérios de seleção (descritos na subseção 2.1.3) foram aplicados em títulos, resumos e palavras-chave. Na fase tipo 2, os mesmos critérios foram aplicados para leitura completa do artigo.

Caso os critérios fossem atendidos, o estudo é selecionado e submetido à fase *Snowballing*, cujas referências foram verificadas (*backward*), além de citações de outros estudos (*forward*).

Figura 2.1: Processo de *Snowballing* usado nesta Tese.



Fonte: Elaborada pelo autor.

Tabela 2.1: *String* de busca usada para a revisão sistemática.

("trust management"OR "trust attacks"OR "trust security") AND ("internet of things"OR "iot"OR "machine to machine"OR "machine-to-machine" OR "m2m") AND ("semantic interoperability"OR "semantic interworking" OR "device abstraction")

Fonte: Elaborada pelo autor.

Citações para outras obras foram localizadas utilizando mecanismo de busca do Google Scholar[1]. As listas de estudos gerados pelo *Snowballing* foram re-avaliadas pelos critérios de seleção até que nenhum novo trabalho fosse selecionado.

Os artigos do tipo *Surveys* e outras revisões sistemáticas são a base para o desenvolvimento de vários tipos de pesquisa em diferentes contextos. Por esta razão, eles geralmente recebem uma grande quantidade de citações (algumas com mais de 2.000), que inviabilizam a fase de *Snowballing*. Para este tipo de publicação específica, aplicamos apenas a fase *backward*, referente às referências ao assunto desta pesquisa.

A *string* exibida na Tabela 2.1 foi aplicada nas principais bases de dados do índice de publicações científicas na primeira semana de setembro de 2017, resultando em uma base inicial de 286 estudos, detalhada na tabela 2.2.

Os estudos iniciais selecionados foram submetidos ao processo *Snowballing* (Figura 2.1) e após quatro iterações detalhadas na tabela 2.3, um total de 48 estudos atenderam a todos os critérios de seleção e questões de pesquisa.

[1] https://scholar.google.com/

Tabela 2.2: Quantidade de estudos identificados por base de dados.

Automated Search	Qtd.
IEEEXPlore	29
ACM Digital Library	1
ScienceDirect.com	18
Scopus	28
ISI Web Of Science	0
Google Scholar	279
Total	355
Duplicados	69
Selecionados	286

Fonte: Elaborada pelo autor.

Tabela 2.3: Estudos selecionados em cada etapa do *Snowballing*.

	Título/Resumo Palavras-chave	Backward	Forward	Leitura Completa	Selecionado
Base inicial	286			77	9
Surveys	33			12	8
SnowBalling-i1		29	49	78	12
SnowBalling-i2		15	12	27	9
SnowBalling-i3		18	29	47	10
SnowBalling-i4		8	2	10	0
Total	319	70	92	251	48

Fonte: Elaborada pelo autor.

2.1.3 Critérios de Seleção

Os critérios de seleção foram utilizados para orientar a seleção dos estudos. Com base em questões de pesquisa, o objetivo foi selecionar as publicações relevantes e excluir aquelas que não possuem contribuição direta ao tema. Os seguintes critérios de seleção e exclusão dos trabalhos foram considerados:

CS1 O estudo apresenta cenários de aplicação, desafios e questões em aberto para o contexto semântico ou gerenciamento da confiança em IoT;

CS2 O estudo apresenta estratégias para integração semântica ou gerenciamento da confiança para IoT, como arquiteturas, métodos ou ferramentas;

CS2 O estudo apresenta alguma abordagem para recursos restritos.

CE1 O estudo não está relacionado à IoT;

CE2 O estudo não está relacionado ao cotexto semântico;

CE3 O estudo é uma versão inicial do um mesmo assunto de uma versão mais detalhada;

CE3 O estudo não possui resumo ou texto completo disponível;

CE3 O estudo é um índice, resumo ou editorial;

CE3 O estudo não foi escrito em inglês, a língua mais comum na comunicação científica.

Na revisão sistemática, um estudo é considerado relevante se satisfizer pelo menos um critério de seleção e não atender a nenhum critério de exclusão.

2.2 Métodos Relacionados ao Reconhecimento Semântico para IoT

O reconhecimento semântico pode colaborar com atividades de integração de recursos de IoT. pode colaborar de modo eficaz em tarefas de integração. Seu objetivo é compatibilizar recursos, aplicações e usuários, novos ou provenientes de sistemas legados e heterogêneos em um contexto

cooperativo. Muitas metodologias de reconhecimento e anotação semântica são utilizadas em IoT, porém com limitações relacionadas ao uso de ferramentas manuais ou semi-automatizadas, que necessitam de conhecimento e trabalho dos administradores de sistemas, arquiteturas centralizadas ou fortemente acopladas [30].

Cassar et. al. [43] desenvolveram uma arquitetura de descoberta de serviços escalável, utilizando técnicas de indexação geoespacial e semântica relacionadas aos serviços disponibilizados pelos sensores. Os recursos IoT e serviços são descritos através de ontologias. Mecanismos de clusterização e baseados em Análise Semântica Probabilística Latente (Probabilistic Latent Semantic Analysis - PLSA) e Atribuição latente de Dirichlet (Latent Dirichlet Allocation - LDA) são utilizados para indexar os recursos IoT e atender buscas semânticas. Este estudo, no entanto, não leva em consideração os atributos de confiança dos recursos envolvidos e não foi verificado quanto a métricas de precisão.

Perera [44] defende que a busca e seleção de sensores devem ser feitas baseadas nas prioridades dos usuários. Seu método considera as características de sensores, como confiabilidade, precisão e duração da bateria. Técnicas de consulta semântica e de definições quantitativo são usadas em conjunto com a comparação de distância euclidiana ponderada entre o índice dos sensores e de suas classificações. Seu modelo foi avaliado apenas em termos de consumo de recursos e tempo de resposta, não evidenciando resultados quando o número de sensores aumenta. Foi necessário remover sensores posicionados distantes do sensor ideal definido pelo usuário para reduzir o número de sensores indexados. Seu método pressupõe que dados de recursos são estáticos e não são suscetíveis a mudanças freqüentes.

Le pouch [45] propôs um método chamado Linked Stream Middleware (LSM), que conecta os dados detectados e a Web semântica. Sua abordagem utiliza conectores (*wrappers*) para coleta e publicação de dados em tempo real, uma interface web para anotação e visualização de dados e um terminal para consultar dados de vinculados transmitidos no fluxo. A anotação de atributos manuais é impraticável nas implantações dinâmicas de grandes sensores.

De et.al. [19] propuseram um *broker* (centralizador) para lidar com a descoberta e mediação semântica para a IoT, semelhante a plataformas tais como Xively[2] ou ThingsSpeak [14] que permite que diferentes fontes de dados estejam conectadas a ele. Essas abordagens exigem que os usuários selecionem e configure manualmente sensores. As arquiteturas baseadas em brokers tam-

[2]https://www.xively.com/

bém são são vulneráveis como um ponto único de falha e complexas para suportar a descoberta entre recursos (M2M).

A IETF (Internet Task Force) defende o Core Resource Directory [46] para a padronização de um diretório de recursos centralizado, que registra os End Points (entidade provedora de dados) em um repositório hierárquico de links, disponibilizando para outros sistemas e serviços, atributos dos objetos como nomes, portas e outras informações relevantes. O projeto SmartSantander [47] de iniciativa da Future Internet Research and Experimentation da European Commissionde, desenvolvida na cidade de Santander na Espanha, possui o objetivo de se tornar um TestBed em escala real (cobertura de uma cidade inteira), executando aplicações e serviços de típicos de IoT e também usa o modelo de diretório de recursos para facilitar a localização de objetos e uma série de subsistemas de gerenciamento, como o *IoTResourceManager* e *PSensRegistrationManager* que alimentam este diretório, como também o *TRConfigurator* e *USNConfigurator* que auxiliam a administração da Testbed. No nível de hardware, cada dispositivo possui separadamente recursos para sua finalidade específica e para o controle da Testebed. Além de utilizar elementos centralizadores, outra limitação da arquitetura proposta é a necessidade de se utilizar dispositivos com boa capacidade de processamento, não amplamente disponíveis em IoT, para executar os serviços de controle da Testbed e o próprio serviço.

Soto et. al. [48] desenharam um modelo federalizado para a troca de informações entre arquiteturas IoT. Sua implementação baseia-se na disponibilização de uma PI (Plataform Instance) padronizada para que sensores possam se conectar e disponibilizar os dados para uma entidade central, que abstrai e gerencializa a federação. Apesar de utilizar protocolos abertos, este modelo proposto é um aprimoramento de um barramento SOA, através do conceito de abstração de dados. O conceito de SOA também é utilizado por outras abordagens. Como o Semantic Middleware desenvolvido por Kiljander et. al. [49] que padroniza os objetos IoT como Webservices e o Mobile Digicovery proposto por Jara et. al. [50] que descreve um arcabouço que possibilita aos usuários registrarem seus próprios sensores em uma infraestrutura comum. Cirani et. al. [51] propuseram uma arquitetura escalável e alto-configurável baseado nos conceitos de peer-to-peer. A limitação da adaptação do conceito de SOA para IoT ocorre pela necessidade de descrição e conhecimento das APIs de integração, como também as tarefas administrativas do barramento.

Os arquiteturas orientadas a serviço podem ser beneficiadas pelo uso de reconhecimento semântico, sendo utilizado por vários pesquisadores para a identificação de objetos, como Hachem

et. al. [52] que descreve um *middleware* orientado a serviço que utiliza um conjunto de ontologias mapeadas para descrever as funcionalidades dos dispositivos. Wang et al. [7] propôs uma completa descrição de ontologia para representar os objetos de IoT no contexto do conhecimento, que pode ser usado para descoberta de serviços e composições dinâmicas. Apesar de detalhadas, os estudos não definem meios para mapeamento para outras ontologias, apenas que a mesma seja compartilhada entre as arquiteturas, o que não é possível em implementações em nível mundial.

As abordagens para o Relacionados ao Contexto Semântico para IoT podem ser agrupadas em duas grandes categorias: Tipos e métodos de anotação. Com relação aos tipos, as soluções podem ser enquadradas ainda em quatro sub-categorias: *Frameworks*, *Gateways*, *Middlewares* e Ferramentas. Os modelos de arcabouços são arquiteturas que visam integrar semanticamente os recursos IoT descrevendo a interação dos elementos dentro de camadas lógicas. Modelos de *Gateways* ou *Brokers*, promovem a integração através de entidades centralizadas, realizando tarefas de tradução, combinação ou anotações em ontologias. O uso de *middleware*, com a finalidade de integração, demandam a instalação em dispositivos com alguma capacidade de processamento e meios de interação com outros objetos. A utilização de ferramentas gráficas de anotação é a maneira mais simples de se realizar o alinhamento de ontologias, mas com a necessidade de um conhecimento prévio dos usuários e inviabilidade para grandes implementações.

Os métodos mais comuns de alinhamento de ontologias são baseados combinação (*Matching*), onde os pesquisadores desenvolvem algorítimos que realizam comparações, identificando semelhanças e diferenças. Métodos baseados em aprendizado de máquina, como o processamento de linguagem natural, lógica fuzzy ou Naive-Bayes também são utilizados com performance superior aos métodos de combinação. A abordagem mais simples é a disponibilização de ferramentas de edição, com foco em facilitar aos usuários a realização de anotações e integrações entre as ontologias. É possível também arranjos híbridos com dois ou mais métodos.

A ferramenta Sense2Web proposta por Barnaghi et. al. [53] tem a finalidade de publicar manualmente dados de sensores, conectando-os (link) a recursos semanticamente anotados, associando ontologias de dados heterogêneas com as saídas de dados sensores físicos, assim como a ferramenta Visual Ontology (VisOntology) de Khriyenko et. al. [11] que utiliza uma abordagem gráfica baseada em ícones que auxiliam usuários não especialistas a realizarem o mapeamento e anotações semânticas em ontologias. O ubiquitous Knowledge Base (u-KB) desenvolvido por Ruta et. al. [15] é um arcabouço que utiliza informações únicas, semanticamente anotadas e

uniformemente disseminadas entre recursos, sem um a necessidade de um controlador central. Sua abordagem é focada em prover disseminação e busca semântica de recursos, abstraindo os meios de conexão entre os recursos como também os meios de se descobrir novos atributos para enriquecer as anotações.

A ferramenta gráfica de Heyvaert et. al. [22] auxilia o mapeamento semântico de ontologias de múltiplas fontes de dados. O painel de modelagem abstrai as linguagens de mapeamento da sintaxe através da visualização de grafos. As definições de entrada são interpretadas em declarações em *rdf mapping language* (RML) , suportando diversas origens de dados. Essa múltiplas fontes são visualizadas por meio de "tabs" no painel de entrada. Abordagens manuais no entanto, necessitam de uma de conhecimento prévio dos usuários e são inviáveis em implementações em larga escala.

O Semantic Smart Gateway (IoT-SG) proposto por Kotis et. al. [54] é um gateway IoT inteligente, que executa de forma centralizada e autônoma a tradução em tempo real de descrições de dados (definições ontológicas) para dados padronizados e anotados, acessíveis por outros sistemas. Um componente de aprendizado em tempo real é utilizado para realizar a descrição semântica de recursos que não possuem definições ontológicas ou apenas simples metadados. De outra forma, o arcabouço desenvolvido por Chen et. al. [55] implementa o modelo de SenaaS (Sensing as a Service, análogo a serviços de Cloud, como PaaS, IaaS, etc.) unindo os provedores de dados com os consumidores através de um Service Broker.

O IoT Hub desenvolvido por Blackstock et. al. [18] é baseado em quatro componentes: o IoT Core, o IoT Model, IoT Hub e o IoT Profiles. O IoT Core define os hubs que disponibilizarão os objetos (things) e os metadados utilizando arquitetura web e RESTful web services; O IoT Model define os modelos requeridos para o entendimento comum entre os objetos e os dados associados em um *hub*, de forma a facilitar o desenvolvimento de ferramentas de integração, como adaptadores, para assegurar a interoperabilidade. O IoT Hub, por sua vez, registra e implementa representações de objetos, URLs, esquemas de descrição, catálogos de buscas e dados. Uma característica necessária a arquitetura é a necessidade de suporte a mecanismos de segurança ao acesso aos recursos e autenticação de objetos e dados. Finalmente o IoT Profiles estabelece os meios de interoperabilidade semântica entre os recursos semelhantes de diferentes *hubs*.

O arcabouço desenhado por Amir et. al. [13] por sua vez, realiza a anotação semântica por um método de perfis sistematizado em três componentes: um motor responsável por anotações

semânticas de busca em recursos, uma aplicação web que publica as funcionalidades através de uma *Representational State Transfer* (RESTful) API e um servidor que possui a finalidade de capturar, publicar e monitorar ativos em tempo real. Como meio para garantir a interoperabilidade entre várias redes de sensores, o arcabouço descreve uma representação virtual para cada ativo e um modelo orientado a serviços e interação semântica. Para seu funcionamento, no entanto, faz-se necessário o conhecimento das APIs dos recursos que serão conectados. Implementações baseadas em *brokers* possuem a limitação de serem um ponto único de falha, o que pode indisponibilizar todo o serviço, bem como necessitam de intensiva atividade de gerenciamento.

Desenhos de arquiteturas com foco na criação de padrões, em um estado inicial de desenvolvimento de tecnologias, são vistas como restrições por fabricantes e pesquisadores, pelo risco de dificultar a inovação e a incompatibilidade com outras iniciativas. Vários estudos propõem padronizações, como a arquitetura Plug & Play de Bröring et. al. [56] que possui o objetivo de promover a comunicação instantânea, interpretando dados crus (raw data) de forma a realizar a combinação funcional entre sensores e modelos de serviços específicos através de um barramento de sensores (Sensor Bus) que conecta os elementos (a) administrador do canal que realiza a conexão entre os sensores e serviços e o (b) mediador que realiza a criação de conceitos e combinação semântica.

O ISIS (integrated M2M Platform) desenvolvido por Song et. al.[10] é uma plataforma M2M orientada as necessidades de dois projetos europeus de IoT, o SENSEI e o CAMPUS21 para a realização interoperabilidade semântica que utiliza uma camada de arquitetura identificada como Data&Actuation Abstraction Layer (DAAL). O DAAL possui a função de realizar a troca de informações entre dispositivos M2M conetados através de *wrappers* (encapsuladores), independente da tecnologia de conexão utilizada pelos agentes inteligentes, mapeando os dados dos recursos em atributos conhecidos.

Outro conceito emergente aplicado a IoT é o *Fog Computing*, que define uma plataforma de processamento, armazenagem e conexão entre objetos e serviços de Cloud, em um contexto heterogêneo e regionalizado, de baixa latência, para a mobilidade e serviços de *streaming* e aplicações em tempo real [57].

Sehgal et. al. [58] definiram um arcabouço de segurança para *Fog Computing* baseados em regras do tipo Se “A” faça “B”, armazenadas em repositórios chamados de Banco de Dados de Segurança Regionais, de forma a ficar mais próximos dos objetos, atendendo sua capacidade compu-

tacional e elevando a segurança na medida que se avança nas camadas em direção a nuvem e por conseguinte, as capacidades computacionais dos objetos envolvidos. Sua abordagem, no entanto, não define como novos recursos serão mapeados como também a necessidade de intervenção de um administrador para a manutenção das regras.

Lee et. al. [59] propuseram uma arquitetura de *Fog Computing* baseado em *gateways* inteligentes que operam de forma semelhantes as redes definidas por Software. Este conceito possui limitações inerentes a esta abordagem, como a necessidade de um controlador central e diferentes hierarquias de gerenciamento de recursos e segurança. Giordano et. al [60] por sua vez descreveu uma arquitetura chamada *Rainbow Platform*, que utiliza o conceito de Agentes Inteligentes agregado a alguns tipos de dispositivos, que através da análise das leituras dos sensores em um método estatístico (médias), agrupam os objetos semelhantes para criar a uma arquitetura de *Fog Computing*. O arcabouço proposto ainda não foi validado em uma implementação real como também não é capaz de detectar outras tarefas, como o consumo de energia.

A alternativa de construir e evoluir uma única ontologia a ser compartilhada entre as arquiteturas é defendida por vários estudos como meio de integração. No entanto, esta abordagem normalmente gera grandes ontologias que se provaram ser impraticáveis em implementações reais, necessitando a criação de segmentações ou ontologias paralelas alinhadas. Como exemplos, temos a solução CANThings de Davoudpour et. al. [20] que tem por foco promover o alinhamento de ontologias baseando-se na criação de uma única ontologia comum, através da combinação semântica automatizada e informações de contexto. O mecanismo de mapeamento baseado em ontologias OWL sendo capaz de analisar e estimar objetos, classes, atributos, relações, funções, regras de restrições e axiomas que podem se gerados para conectar quatro componentes relvantes de IoT: Coisas, Pessoas, Lugares e Dados.

O arcabouço Machine-to-machine Measurement (M3) desenvolvido por Gyrard et. al. [61] auxilia usuários finais e desenvolvedores, através da disponibilização de uma abstração de alto nível dos dados de sensores, para interpretá-los e anotá-los semanticamente, gerando aplicações compatíveis. Esta abstração é realizada pela a criação, armazenamento e compartilhamento de *templates* entendidos por aplicações heterogêneas.

O modelo proposto por Xu et. al.[21] é baseado na construção de uma nova ontologia que integra todos os recursos IoT, partindo de uma ontologia básica, no caso a SSN (Semantic Sen-

sor Network Ontology [3]) e continuamente incrementá-la com novas anotações, originárias das definições extraídas de dados puros (raw data) através de algorítimos de processamento de linguagem natural. Cada conceito novo que possibilita a geração de uma nova tupla de definições na ontologia é recalculado de forma a evitar similaridades e repetições.

Em paralelo, o modelo semântico de inclusão automatizada defendido por Hur et. al. [17] integra várias plataformas públicas de IoT disponíveis no mercado [4] [5] [6], através de um processador de descrição semântica (Semantic Service Description (SSD)), que atua na consistência semântica entre as plataformas, através da criação de uma ontologia unificada que descreve a representação do conhecimento, compartilhando entre elas.

Em outra abordagem, a arquitetura desenvolvida por Xiao et. al. [62] realiza o mapeamento semântico através da estratégia de separação de camadas de apresentação dos dispositivos em três modos: *real device*, *common device*, e *virtual device*. A ideia principal do arcabouço é resolver o problema de interoperabilidade semântica mapeando dispositivos reais em dispositivos comuns e posteriormente em dispositivos virtuais.

Outros estudos procuram atender o dinamismo da IoT, desenhando arquiteturas mais flexíveis, muitas vezes apoiados em aprendizado de máquina. O SPITFIRE de Pfisterer et. al. [16] é uma abordagem semi-automatizada de anotação semântica baseada no princípio de que diferentes sensores do mesmo tipo tendem a exercer um comportamento semelhante no mesmo contexto. A solução realiza o cálculo de inferências utilizando um vocabulário integrado de sensores e objetos, entidades semânticas (como uma abstração de alto-nível de estados possíveis dos sensores), geração semi-automatizada de descrições dos sensores em uma busca baseada em estados atuais (ligado ou desligado). Este método ainda necessita de intervenção humana para incluir os sensores, outros atributos, validação das detecções e padronização dos objetos.

O *middleware* apresentado por Ming et. al. [23] executa a anotação semântica em arquiteturas fracamente acopladas, utilizando aprendizado de máquina e ontologias, onde os dados de entrada são analisados em três parâmetros: Demanda dos usuários, *web-ervices* e dados do recurso. As ontologias de domínio são geralmente criadas por especialistas como uma coleção não refinada de classificações (*coarse-grained*), ao mesmo tempo que os *web-services* e dados de recursos

[3] https://www.w3.org/2005/Incubator/ssn/ssnx/ssn
[4] https://evrythng.com/
[5] https://www.carriots.com/
[6] https://www.compose.io/

podem pertencer a vários domínios de classificação. Desde que a ontologia de domínio possua referências nas três abordagens, é possível a aplicação de mecanismos de classificação de texto para a extração de uma outra ontologia de domínio mais refinada, via um classificador Naive Bayes. Diferentemente do SPITFIRE [16], esta abordagem utiliza apenas os atributos léxicos dos recursos, ignorando os tipos de dados utilizados, podendo gerar o reconhecimento e registro equivocado de dispositivos ou *outliers*.

Malik et. al. [63] desenvolveram um *middleware* baseado em multi agentes trabalhando em dois estágios de transformação de dados, mapeando um modelo particular de informação em uma estrutura relativamente sancionada do mesmo grupo de representação da informação, ou seja, traduz múltiplas fontes dados em formatos como o csv, XML ou bases sociais em uma única definição RDF. Os agentes trocam informações entre si para agruparem múltiplas fontes de dados, pela a execução de algorítimos de combinação (*matching mechanism*) de *templates* XML, com a representação do recurso e também por reconhecimento de padrões de dados em saídas temporais (time series). A restrição desse método esta relacionado na dependência de mais de um agente ativo para a realização do serviço, além de não suportar outras infraestruturas que não executem o *middleware*.

Uma característica importante das soluções de integração semântica são sua compatibilidade com os dados de entradas e saídas. Foram observados nos estudos avaliados que a maioria são compatíveis com vários padrões de dados de entrada, como csv, xml, texto puro, entre outros. Entretanto, é verificado uma tendência entre os pesquisadores na utilização do RDF como padrão para os dados de saída, como linguagem mais adequada para anotação e integração de ontologias.

2.3 Métodos Relacionados ao Gerenciamento da Confiança para IoT

Os recursos e os dados da IoT correm o risco de ataques a confiança (Trust). Os ataques a confiança podem ser auto-promoção, ataques via difamação, ataques de serviço oportunista e ataques On-Off. O gerenciamento e as técnicas para confiança ajudam a proteger as implementações de IoT e devem considerar a escalabilidade, mobilidade do nó, relações sociais e experiências de uso do serviço [34].

A avaliação de confiança é uma parte essencial de um esquema de gerenciamento de confiança. Os métodos de avaliação de confiança podem ser classificados em confiança direta e confiança indireta. A confiança direta refere-se a métodos que são capazes de inferir pontuação de confiança devido a observações diretas de dados. A confiança indireta usa a reputação e recomendação de outros pares. As pontuações de confiança podem ser gravadas em um nó central conhecido ou em um terceiro autorizado. Em um modelo descentralizado de avaliação de confiança, um nó calcula um valor de confiança para cada interação entre nós [34].

O esquema proposto por Chen et. al. [31] usa a relação social e o status energético dos nós. Os tempos necessários para as transações são considerados para melhorias dinâmicas de desempenho. O seu esquema pressupõe que os nós com menos energia terão um valor de confiabilidade restrito e sua oportunidade de cooperação será diminuída para tornar a rede estável. Esta metodologia, no entanto, não é aplicada a todos os cenários IoT. Os nós confiáveis capazes de fornecer acesso aleatório e dados ao sistema podem ser erroneamente identificados como nós não confiáveis. Em outra abordagem [64], os autores usaram um protocolo social de gerenciamento de confiança adaptativo, com o objetivo de escolher as melhores configurações de parâmetros de confiança em resposta à alteração das condições sociais do IoT. O método de pesquisa de tabela proposto aplica os dinamicamente os resultados da análise para avaliar a confiança. Métodos de abordagem social demandam tempo para atingir pontuações de confiança uteis para que um nó seja selecionado.

Nitti et. al. [33] definiram dois modelos iniciais de gerenciamento de confiabilidade. O modelo subjetivo, que cada nó calcula a confiabilidade de seus vizinhos pela sua própria base de experiência e opinião de outros vizinhos e o modelo objetivo, que distribui informações sobre cada nó. O modelo proposto, quando aplicado em cenários P2P, aumentou consideravelmente o tráfego de rede para troca de *feedbacks*.

Saied et. al. [65] propuseram um sistema de reputação para um sistema IoT orientado à serviços, considerando a qualidade do serviço (QoS) e as informações de contexto, como o tipo de serviço e capacidade de nó (e.g. status de energia) como métricas de confiança a ser calculada por uma média ponderada. Ele também usa um gerenciador centralizado para armazenar todos os relatórios de reputação. Namal et. al. [66] considera as métricas diretas de confiança de QoS, como disponibilidade, confiabilidade, irregularidades e capacidade. O método proposto usa uma média ponderada estática das experiências passadas e novos dados de QoS do sensor. Assim como o trabalho de Saied et. al., a maioria das pesquisas sobre a confiança da IoT não considera a

descoberta do serviço. Elas partem do princípio que os dispositivos já estejam conectados e suas relações de confiança foram previamente avaliadas.

A abordagem de gerenciamento distribuído de Mendoza et. al. [67] calcula o valor de confiança localmente pelos nós. O valor da confiança é baseado em observações diretas, através da disponibilidade do serviço do nó relacionado. Este esquema acarreta um alto o consumo de tempo e recursos. Foram necessários 120 minutos para preencher uma tabela local com apenas dois nós. Outra desvantagem desta abordagem é que não considera o nível de confiança inicial de um vizinho.

O RealAlert é o esquema de detecção seguro e confiável baseado em políticas proposto em por Li et.al. [68]. Neste esquema, os atributos de confiabilidade de dados e nós IoT são avaliados usando dados anômalos e informações contextuais que representam o ambiente sob o qual os dados foram obtidos. As regras da política são definidas para especificar como avaliar a confiabilidade em diferentes situações. Novos dispositivos ou novas observações normais podem ser considerados um nó suspeito devido uma política desatualizada.

O modelo quantitativo de valor de confiança baseado em atributos de decisão multidimensionais de Yu et. al. [69] usa o valor de confiança diretamente monitorado, medido a partir dos aspectos de comunicação da rede. A capacidade de encaminhamento de pacotes, a taxa de repetição, a consistência do conteúdo, atraso e integridade são avaliadas. O modelo adota a teoria D-S para calcular a confiança. Sua desvantagem está relacionada a grande quantidade de dados coletados de vários dispositivos no ambiente IoT. A quantidade de dados aumenta de forma exponencial durante o fluxo contínuo de transmissão, sendo difícil de gerenciar através dos métodos tradicionais de análise de dados de comunicação de rede.

O método de descarga baseado em confiança para comunicações M2M proposto por Boustanifar et. al. [70] adapta a aprendizagem por reforço a construção de um sistema de *feedback*. O nível de confiança de um nó inicializado é atualizado após cada comunicação para permitir que o este avalie mais precisamente novas interações. É focado em como a confiança pode melhorar o consumo de energia e a velocidade de computação dos dispositivos, aprimorando a disponibilidade do sistema. No entanto, este esquema não considera os diferentes serviços fornecidos por um vizinho e a confiabilidade dos dados de confiança coletados de cada nó.

O modelo de segurança adaptativa proposto por Hellaoui et. al. [71] é baseado no método de avaliação de confiança composto por três componentes complementares: experiências, observa-

ções e recomendações. Seu principal objetivo é a redução do consumo de recursos em redes ad hoc. A arquitetura de cluster defendida por BenAbderrahim et. al. em [72] aborda o gerenciamento de confiança para IoT com base na semelhança de interesse de cada cluster. Seu mecanismo de predição usa filtros de Kalman para estimar antecipadamente o valor da confiança.

O estudo de Zhang et. al. [73] usa um esquema de relacionamento de confiança baseado em redes de sensores sem fio em cluster (M2M). Seu modelo de nuvem implementa a conversão entre dados qualitativos e quantitativos de nós sensores (métricas de confiança). Para calcular a confiabilidade dos nós sensores, o método considera os fatores de comunicação, mensagens e energia. Uma atribuição dinâmico de uma nota de peso para cada fator de confiança é usada para detectar ataques. As soluções em nuvem são ameaçadas por problemas relacionados ao fornecedor (protocolos proprietários ou projetos de ponta-a-ponta), potência de dispositivos restritos e métodos incompatíveis para a descoberta de serviço.

O esquema de gerenciamento da confiança definido por Sony et. al. [74] busca discriminar erros temporários de comportamentos maliciosos para detectar e defender contra o ataques OA. Ele utiliza a previsibilidade da confiança, calculada como a relação entre o bom comportamento e o comportamento total no sistema, e uma janela deslizante estática que registra o histórico de comportamento anterior. Este esquema precisa de tempo para calcular todo o comportamento do sistema. Mantendo um registro de comportamento estático, ele não pode acomodar novos atos confiáveis.

Outros estudos descrevem proposições de arquiteturas para tratar problemas de específicos de segurança, como a infraestrutura proposta por Zhou et. al [75], com foco na segurança da distribuição de conteúdo multimídia em IoT, baseado na classificação do tráfego relacionado. O esquema de gerenciamento de chaves é determinado pelo tráfico multimídia, controle do serviço, controle dos usuários e no controle de fluxo, definindo a escalabilidade, pela verificação da quantidade, distribuição ou tráfego gerado pelos recursos. A implementação possui a limitações relacionadas a sincronização de chaves e de ineficiência no gerenciamento de uma grande quantidade de recursos.

No contexto de autenticação, Kothmayr et. al [76] definiram um esquema de duas vias baseado no protocolo padrão Datagram Transport Layer Security (DTLS), utilizando criptografia RSA, protocolo IPv6 e implementado em redes de baixa potência (6LoWPANs). No entanto, o método de criptografia gera um alto *overhead* na rede de comunicação. Lee et. al [77], utilizando um hash

simplificado (para objetos limitados), desenvolveram um sistema leve de trocas de chaves entre usuários, *gateways* e nós de sensores, onde o *gateway* nunca é contactado, com a limitação de não garantir a segurança da fia-a-fim da comunicação.

Turkanovic [78] apresenta um método de autenticação baseado em Curva Criptográfica Elíptica (ECC), que não sobrecarrega processadores limitados, utilizado para atribuir políticas de controle de acesso, gerenciado por uma autoridade central. Seu protocolo no entanto, transmite duas vezes cada cilo de autenticação, aumentando o consumo de baterias dos nós. Para a resoluções de problemas de confidencialidade, Du et. al. [79], propôs um um protocolo leve para o gerenciamento de chaves, baseado no conceito infraestrutura de chaves públicas (PKI) para comunicação segura, não abordando eficientemente, o gerenciamento de de autenticação de dispositivos, ficando vulneráveis a ataques do tipo *man-in-the-middle*.

O conceito de *Trust* (confiança) aplicado a IoT possui um importante papel para o atendimento de requisitos de segurança da informação, pela garantia na utilização de fontes confiáveis no complexo ambiente que coleta dados em ambientes físicos, interações humanas e redes de comunicação heterogêneas para aplicações que necessitam oferecer serviços de forma pervasiva [30].

Bao et. al. [80] definiram uma abordagem dinâmica para o gerenciamento do Trust, desenvolvendo um protocolo que analisa os relacionamentos sociais entre os objetos, utilizando scores para a honestidade, cooperação e interesse da comunidade. Este método se mostrou não preciso, pois recursos com restrições computacionais confiáveis recebem uma baixa nota de confiança por não repassarem informações sobre outros nós da rede.

Bernabe et. al. [81] por sua vez, utilizaram um método chamado TACIoT (*Trust-aware access control mechanism for IoT*) que auxilia dispositivos a realizar autenticações de forma mais segura, utilizando parâmetros de qualidade de serviço, reputação, aspectos de segurança e relacionamentos sociais, computados por um sistema de controle *Fuzzy*, porém desenhando para um aplicação específica de provisão de serviço.

Rafey et. al. [82] desenharam um modelo chamado de CBSTM-IoT (*Context-based Social Trust Model for The Internet of Thing*) que através de interações sociais bi-direcionais e recomendações indiretas melhoram a confiança entre os objetos. Esta abordagem necessita ser evoluída para a identificação de falsos negativos. A maioria dos modelos propostos atualmente utilizam conceitos de redes sociais, nem sempre fáceis de serem observados em implementações M2M

(*Machine-to-Machine*), bem como necessitam definir previamente os atributos avaliados.

Outros estudos também descrevem proposições de arquiteturas para tratar problemas de específicos de segurança, como a infraestrutura proposta por Zhou et. al [75], com foco na segurança da distribuição de conteúdo multimídia em IoT, baseado na classificação do tráfego relacionado. O esquema de gerenciamento de chaves é determinado pelo tráfico multimídia, controle do serviço, controle dos usuários e no controle de fluxo, definindo a escalabilidade, pela verificação da quantidade, distribuição ou tráfego gerado pelos recursos. A implementação possui a limitações relacionadas a sincronização de chaves e de ineficiência no gerenciamento de uma grande quantidade de recursos.

2.4 Considerações do Capítulo

Neste capítulo foi apresentado o estado da arte relacionado aos métodos de reconhecimento semântico de recursos IoT como também ao gerenciamento da confiança e detecção de ataques. O método de *Snowballing* de revisão sistemática permitiu a seleção de 48 trabalhos relevantes ao tema desta pesquisa. De manaira geral, os diversos métodos avaliados alcançam eficiência em focos específicos (serviço ou segurança), mas não de maneira integrada. Além disso, não foi identificado estudos que diferenciam nós atacantes de nós defeituosos em um cenário de ataques do tipo ON-Off.

No próximo capítulo será detalhado o método proposto neste trabalho, que realiza o reconhecimento semântico de recursos IoT e simultaneamente colabora para o gerenciamento da confiança, avaliando os metadados dos objetos e identificando ataques.

Capítulo 3

Método Proposto

Neste capítulo é apresentado o método que realiza o reconhecimento semântico de recursos IoT e identifica ataques à confiança do tipo *On-Off*, pela avaliação dos atributos dos objetos. Inicia-se com as considerações preliminares sobre o método, seguindo com suas características de implantação, baseadas em aprendizado de máquina e no conceito de janela elástica dinâmica.

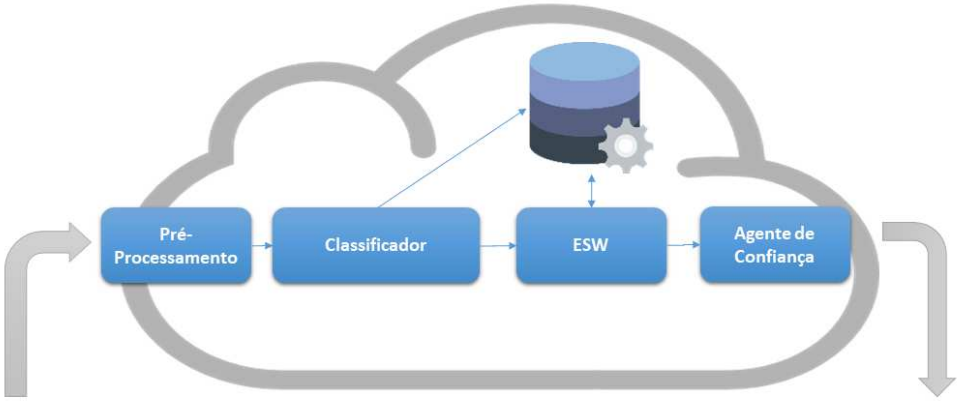
3.1 Um Método para o Gerenciamento da Confiança na Identificação de Recursos e Detecção de Ataques para a Internet das Coisas

O método proposto neste trabalho visa atender requisitos de integração de arquiteturas e objetos de IoT, através do reconhecimento e identificação de atributos semânticos dos recursos, além de tratar os ataques ao gerenciamento da confiança. Esta abordagem é implementada através de um classificador baseado em aprendizado de máquina e um algoritmo de janela elástica dinâmica (JED) utilizado para a análise dos dados em um intervalo de tempo variável. O objetivo principal é auxiliar a integração confiável de sistemas IoT, através do reconhecimento, padronização e anotação semântica dos atributos dos recursos conectados, ao mesmo tempo que colabora para gerenciamento da confiança.

São características de desenho desse método:

1. Não necessita de configuração específica para se integrar com outras arquiteturas ou dispo-

Figura 3.1: Fluxo de meta-dados na nuvem para o método proposto.



Fonte: Elaborada pelo autor.

sitivos, reconhecendo e padronizando atributos de objetos de forma transparente;

2. Não demanda a implementação integral de todos as camadas de IoT (percepção, *gateways* ou *brokers*), podendo funcionar de forma independente, regional e/ou incompleta;

3. Ser compatível com outras infraestruturas, tecnologias e protocolos abertos existentes;

4. Possuir a capacidade de reconhecer novos recursos e padronizar suas definições semânticas, como também detectar ataques ao gerenciamento da confiança, calculando a acurácia da identificação;

5. Atender dispositivos restritos, providos com baixa capacidade de processamento, armazenagem e comunicação.

Este método de reconhecimento de recursos e gerenciamento de confiança foi projetado para ser acessado via arquitetura de nuvem (*Cloud Computing*), através de uma *Application Programming Interface* (API) e protocolo de *Representational State Transfer* (REST). Na Figura 3.1 ilustra-se o fluxo de metadados através do método na nuvem.

O fluxo de informações no método inicia-se a a partir do envio de um metadado relacionado aos atributos acerca de um recurso IoT por um usuário ou outro recurso do sistema, que possui interesse em identificar e padronizar as informações semânticas. Na fase de pré-processamento, as informações são submetidas a um processo de extração específico para tipo de dado relacionado. Os dados numéricos são padronizados removendo a média e a escala para variância unitária,

Figura 3.2: Intervalo esperado de valores confiáveis.

Fonte: Elaborada pelo autor.

enquanto os dados de texto são processados via HashingVectorizer[83]. Esse processo converte texto (*n-grams*) para uma matriz de ocorrências e define um nome para esta seqüência, que será utilizada posteriormente para o mapeamento de índices. Essa abordagem foi selecionada porque não armazena um dicionário de vocabulário na memória e também pode ser usada em *streaming* (implementação parcial).

Os dados pré-processados são submetidos a um classificador de aprendizagem de máquina para identificar sua classe. Os valores do intervalo para o treinamento do classificador, como por exemplo o intervalo de temperatura anual para uma cidade, são assumidos como valores confiáveis (Figura 3.2). Caso contrário, os valores fora de padrão são marcados como *outliers*.

Após a fase de pré-processamento, o classificador confirma se identificou uma classe específica e retorna o valor da função de decisão. A função de decisão é a distância da amostra avaliada ao hiper-plano definido pela função de decisão do modelo, sendo utilizada pelo método para determinar a nota (score) de confiança. O grau de separação alcançado no hiper-plano tem maior valor para os pontos de treinamento mais próximos de qualquer classe (a margem funcional). Um valor de função de decisão alto (ou positivo) corresponde à garantia de identificação da uma determinada classe. Para a realização do treinamento do classificador, podem ser usados qualquer conjunto de dados contendo atributos relevantes para as aplicações como sensores, tipos, IDs, coordenadas GPS, entre outros relacionados aos objetos IoT previamente reconhecidos ou de referência semântica.

A determinação da confiança dos dados de identificação semântica de novos recursos é realizada pela interpretação matemática da função de decisão criada durante o treinamento do classificador. Um classificador de aprendizado de máquina supervisionado é treinado por um conjunto de dados anotados $(D_l = (x_i, \omega_j) \in X \times \Omega; i = 1, 2, ..., m; j = 1, 2, ..., c)$ sendo representado por uma função $f : X \rightarrow \Omega$, que relaciona elementos $x \in X$ a uma classe $\omega \in \Omega$. O conjunto X

é denominado pelo espaço de atributos que contém o conjunto de padrões que f classifica.

Na Figura 3.3 exibe-se uma plotagem de uma função de decisão e seu funcionamento como método de reconhecimento de recursos, determinação da confiança e detecção de *outliers*. A função de decisão f foi obtida pelo treinamento algoritmo do classificador utilizando dados previamente anotados (pontos azuis que representam a base de treinamento). Um classificador adequadamente treinado gera uma função que delimita a maioria das amostras anotadas. Os pontos assinalados em verde representam novos objetos corretamente identificados. Os novos recursos válidos submetidos à função de decisão f treinada, geram resultados dentro ou próximos ao domínio da função. Por outro lado, elementos que se apresentam distantes da função, ou seja, com pouca ou nenhuma relação são marcados como *outliers*.

Vários tipos de algoritmos como os Naive Bayes Gaussianos, Redes Neurais, Random Forest entre outros executam as tarefas de classificação [83] que poderiam ser utilizados para a implementação do método. Entretanto, as funções de decisão geradas pelos algoritimos baseados em Máquinas de vetores de suporte (SVM), em especial a implementação do *kernel* linear, colaboraram de forma significativa para a performance do método (ver Capítulo 5).

As máquinas de vetores de suporte (SVMs) são um conjunto de métodos de aprendizado supervisionados usados para classificação, regressão e detecção de *outliers*. Dentre suas vantagens, destacam-se sua eficiência para tratar espaços dimensionais elevados e casos em que o número de dimensões é maior que o número de amostras. Além de uso eficiente de memória [83].

O LinearSVC é a implementação de SVM para o caso de um *kernel* linear [84]. Dado um vetor de treinamento $x_i \in R^n, i = 1, ..., l$ em duas classes, e um vetor $y \in R^l$, onde $y_i = \{1, -1\}$, um classificador linear gera um vetor de pesos w como um modelo. A função de decisão criada será $\text{sgn}\left(w^T x\right)$.

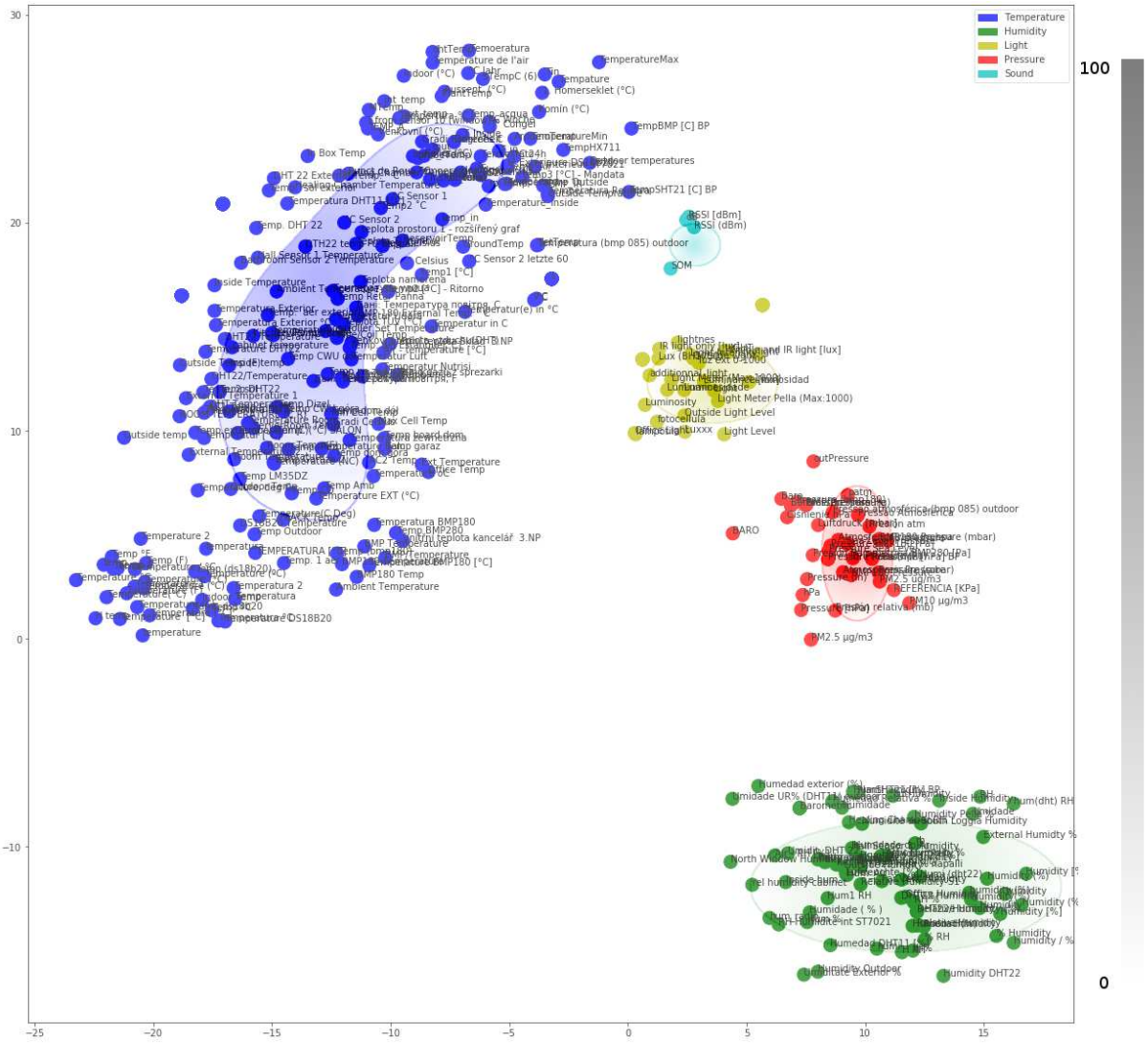
Este classificador linear resolve o problema fundamental da SVM:

$$\min_w \frac{1}{2} w^T w + C \sum_{l}^{i=1} \left(\max\left(0, 1 - y_i w^T x_i\right)\right)^2$$

e seu dual: $\min_\alpha \frac{1}{2}\alpha^T \bar{Q}\alpha - e^T\alpha$. Onde e é o vetor de todos os uns, $Q = Q + D$, D é a matriz diagonal e $Q_i j = y_i y_j x_i^T x_j$.

Uma máquina de vetores de suporte constrói um hiperplano ou um conjunto de hiperplanos

Figura 3.3: Exemplo de uma função de decisão.



Fonte: Elaborada pelo autor.

em um espaço de alta ou infinita dimensão, A separação é alcançada pelo hiperplano que possui a maior distância até os pontos de dados de treinamento mais próximos de qualquer classe (chamada margem funcional), pois em geral quanto maior a margem menor o erro de generalização do classificador.

Com base na observação da distância euclidiana da amostra identificada à função de decisão é possível predizer uma pontuação (score) de confiança. A pontuação de confiança é atribuída pela distância de uma amostra no hiperplano de decisão. A implementação realizada neste trabalho utilizou os resultados da função ($decision_function(X)$) gerados pela biblioteca Scikiy-Learn [83] agregado a um cálculo para a apresentação em forma de percentual. O uso da função de

decisão para o cálculo de confiança é utilizado com sucesso em implementações em várias áreas, como no mercado financeiro [85], filtros de anti-spam [86], sistemas seguros de autenticação utilizando reconhecimento facial [87], entre outros.

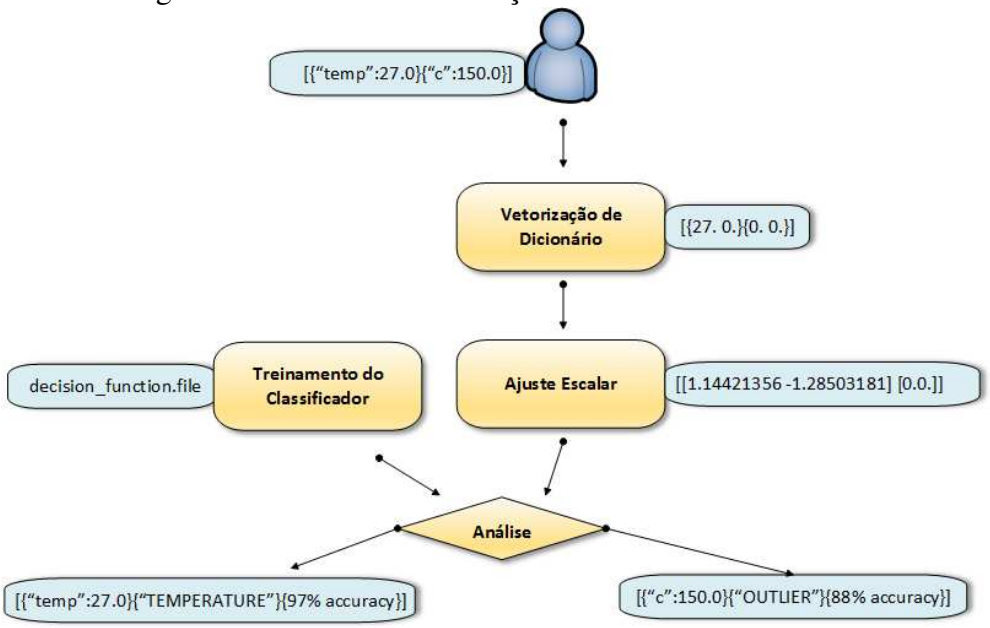
Do mesmo modo, a pontuação obtida do cálculo da distância da função de decisão pode ser também utilizada para a identificação de anomalias. Uma valor baixo pode indicar que o recurso avaliado possui um ou mais atributos fora de padrão, destoantes das amostras de treinamento. Abordagem semelhante foi utilizada por Ban et. al [88] para detecção de anomalias em aplicativos para a plataforma Android e por Wang. et. al. [89] para ameaças às redes definidas por software.

Durante o processo de reconhecimento semântico, os novos recursos a serem identificados são analisados pelo classificador previamente treinado, que implementa a função de decisão. As saídas possíveis dessa análise são (a) o tipo de recurso reconhecido ou se o mesmo é um (b) *outlier* (não detectado). O método proposto também gera uma nota de acurácia, obtida pela distância da amostra em relação à função de decisão. No exemplo ilustrado na Figura 3.4, os recursos "temp=27" e "c=15" foram reconhecidos como um recurso relacionado a temperatura e um *outlier* respectivamente. O primeiro gerou uma nota de acurácia de 97%, mostrando o grau de confiança obtido pelo sistema ("temp" se referindo a temperatura e "22" dentro do espectro possível observado para esse fenômeno), enquanto a detecção do *outlier* não alcançou o mesmo grau de certeza (88%).

A partir dessas duas saídas (recurso e acurácia), é possível inferir a existência de anomalias. Um recurso não detectado com uma alta acurácia pode ser interpretado como um anomalia, pois seus atributos não são compatíveis com os padrões observados na base de treinamento do classificador. Do mesmo modo, recursos com baixa acurácia podem ser indicativos de falsos positivos e negativos. Quando estes fenômenos ocorrem em quantidades elevadas, sugere-se a necessidade de ajuste no classificador. Na Tabela 3.1 podemos visualizar um resumo das interpretações possíveis dos resultados gerados pelo método.

Assim como o classificador, a janela elástica dinâmica (JED), desenvolvida exclusivamente para o método (Figura 3.5), é uma fase importante no fluxo de dados analisados, sendo utilizada para aumentar a confiança da análise em uma abordagem temporal. Sua implementação viabiliza a identificação de nós atacantes para gerenciamento da confiança, como também possibilita apontar nós defeituosos. Um nó que realiza um ataque do tipo OA, envia valores de leitura bons (confiáveis) e ruins (não confiáveis) de forma discricionária. Sistemas que operam em estado

Figura 3.4: Processo de anotação de um recurso avaliado.



Fonte: Elaborada pelo autor.

Tabela 3.1: Interpretação dos resultados de detecção e acurácia para a identificação de anomalias.

Recurso	Acurácia	Interpretação
Não detectado	Alta	Anomalia
Não detectado	Baixa	Falso negativo
Detectado	Alta	Novo recurso
Detectado	Baixa	Falso positivo

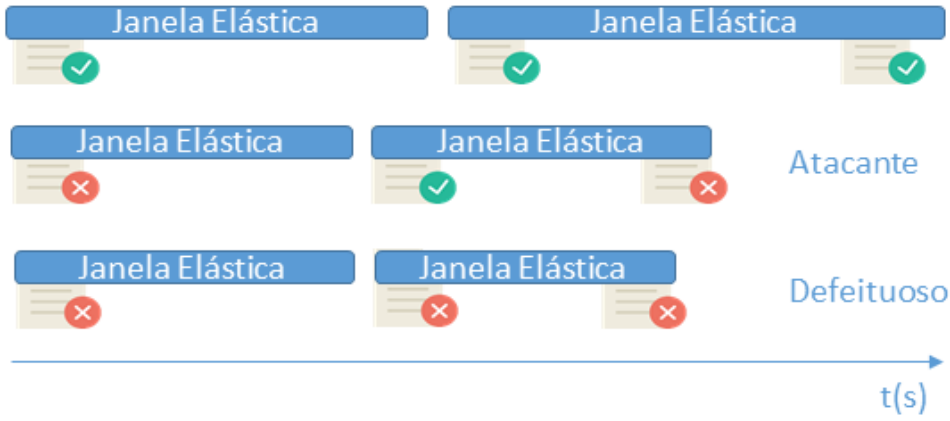
Fonte: Elaborada pelo autor.

considerado normal, esperam apenas bons valores ao longo do tempo. Quando o classificador identifica uma classe e retorna um baixo valor para a função decisão, o método sinaliza dúvidas quanto à confiança e o recurso avaliado passa ser testado novamente em um período de tempo maior.

Em cada interação, a JED é agregada pelos valores da função de decisão. Os baixos valores da função de decisão levam a janelas elásticas maiores e permitem que o método analise a variância em comportamentos do nó por mais tempo.

A tamanho da JED é calculada utilizando os valores de função de decisão. Os baixos valores da função de decisão (valores negativos) aumentam o tamanho das janelas de tempo de avaliação e permitem que o método analise a variância de confiança em período maior. O tamanho da janela

Figura 3.5: A janela elástica dinâmica (JED).



Fonte: Elaborada pelo autor.

elástica dinâmica é calculado pela equação 3.1.

$$JED = JED - FuncaoDeDecisao \tag{3.1}$$

O agente de confiança (Figura 3.1) é o responsável por salvar o valor do tamanho da JED em memória para referência futura. Ele também padroniza os dados reconhecidos para o recurso avaliado, disponibilizando as seguintes informações: a classe identificada, se o mesmo é confiável, atacante OA ou defeituoso. Uma $string$ de saída é gerada contendo os metadados originais, o tipo de recurso reconhecido (classe) e uma nota de confiança. O tipo de recurso e a nota de confiança são projetados para auxiliar o gerenciamento de confiança a decidir sobre o uso de uma informação fornecida por um recurso IoT cooperativo. O agente de confiança implementa as decisões mapeadas na Tabela 3.2.

3.1.1 Definições para a Implementação

Alguns detalhes de implementação devem ser considerados no método proposto. Duas variáveis deverão ser inicializadas para o funcionamento da Janela Elástica Dinâmica (Algorítimo 20): $eswAlpha$ (linha 1) e $swInit$ (linha 2). O $eswAlpha$ permite que o administrador do sistema defina o valor a ser considerado como confiável em uma nota da função de decisão (linhas 11,14,37). Também é utilizado para calcular o grau de crescimento de uma JED (linha 30). A variável $swInit$ registra o tempo JED inicial (em segundos) para um novo recurso.

Algoritmo 1: Algoritmo de implementação do método.

input : A metadata m $(ID, read)$
output: A predicted type:$(Tusted, On-OffAttacker, Broken)$

```
1  eswAlpha ← alpha;
2  eswInit ← beta;
3  NewPrediction ← Classifier.Predict(m);
4  NewDecisionFunction ← Classifier.DecisionFunction(m);
5  if m in Database then
6  |   m.DynamicWindow ← (eswInit + time()) − NewDecisionFunction;
7  |   m.prediction ← NewPrediction;
8  |   if m.DynamicWindow ≥ Time() then
9  |   |   if NewPrediction == −1 and m.prediction == −1 and
   |   |    NewDecisionFunction ≤ eswAlpha then
10 |   |   |   m.prediction ← 0;
11 |   |   end
12 |   |   if NewPrediction != m.prediction and
   |   |    NewDecisionFunction ≥ eswAlpha then
13 |   |   |   m.prediction ← −1;
14 |   |   end
15 |   |   if NewPrediction != m.prediction and
   |   |    NewDecisionFunction ≤ eswAlpha then
16 |   |   |   m.prediction ← m.prediction;
17 |   |   end
18 |   end
19 end
20 m.DynamicWindow ← m.DynamicWindow − NewDecisionFunction;
```

Tabela 3.2: Decisões de saída para duas análises em uma mesma Janela Elástica Dinâmica.

Janela Elástica Dinâmica						**Saída**	
Leitura 1				Leitura 2		Recurso	Tamanho da JED
Classe	Alto	Baixo		Classe	Alto	Baixo		
x	x			x	x		Confiável	Diminui
x	x			x		x	Confiável	Aumenta
x		x		x	x		Confiável	Diminui
x		x		x		x	Confiável	Aumenta
x	x				x		Atacante	Diminui
x	x					x	Atacante	Aumenta
x		x			x		Atacante	Diminui
x		x				x	Atacante	Aumenta
	x			x	x		Atacante	Diminui
	x			x		x	Atacante	Aumenta
		x		x	x		Atacante	Diminui
		x		x		x	Atacante	Aumenta
	x				x		Defeituoso	Diminui
	x					x	Defeituoso	Aumenta
		x			x		Defeituoso	Diminui
		x				x	Defeituoso	Aumenta

Fonte: Elaborada pelo autor.

A ação realizada na linha 9 (Algorítimo 20) verifica se a JED relacionada a um recurso é maior do que o temporizador do computador e deva ser considerada nesta análise em especial. Os novos tamanhos JED são calculados usando valores das JED anteriores subtraídos pelo novo valor da função de decisão. Assim, os valores da função de decisão negativos (não confiáveis) agregam o tamanho da JED (linha 30). As decisões de saída para duas análises na mesma JED são implementadas nas linhas 7, 11, 14 e 17.

O método de identificação de recursos e gerenciamento de confiança retorna resultados através de uma consulta REST/API. Na Figura 3.6 apresenta-se um exemplo de uma análise realizada pelo método. O servidor do método é acessado através do protocolo *Constrained Application Protocol* (CoAP) e retorna os resultados em dados em formato JSON. Como exemplo, o objeto $ID : 14$ com metadados assinalando 46 (graus Celsius) foi marcado como um atacante $On - Off$. Uma pontuação de confiança (o valor da função de decisão) também é apresentada.

Outros possíveis resultados são $Trusted$ para dispositivos confiáveis ou $Broken$ (defeituoso) para dois ou mais valores de falha dentro da mesma JED.

Da mesma maneira, caso o recurso analisado emitir continuamente atributos fora de padrão

Figura 3.6: Exemplo de um resultado de saída pelo servidor de gerenciamento de confiança inteligente.

Fonte: Elaborada pelo autor.

dentro da mesma JED, o mesmo será considerado defeituoso e o método disponibilizará esta informação ao gerenciamento da infraestrutura ao mesmo tempo que aumentará a janela de avaliação, de forma incrementar a abrangência e continuidade da avaliação.

3.2 Considerações do Capítulo

Neste capítulo foi apresentado o método que realiza o reconhecimento semântico de recursos IoT e colabora para o gerenciamento da confiança, avaliando os metadados dos objetos e identificando ataques.

O método proposto colabora para arquiteturas IoT na compreensão dos dados (atributos e leituras) destinados aos recursos, identificando as características do objeto emissor, avaliando a confiança das informações e realizando anotações semânticas padronizadas, que podem ser utilizadas pelas aplicações e usuários. Uma vez que um recurso reconhece e confia em outro objeto identificado, estes estão aptos a realizarem trocas de informações dentro da mesma ou entre outras arquiteturas.

Além da utilização de aprendizado de máquina, o método faz uso da técnica da Janela Elástica Dinâmica, que avalia os objetos IoT em um período de tempo variável, dimensionado pelo grau

de confiança observado nos atributos semânticos e leituras relacionados a estes recursos.

Igualmente, por não demandar implementação integral de todos as camadas de IoT, esta abordagem pode funcionar de forma independente, auxiliando implementações mais complexas como Fog Computing ou mais simples, como ponto-a-ponto (P2P). Além disso, seu funcionamento não demanda a utilização de protocolos proprietários ou customizados, o que facilita sua aplicação em várias implementações, como também a compatibilidade com outras infraestruturas existentes. Essas características integradas propiciam sua adaptação a novos tipos de recursos e aumento na quantidade de objetos conectados.

No próximo capítulo serão detalhadas as metodologias utilizadas para validar o método proposto.

Capítulo 4

Metodologia de validação

Neste capítulo são descritas as metodologias utilizadas para validar o método proposto. As capacidades de reconhecimento semântico de novos recursos IoT e identificação de atacantes foram verificadas utilizando dados reais provenientes de *SmartCities* e também simulações. Da mesma forma, foram realizados ensaios para aferir o desempenho do método em comparação com outros estudos.

As bases de dados utilizadas para treinamento e testes foram obtidas em plataformas de suporte à diversos projetos de de cidades inteligentes e também geradas por simulação. O reconhecimento semântico e a janela deslizante dinâmica introduzida neste trabalho foi desenvolvida de modo a ser compatível com ambientes computacionais aplicáveis a situações reais. A performance na detecção de ataques foi possível de ser comparada com outro estudo relevante a identificação de nós defeituosos foi validade por ensaio de clusterização. Por fim, classificadores de aprendizado de máquina foram comparados de forma a verificar a eficiência da JED.

4.1 Bases de dados Utilizadas para Treinamento e Testes

Para a avaliação do método, foram preparadas bases de dados de treinamento e testes, com dados relacionados a sensores e suas respectivas leituras, disponibilizados publicamente a partir de plataformas de IoT e infraestruturas de cidades inteligentes. Esta pesquisa buscou utilizar, sempre que possível, dados reais de *SmartCities*. Alguns destes dados encontram-se disponibilizados em plataformas para download, como também acessados em tempo real. Para dos dados em

tempo real, foi necessário o desenvolvimento de um aplicativo para acesso e persistências das informações obtidas.

Um ambiente *Swarm* de IoT é composto de vários recursos heterogêneos, que mesmo sendo do mesmo tipo e função, carregam diferentes atributos semânticos e leituras observadas. Para que se fosse possível se aproximar de uma situação de *Swarm*, esta pesquisa buscou acessar informações mais heterogêneas possíveis. Desta forma, foi escolhido utilizar dados de uma plataforma de publicação de leituras de sensores comunitária.

A plataforma ThingSpeak [14] é uma plataforma de Internet de Coisas (IoT) que permite a aquisição e armazenamento de dados de sensores na nuvem e também promove o desenvolvimento integrado de aplicações IoT. Os dados de recursos IoT podem ser enviados para a plataforma a partir de diferentes hardwares baseado em Arduino, Raspberry Pi, BeagleBone, ESP-8266 entre outros equipamentos. Da mesma forma, o uso de padrões abertos facilitam o acesso aos dados dos recursos disponibilizados na plataforma. Em março de 2017, esta comunidade contava com 14426 canais públicos e cerca de 43 mil sensores conectados. Esta diversidade possibilitou criar bases de treinamentos e testes com objetos com diversos tipos de identificações, mas sematicamente equivalentes (como por exemplo os *IDs* de sensores de temperatura escritos em vários idiomas), além da abrangência de leituras observadas.

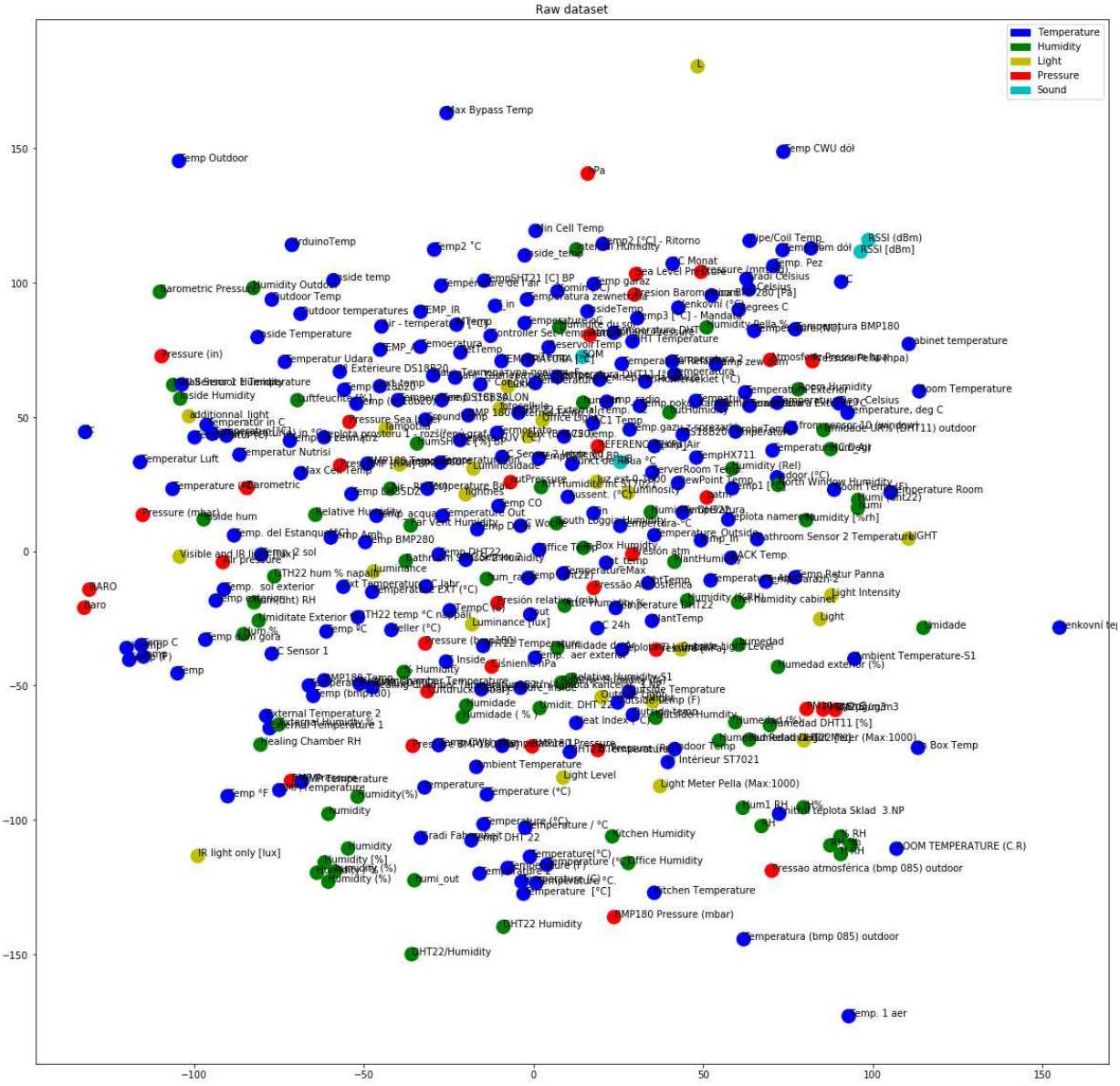
Os conjuntos de dados para treinamento foram extraídos da plataforma na primeira semana de março de 2017, através de um *web crawler* escrito em linguagem Python, totalizando 200mil leituras e encontram-se disponíveis no repositório desta tese[1]. Todos os dados obtidos para as bases de dados foram manualmente classificados, sendo anotado para cada objeto a sua respectiva classe. Assim, o exemplo de um sensor com o ID= ”lamp”, foi classificado como o tipo ”LIGHT”. Da mesma forma um outro objeto com o ID= ”db” foi classificado como sendo do tipo ”SOUND”. Objetos com *IDs* numéricos ou não reconhecidos via ferramenta de tradução foram anotados como ”OUTLIER”.

A visualização (Figura 4.1) é um exemplo deste conjunto de dados e foi gerada através da técnica t-Distributed Stochastic Neighbor Embedding (t-SNE), comumente usada para redução dimensional e visualização de conjuntos de dados de alta dimensão.

Os classificadores utilizados pelo método foram treinados utilizando um mesmo subconjunto de 2000 amostras embaralhadas de diversos tipos de sensores e leituras. Outras quantidades de

[1] https://github.com/jcaminha/thesis

Figura 4.1: Conjuntos de sensores em um ambiente *swarm*.



Fonte: Elaborada pelo autor.

objetos (acima de 3000) também foram avaliadas, mas pouco influenciaram na precisão final dos modelos. Da mesma forma, um outro subconjunto de 2000 amostras foi utilizado como base de validação cruzada, de forma a balizar a otimização dos parâmetros exigidos pelos classificadores.

Para as atividades de testes e validação, foram utilizados dados reais de projetos de *Smart-Cities* IoT que continham sensores do mesmo tipo, mas com atributos semânticos diferentes. Os tipos de sensores avaliados, assim como suas leituras, foram baseados em cinco infraestruturas de cidades inteligentes, listados na Tabela 4.1. Os projetos SmartSantander [90] na Espanha, PE-

Tabela 4.1: Tipos de sensores obtidos nos projetos de *SmartCities*.

Projeto	**Temperatura**	**Umidade**	**Iluminação**	**Som**
SmartSantander[90]	temperature	relativeHumidity	luminousFlux	sound
Trentino[93]	temperatura	umidità	-	-
Thermui[91] in Greece	$\theta\epsilon\rho\mu o\kappa\rho\alpha\sigma\acute{\iota}\alpha$	$\nu\gamma\rho\alpha\sigma\acute{\iota}\alpha$		
Smartcitizen - Europe[92]	DHT22 - Temperature	DHT22 - Humidity	PVD-P8001	POM-3044P-R
Aarhus in Denmark[94]	tempm	hum	-	-

Fonte: Elaborada pelo autor.

OPLE Smart Cities da cidade grega de Thermi [91], SmartCitizen [92], OPENData Trentino da cidade de Trentino na Itália [93] e CityPulse, na cidade de Aarhus na Dinamarca [94] foram utilizados apenas para os testes, de forma a validar o conceito proposto. Um total de 2358 amostras de leituras foram usadas em todos os projetos.

O acesso às informações dessas *Smartcities* pode ser feito de três maneiras: Pelo *download* das bases de dados disponíveis nos repositórios do projeto Fi-WARE [95]; Através do *middleware* mantido pelo projeto OpenIoT[2] ou por uma aplicação que acesse as APIs disponibilizadas no projeto FI-Ware via OpenIoT, sendo esta a opção utilizada neste trabalho. Foi desenvolvida uma aplicação em linguagem Python 3.6 para acesso aos dados das *Smartcities* de modo a facilitar seu uso nas plataformas utilizadas nesta pesquisa. O código fonte desta implementação pode ser visualizado no apêndice A.

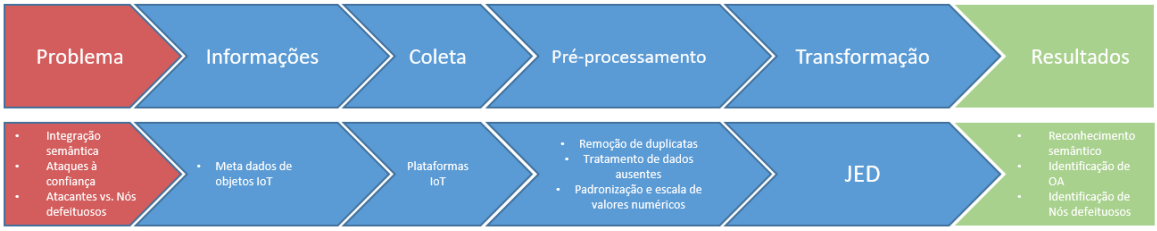
A aplicação desenvolvida possui o seguinte ciclo de funcionamento: No primeiro momento ela acessa a plataforma FI-Ware e obtém os dados dos sensores de uma determinada *Smartcities*. Esses dados são submetidos à aplicação que executa o método (Apêndice B) que retorna o tipo de objeto reconhecido e outras informações. Estas informações podem ser gravadas em banco de dados ou mantidas em memória, a critério do desenvolvedor e da necessidade do serviço.

4.2 Reconhecimento Semântico e Janela Deslizante Dinâmica

A implementação do algoritmo da Janela Elástica Dinâmica (Algorítimo 1) e da API foram realizadas através da linguagem de programação Python 3.6. Da mesma forma, os classificadores (Tabela 4.3) foram treinados e modelados utilizado as implementações da biblioteca Scikit-learn e LIBSVM library [83], versão 0.18.0 em um computador com um processador Core i7 3.60GHz

[2]https://github.com/OpenIotOrg/openiot

Figura 4.2: Fluxo de informação baseado no modelo KDD.



Fonte: Elaborada pelo autor.

executando o sistema operacional Windows 10.

A biblioteca Scikit-learn é um módulo Python que integra uma ampla gama de recursos do estado-da-arte em modelos de aprendizado de máquina supervisionado e não supervisionado para aplicações de média escala. Seu foco prioritário é a utilização de aprendizado de máquina por não especialistas, pelo uso de uma linguagem programação de alto nível. Sendo distribuída sob licença BSD (*Berkeley Software Distribution*) sua ênfase é facilidade de uso, desempenho, documentação e consistência. Esta biblioteca já foi referenciada desde 2011 por cerca de 18.000 trabalhos acadêmicos que utilizam alguma implementação de aprendizado de máquina, além de diversas aplicações comerciais.

O método de validação do fluxo de informação utilizado foi baseado no modelo KDD (Descoberta de conhecimento em bancos de dados) proposto por Saitta et. al. [96] o qual organiza o processo de aprendizado supervisionado nas fases de identificação e entendimento do problema, reconhecimento de informações relevantes, coleta, pré-processamento, transformação e mineração de dados, finalizando com a avaliação e interpretação de resultados.

Neste trabalho, após a delimitação do objeto de estudo (reconhecimento e integração semântica, identificação de ataques à confiança e nós defeituosos), foram pesquisadas bases de dados relevantes relacionadas a cidades inteligentes e também à simulação de dados de ataques. Os dados acessados em plataformas IoT, foram pré-processados, sendo removido dados duplicados e ausentes, além da padronização (escala) dos valores numéricos. Por fim, o método de Janela Elástica Dinâmica foi aplicado e analisado os resultados. Podemos verificar na Figura 4.2 uma comparação das atividades realizadas nesta pesquisa com a metodologia proposto por Saitta et. al. [96].

As bases de dados de treinamento em testes foram preparadas utilizando as recomendações de

pré-processamento da biblioteca Scikit-Learn [83], de forma a melhorar a performance dos métodos de aprendizado de máquina. Os dados passaram pelo processo de sanitização, observando as atividades de remoção de duplicatas, remoção de objetos com rótulos omitidos onde alternadamente algum ou todos os dados estavam ausentes, padronização dos tipos de dados e escala de valores numéricos. Esta atividades foram realizadas utilizando a biblioteca de análise de dados Pandas[3].

Para avaliar a solução utilizamos uma configuração experimental. O servidor do modelo foi instalado em um Raspberry Pi 3 modelo B, rodando o sistema operacional Raspbian[4], sendo responsável por atender as solicitações de dois nós locais ESP-8266 Wifi, que estão atuando como consumidores de informações, como também para conectar a plataforma FI-WARE-Lab [5]. Esta configuração de laboratório de hardware foi utilizada para realizar descobertas de recursos, bem como validar diferentes abordagens de integração. O código-fonte da implementação do método de Janela Elástica Dinâmica proposto neste trabalho está transcrito no Apêndice B.

4.3 Detecção de Ataques e Nós Defeituosos

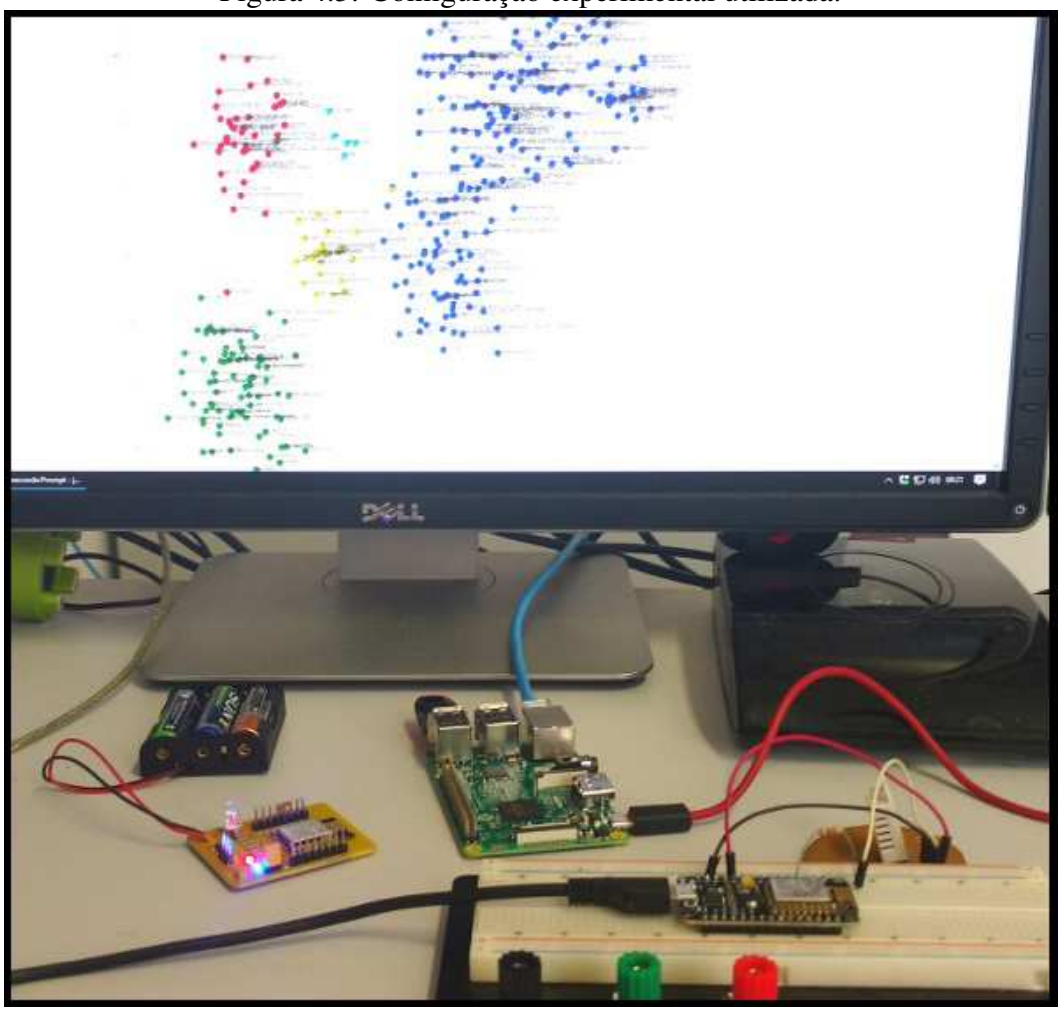
Para validar a detecção de ataques, foram preparados dois experimentos. Uma simulação de computador e um cenário do mundo real. As cenários reais e simulados contém dados anotados para o nós confiáveis, nós atacantes OA e nós defeituosos.

A configuração da simulação consiste em um conjunto de 51 nós (Figura 4.4). Um nó foi configurado como o objeto de destino, que representa o consumidor das informações e executa o método proposto neste trabalho. Quarenta nós atuaram como nós confiáveis (laranja), fornecendo apenas valores corretos. Cinco nós (rosa) são atacantes OA, fornecendo aleatoriamente valores confiáveis e não confiáveis. Também simulamos cinco nodos defeituosos (amarelo), que sempre enviam dados não confiáveis.

Para os experimentos de simulação, foi utilizado o simulador Cooja para o sistema operacional Contiki 3.0 [97]. A simulação foi executada durante duas horas em uma computador de mesa com processador Core i7 de 3,60 GHz executando o Windows 10 gerando um conjunto de 4844 amostras de dados. O simulador Cooja é popular dentro da comunidade de pesquisa WSN e IoT

[3]https://pandas.pydata.org/
[4]https://www.raspberrypi.org/downloads/raspbian/
[5]https://cloud.lab.fiware.org/

Figura 4.3: Configuração experimental utilizada.

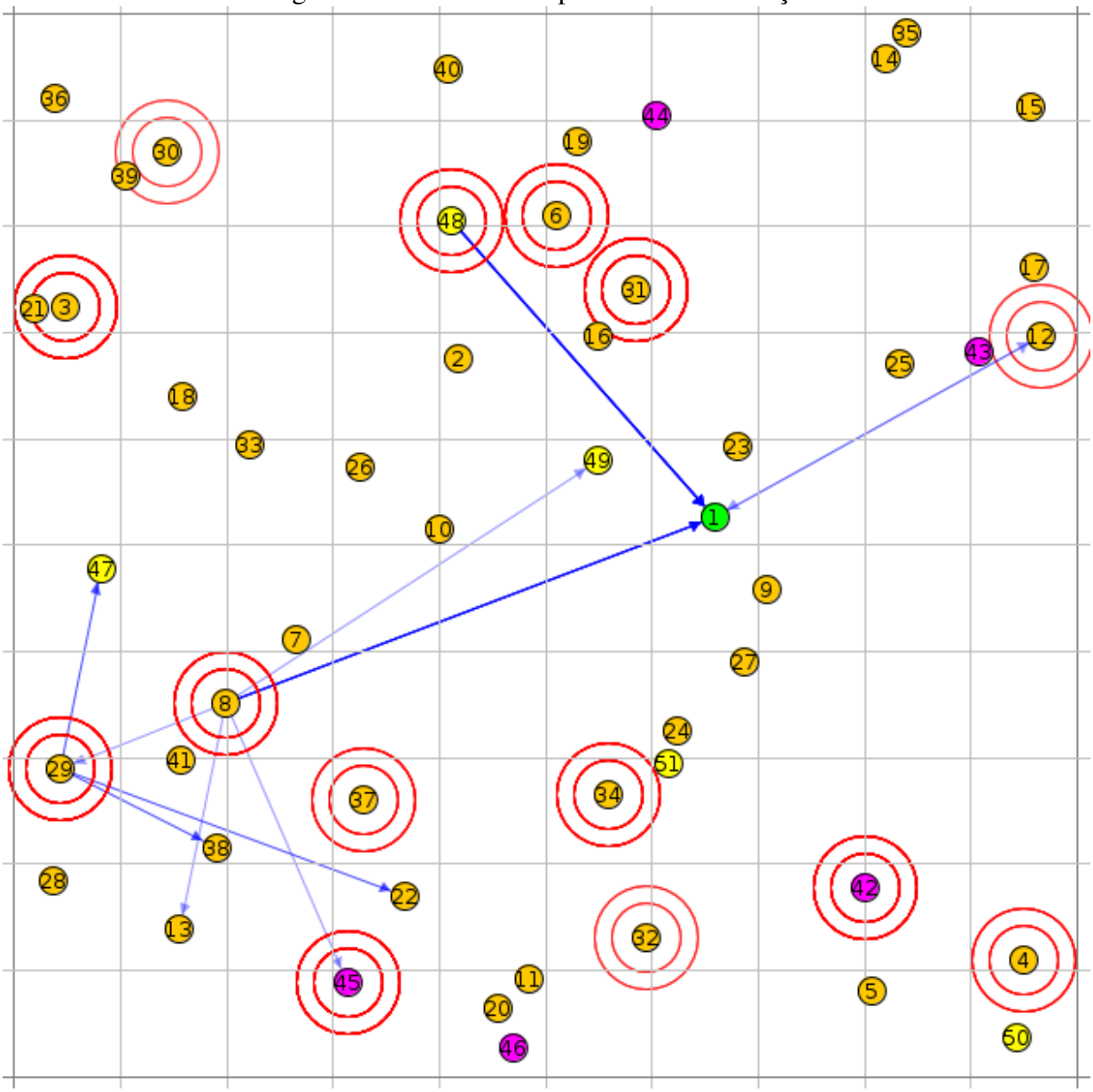


Fonte: Elaborada pelo autor.

e os resultados podem ser comparados com outros estudos. Os parâmetros de configuração da simulação estão listados na Tabela 4.2

Para avaliar o método em um cenário real, também utilizamos 4111 amostras de dados de temperatura obtidos de fevereiro a março de 2015 da cidade de Aarhus, localizado na Dinamarca [94]. Esta cidade tem um intervalo de temperatura regular durante esta época do ano, variando de -3 a 16 graus Celsius (Figura 4.5). Um total de 500 amostras de ataque foram simuladas usando observações aleatórias de temperatura fora de alcance (de -23 a 36 graus Celsius) e injetadas no conjunto de dados do teste.

Figura 4.4: Cenário de experiência da simulação.



Fonte: Elaborada pelo autor.

4.4 Comparação de Classificadores

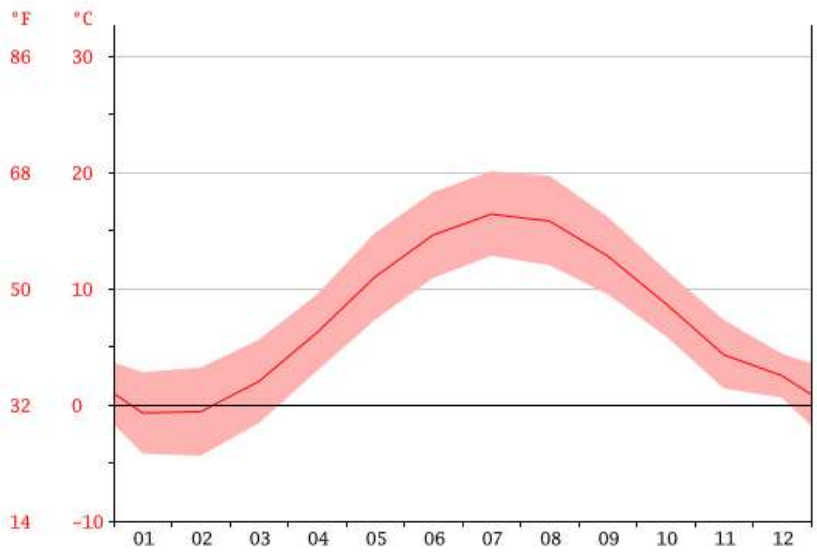
Um ensaio comparativo também foi conduzido para identificar como o método proposto pode ser comparado a soluções utilizando classificadores de aprendizado de máquina. Foram realizados testes com os classificadores supervisionados baseados em rede neural (multi-layer perceptron), como também o linear SVM, Random Forest, Naive Bayes (GaussianNB). Todos os classificadores foram treinados e testados utilizando o mesmo conjunto de dados. Na Tabela 4.3 estão listadas

Tabela 4.2: Resumo dos parâmetros de simulação.

Parâmetros	**Valores**
Simulador	Cooja sobre Contiki 3.0 OS
Ambiente de rádio	Unit disk graph medium (UDGM): dist. loss
Área de implementação	400m X 400m
Tipo & no. de nós	Sky mote, 50 emissores & 1 sorvedouro
Alcance dos nós	50m
Camada física	IEEE 802.15.4
Camada MAC	IPv6
Camada de rede	RPL
Camada de transporte	UDP
Duração da simulação	2h
Taxa de envio	1 pacote a cada 20-60 segundos

Fonte: Elaborada pelo autor.

Figura 4.5: Escala de temperatura observada na cidade de Aarhus.



Fonte: Extraído de https://en.climate-data.org/location/302/.

as configurações usadas em cada classificador testado.

A precisão dos classificadores foi calculada pela Fórmula 4.1, onde P é definido pelo número de positivos verdadeiros (Tp) sobre o número de positivos verdeiros mais a quantidade de falsos positivos (Fp), enquanto o *Recall* se refere ao número de positivos verdadeiros (Tp) sobre a quantidade de positivos verdadeiros e falsos negativos (Fn) somados (Equação 4.2).

$$P = \frac{Tp}{Tp + Fp} \tag{4.1}$$

Tabela 4.3: Configuração dos classificadores.

Classificador	**Configuração**
Linear SVM	(C=0.025, cache_size=200, class_weight=None, coef0=0.0,,decision_function_shape=None, degree=3, gamma='auto', kernel='linear',,max_iter=-1, probability=False, random_state=None, shrinking=True,,tol=0.001, verbose=False)
Random Forest	RandomForestClassifier(n_estimators=10, criterion='gini', max_depth=None, in_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=1, random_state=None, verbose=0, warm_start=False, class_weight=None)
Neural Net	MLPClassifier(activation='relu', alpha=1, batch_size='auto', beta_1=0.9,,beta_2=0.999, early_stopping=False, epsilon=1e-08,,hidden_layer_sizes=(100,), learning_rate='constant',,learning_rate_init=0.001, max_iter=200, momentum=0.9, nesterovs_momentum=True, power_t=0.5, random_state=None,,shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1,,verbose=False, warm_start=False)
Naive Bayes	GaussianNB(priors=None)

Fonte: Elaborada pelo autor.

$$R = \frac{Tp}{Tp + Fn} \tag{4.2}$$

Um sistema com alto *recall* mas baixa precisão retorna muitos resultados, sendo a maioria incorretos, enquanto outro com alta precisão e baixo *recall* tem um comportamento oposto: retorna poucos resultados, porém corretos. Um sistema ideal possui uma alta precisão com um alto *recall* [83].

4.5 Considerações do Capítulo

Neste capítulo foram detalhadas as metodologias utilizadas para validar o método proposto neste trabalho. As capacidades de reconhecimento semântico de novos recursos IoT e identificação de atacantes foram verificadas utilizando dados reais provenientes de *SmartCities* e também simulações. Da mesma forma, foram realizados ensaios para aferir o desempenho do método para comparação com outros estudos.

Também foram descritos os classificados testados para o processo de aprendizado de máquina utilizado no método, comparando os principais abordagens usadas atualmente.

No capítulo seguinte, serão apresentados e discutidos os resultados desta validação.

Capítulo 5

Resultados e Discussões

Neste capítulo são apresentados os resultados da validação do método proposto. Os resultados para as capacidades do método em reconhecer semanticamente os atributos dos objetos e identificar ataques são mostrados separadamente. O capítulo inicia-se com a os resultados referentes a identificação semântica de recursos, segue com o detalhamento da performance na identificação de atacantes e nós defeituosos e finaliza com uma análise das ameaças à validação deste trabalho.

5.1 Identificação Semântica de Recursos

O método proposto neste trabalho foi avaliado em tarefas de descoberta de recursos em cinco infra-estruturas de cidades inteligentes disponíveis publicamente, acessadas através da plataforma FIWARE-Lab [95]. Um conjunto de 2358 sensores heterogêneos (diferentes identificações, metadados e leituras) foi avaliado. O método descobriu 2263 recursos IoT com precisão de 96%.

As informações originárias dos recursos acoplados na infraestrutura foram pré-processados, normalizados e submetidos ao um método de aprendizado de máquina supervisionado. Na solução proposta neste trabalho, o classificador utilizado foi o LinearSVC, que possui a capacidade de tratar de forma eficiente uma grande quantidade de amostras numéricas [98]. A biblioteca Scikit-Learn [83] possui a implementação do método "*decision_function*", que retorna os índices de confiança para amostras avaliadas. A pontuação de confiança para uma amostra é a distância assinalada dessa amostra para o hiperplano de decisão. Estes valores são utilizados para o dimensionamento da JED.

Tabela 5.1: Exemplos de Saídas.

Leitura	**Saída estimada**	**Nota de confiança (%)**	**Saída esperada**
t=50	TEMPERATURE	96,47	TEMPERATURE
temp=50	TEMPERATURE	99,04	TEMPERATURE
pte=180	TEMPERATURE	29,54	TEMPERATURE
light=50	LIGHT	06,50	LIGHT
light=500	LIGHT	100,00	LIGHT

Fonte: Elaborada pelo autor.

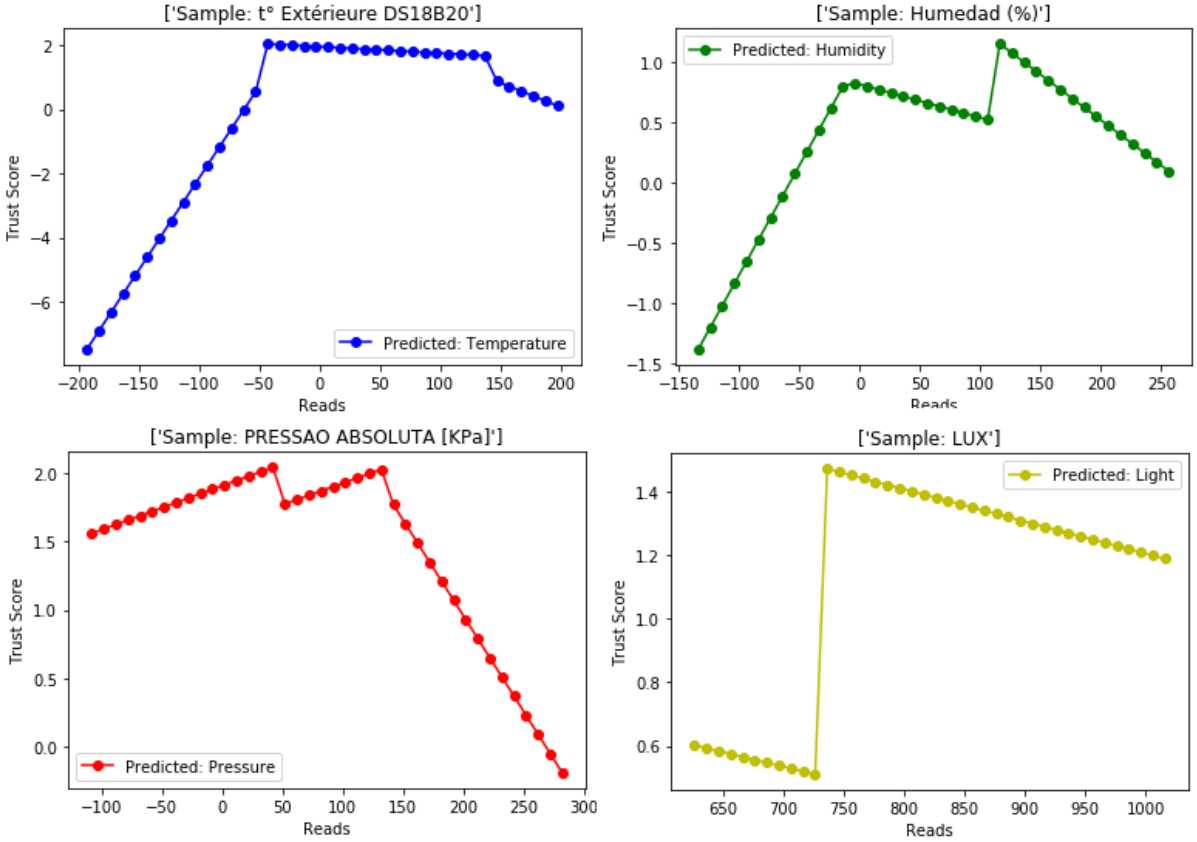
Apesar de não ter figurado como a técnica mais eficiente no reconhecimento das classes para o tipo de problema estudado (Tabela 5.2), este classificador foi escolhido pela facilidade em determinar e calcular a função de decisão, como também demonstrar que o método proposto não depende exclusivamente da eficiência de classificadores. Os outros classificadores avaliados, por sua vez, também não retornavam resultados significativos relacionados a função de decisão que pudessem ser utilizados pelo JED.

Durante o processo de reconhecimento, o método recebeu dados de solicitação de um aplicativo acerca de um recurso IoT e responde com a informação semântica descoberta e uma pontuação de confiança. Considere o exemplo de que uma leitura ”t = 50” foi prevista como um sensor de temperatura com 96,47% de confiança. Esse valor foi obtido pelo método devido ao fato de que ”t” foi definido como nome relacionado semanticamente a um sensor de temperatura durante o processo de treinamento. No entanto, "t"também pode ser um identificador para um sensor de luz, uma vez que ”t” faz parte do termo ”light”. O aspecto diferencial observado neste caso particular foi o valor da leitura: o valor 50 °C está mais relacionado a um sensor de temperatura do que a um sensor de luminosidade, dado o possível gradiente de valores para este tipo de recurso. Na Tabela 5.1 estão listados outros exemplos de saídas de consultas.

Ao analisar as leituras *”t=50”* e *”temp=50”* (Tabela 5.1) podemos compreender como a variação semântica modifica a nota de confiança. Quanto menos ambígua a semântica da etiqueta do atributo, maior será essa nota. Do mesmo modo, a variação nas leituras numéricas para o mesmo rótulo também influencia de maneira direta esta nota, como nos exemplos ”light=50” e ”light=500” que retornaram as notas de confiança 6.50 e 100, respectivamente.

Na Figura 5.1 podemos observar exemplos da correlação entre as leituras disponibilizadas e as notas de confiança. Para cada tipo de recurso, após a identificação de sua classe utilizando os metadados semânticos (etiquetas) e numéricos (leituras), foi possível verificar como a nota de con-

Figura 5.1: Correlação entre as leituras observadas e as notas de confiança.



Fonte: Elaborada pelo autor.

fiança varia conforme os valores das leituras se distanciam dos valores esperados (possíveis) para aquele tipo de recurso. Com essa informação, um sistema pode selecionar o melhor (mais confiável) recurso dentre outros disponíveis em um mesmo contexto. Outra característica importante é que o uso de classificadores baseados em aprendizado de máquina facilita a absorção de variações nas leituras observadas sem que as mesmas sejam apontadas como *outliers* ou pertencentes a outra classe.

O exemplo do sensor de temperatura ”t Extérieure DS18B20” (gráfico superior esquerdo na Figura 5.1) evidencia o funcionamento do método. Um atributo semântico (nome) escrito em idioma francês e acrescido de códigos diversos foi corretamente mapeado como um sensor de temperatura. Valores distantes das variações esperadas para um sensor de temperatura (-200 ou +200) começam a receber notas negativas de confiança.

Os saltos nos valores das notas de confiança, como observadas nos gráficos superior direito e inferiores, podem ser explicados pelo conjunto de dados de treinamento utilizado. Como o

Tabela 5.2: Comparação dos classificadores para o reconhecimento semântico.

Classificador	**Precisão**	**Recall**	**F1-Score**
Linear SVM	0,88	0,71	0,74
Naive Bayes	0,92	0,82	0,85
Neural Net	0,92	0,81	0,84
Nearest Neighbors	0,91	0,84	0,87
JED	**0,96**	**0,85**	**0,88**

Fonte: Elaborada pelo autor.

conjunto de dados de treinamento foi extraído de leituras reais, os valores que geraram os saltos estão relacionados as amostra faltantes nesta base de treinamento. Por outro lado, a função de decisão criada pelo classificador de aprendizado de máquina permitiu atenuar e absorver essas ausências.

Na Figura 5.2 é possível observar um exemplo da precisão do método na identificação do tipo exato dos objetos para a variedade dos atributos semânticos apresentados na base de testes. Os círculos preenchidos representam os tipos verdadeiros dos recursos enquanto os círculos em volta, os tipos que foram identificados. Nesta visualização, todos os recursos foram previstos corretamente.

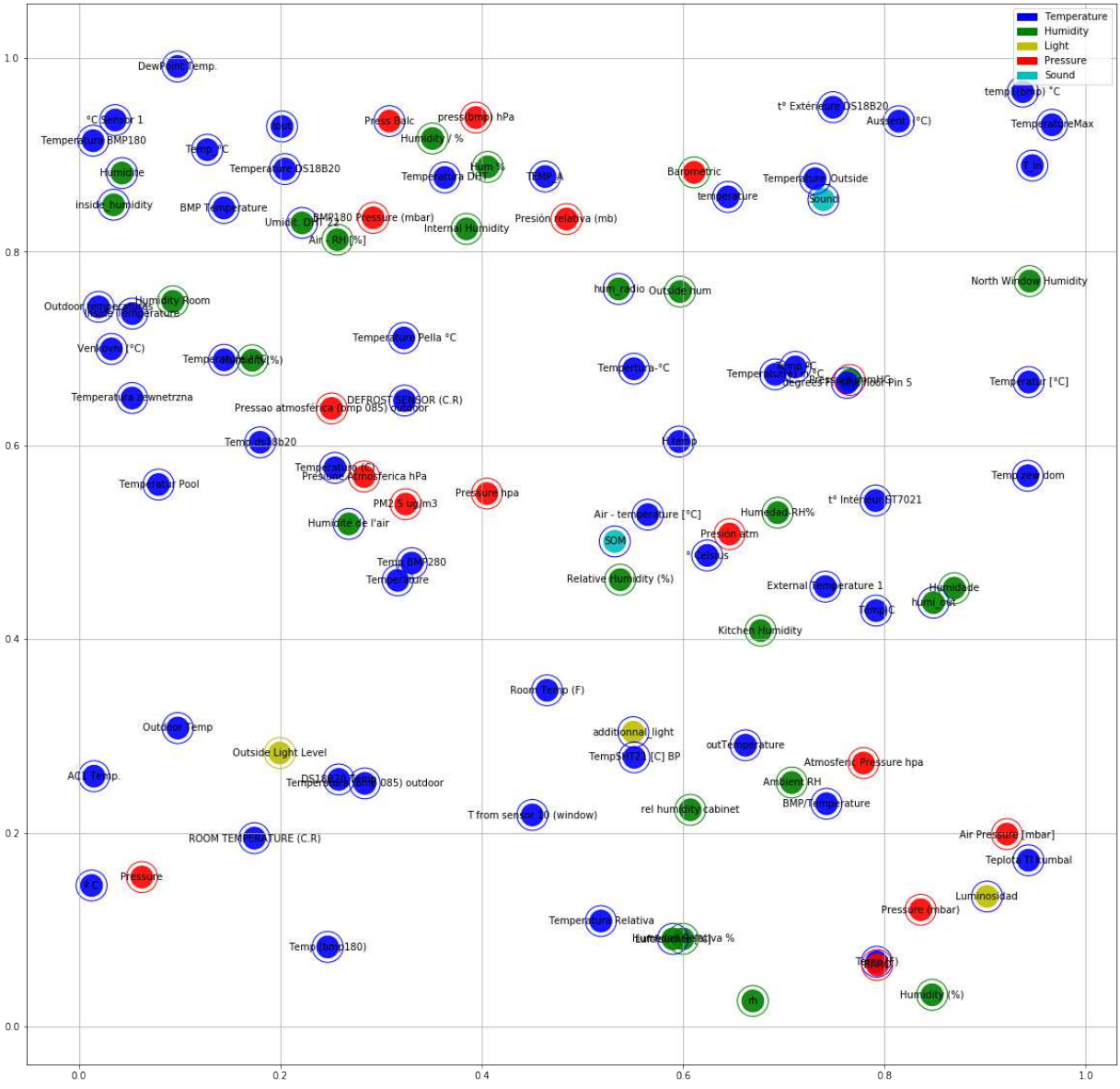
Finalmente, para mensurar a eficiência geral do método proposto, foi realizada uma comparação com outros métodos classificadores baseados em aprendizado de máquina supervisionado. Utilizando a mesma base dados para treinamento e testes, o método formado pela janela elástica dinâmica e classificador LINEAR SVM alcançou 96% de precisão. frente aos classificadores Naive Bayes e baseados em rede neural (92%). Outros resultados desta comparação estão descritos na Tabela 5.2.

Destaca-se ainda que o método proposto neste trabalho, mesmo utilizando um classificador que isoladamente entregou uma performance inferior, conseguiu atingir resultados superiores em todos as métricas comparadas (precisão, *recall* e F1-Score).

5.2 Identificação de Atacantes OA e Nós Defeituosos

O método introduzido também foi capaz de detectar OA em IoT com 97% de precisão em um conjunto de dados real e com 96% de precisão no ambiente simulado. Em comparação com outros estudos, o método foi 95% mais rápido e 5% mais preciso na identificação de OA. O recurso

Figura 5.2: Recursos identificados na base de testes.

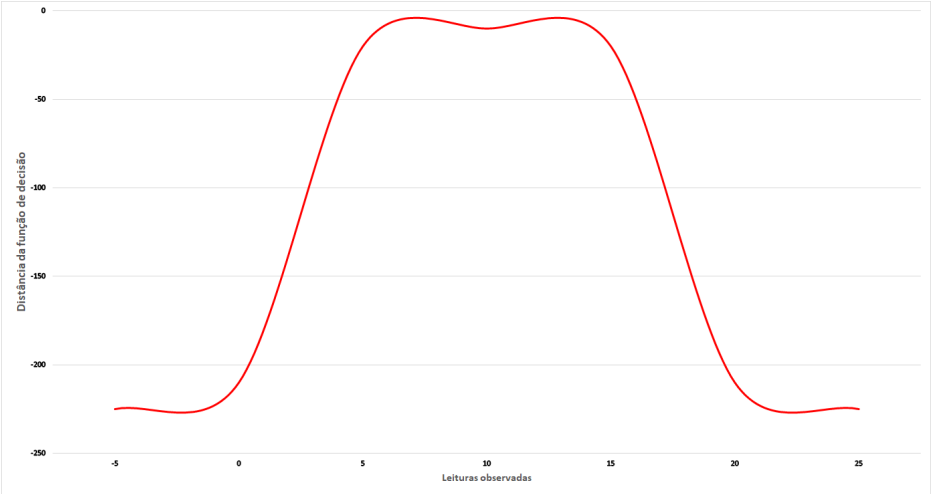


Fonte: Elaborada pelo autor.

de janela elástica dinâmica permitiu diferenciar nós quebrados ou com defeito entre os recursos atacantes.

O classificador utilizado como suporte a JED retorna um valor de distância de uma amostra avaliada a partir da função de decisão de hiper-plano. Se uma classe for identificada, as observações normais (confiáveis) têm valores próximos de 0, enquanto os ataques (leituras erradas ou fora do alcance) possuem valores de distância altos (até -200). Na Figura 5.3 exibe-se as notas de confiança (distância da função de decisão) para um conjunto de leituras de temperaturas

Figura 5.3: Distância da função de decisão para leituras de temperaturas.



Fonte: Elaborada pelo autor.

observadas.

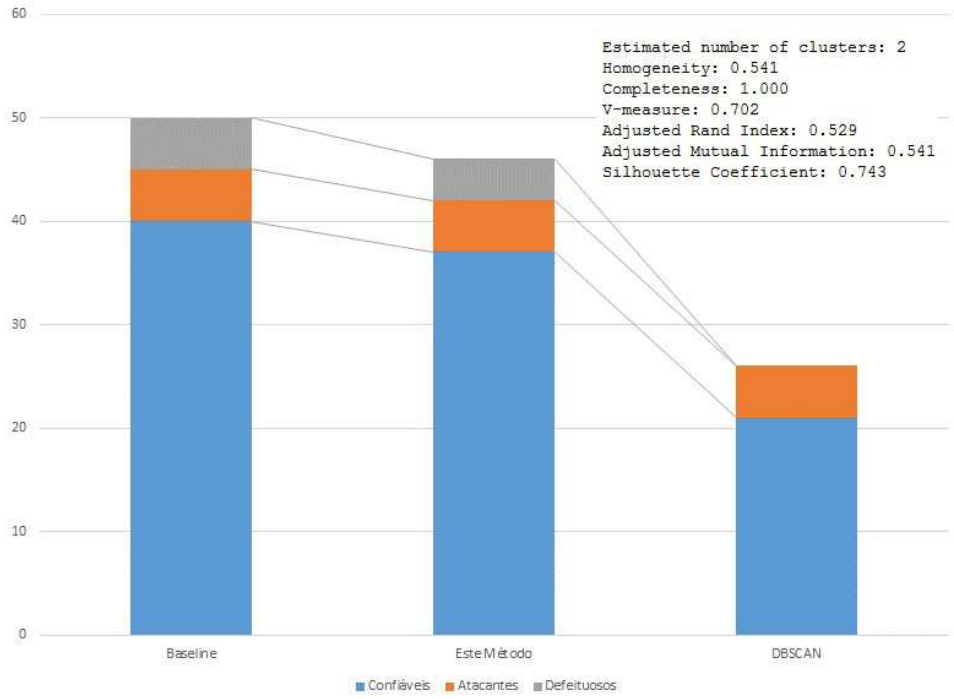
Para essa avaliação, O classificador foi treinado com um conjunto de 2000 leituras relacionadas ao intervalo de temperatura observado na cidade de Aarhus (-3 a 16 graus Celsius). As leituras de teste dentro dessa faixa alcançam valores de distância da função de decisão perto de zero, enquanto outros valores recebem valores negativos.No cenário real, o método foi capaz de identificar 485 OAs dentro das 4611 amostras avliadas, alcançando uma precisão de 97%.

Em várias situações, nem todos os dispositivos que emitem leituras fora de padrão são atacantes. Alguns podem ser dispositivos em um estado de mau funcionamento. Como não foi identificado pesquisas relevantes na diferenciação entre OA e erros enviados por nós defeituosos, comparamos a solução proposta com o algoritmo de clusterização *Density-based spatial clustering of applications with noise* (DBSCAN) [99]. O DBSCAN identificou apenas duas classes (identificados e *outliers*) . Os resultados na Figura 5.4 mostram-se os nós positivos verdadeiros (bons, atacantes e nós quebrados) previstos a partir do método apresentado e do DBSCAN, em comparação com os nós esperados (linha de base).

A base de testes possuía cinco nós anotados como defeituosos, destes quais quatro foram identificados pelo o JED. Por sua vez, todos os nós atacantes foram precisamente reconhecidos pelo método introduzido neste trabalho.

Em um outra visualização (Figura 5.5) apresentam-se os nós que foram identificados pelo método proposto em ambiente de simulação. Os círculos cheios são os nós reais. Os amarelos são

Figura 5.4: Comparação da JED com o algoritmo de clusterização DBSCAN.



Fonte: Elaborada pelo autor.

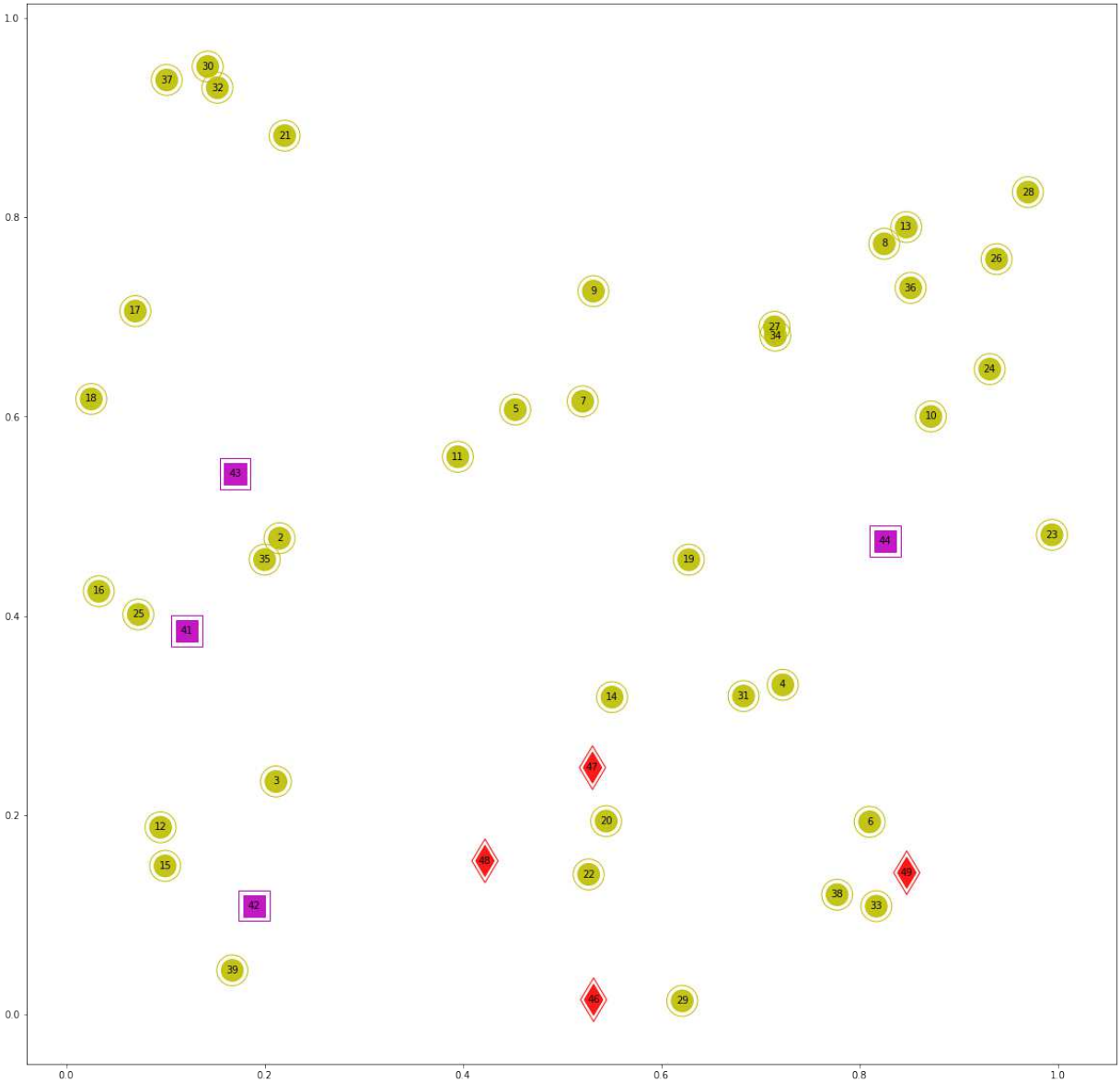
os nós confiáveis, os cianos são nós atacantes e os vermelhos são os nós defeituosos. O círculo ao redor dos nós representa a classe que identificada. Confirmando os resultados do obtidos na comparação com o DBSCAN, os quatro nós anotados como defeituosos e os cinco atacantes foram identificados pelo o JED.

Em suma, o método introduzido foi capaz de encontrar três bons nós, dois atacantes e dois nós quebrados a mais que os outros métodos supervisionados em ambiente simulado.

A performance na identificação de OAs foi comparada com o estudo [67] de Mendoza et. al. Os pesquisadores utilizaram como dados de entrada o número, a posição e o volume de tráfego de nós maliciosos, exigindo aproximadamente 120 min para determinar OAs. O método introduzido neste trabalho identificou OAs em um tempo médio de 5 min (95% mais rápido) com 96% de precisão.

Na Figura 5.6 mostra-se o tempo gasto para identificar OA. Os nós 31* (confiável), 8* e 32* (atacantes) são do estudo comparado e os nós 5, 9, 22 (confiável), 41, 42 e 43 (OA) são provenientes do cenário simulado neste trabalho. Os nós 46, 47 e 48 são nós defeituosos e também foram identificados em 7 min. As pontuações de confiança positivas estão relacionadas a nós

Figura 5.5: Recursos identificados pelo método.



Fonte: Elaborada pelo autor.

confiáveis enquanto a pontuação de confiança negativa referem-se aos atacantes, respectivamente.

A alta performance do método explica-se pela sua abordagem que utiliza as interações entre vizinhos e persistência das notas de confiança, não necessitando da propagação das estimativas de confiança por todo o sistema para convergir uma nota de confiança final.

Figura 5.6: Tempo necessário para identificar atacantes OAs.

Fonte: Elaborada pelo autor.

5.3 Ameaças à Validação

Algumas ameaças à validação deste trabalho precisam ser enumeradas ou tratadas dentro das possibilidades. Com referência às bases de dados utilizadas para estudos, não conseguimos identificar bases de referências (*golden standards*) que podem ser utilizadas em comparações entre estudos, o que nos levou a optar por utilizar dados reais de IoT e dados simulados. Entretanto, destaca-se a baixa quantidade de base de dados públicas de *Smarticities*, bem como a quantidade desbalanceada de tipos de sensores e registros de amostras disponíveis. Para o tratamento desta condição, foram escolhidas cidades com os mesmos tipos de sensores e determinado uma variação máxima de 15% no número de amostras entre os *datasets* de forma a não gerar um conjunto de dados pequeno ou desbalanceado. Erros de leitura ou eventuais não padronizações de dados observadas foram marcados manualmente como *outliers*.

As saídas de simulações podem variar de cada período de execução. Para minimizar essa ameaça, cada simulação foi executada três vezes, sendo utilizado a média dos resultados. Os responsáveis pela biblioteca scikit-learn [83] alertam que os classificadores de usuários usam valores de semente aleatórios em tarefas de ajuste, o que corrobora ligeiramente para resultados de preci-

são diferentes. Como nas simulações, anotamos valores médios de três rodadas de treinamento. O conjunto de dados real usado foi sanitizado removendo valores NAN e dados não relacionados.

A seção de trabalhos relacionados pode ter sido ameaçada ao não considerar algum estudo relevante. Para minimizar esse risco, consultamos os principais bancos de dados de indexação para verificar referências e citações de/para estudos selecionados. No entanto, alguns estudos podem não ter sido considerados devido à limitação técnica dos motores de busca, a possibilidade de bancos de dados eletrônicos não representarem a lista completa de todos os estudos disponíveis e a não disponibilidade para a rede acadêmica brasileira ou o acesso exclusivo apenas para a versão impressa.

5.4 Considerações do Capítulo

Neste capítulo foram apresentados os resultados da validação do método proposto. Os resultados relacionados a capacidade do método em reconhecer semanticamente os atributos dos objetos e identificar ataques são mostrados separadamente.

O método utilizando o classificador Linear SVM com a implementação da Janela Elástica Dinâmica alcançou uma precisão de 96% na identificação semântica de atributos IoT, sendo 4,3% mais eficaz que o melhor método de aprendizado de máquina implementado de forma isolada. Mesmo utilizando um classificador com uma precisão inferior, a JED permite uma agregação de performance geral ao método, como também flexibiliza o modelo para a aplicação em outros problemas, como a segurança.

Para ameaças à confiança do tipo *On-Off*, o método foi 95% mais eficiente em termos de performance, identificando todos os nós atacantes em cinco minutos de simulação, em contraste aos 120 minutos utilizados pelo o estudo comparado. Em aplicações dinâmicas e em tempo real para IoT, a detecção rápida de atacantes é fundamental para a proteção de todo o sistema.

De maneira inédita, o método também permitiu separar nós defeituosos de nós atacantes, identificando quatro dos cinco nós defeituosos no ambiente simulado, sendo ao mesmo tempo 100% mais eficiente ao método de clusterização utilizado para a comparação. Assim, o método também colabora para a continuidade dos serviços IoT, uma vez que os administradores de um sistema são alertados da presença de nós defeituosos e providenciarem a devida manutenção.

No capítulo seguinte, serão realizadas as discussões finais e conclusão deste trabalho.

Capítulo 6

Conclusão

Neste trabalho foi apresentado um método para o reconhecimento semântico de recursos IoT, que auxilia a integração de infraestruturas e serviços, colaborando para o gerenciamento da confiança, através da avaliação automatizada da confiança dos objetos IoT ao analisar os atributos do provedor dos dados. A aplicação do método introduzido foi capaz de detectar ataques do tipo On/Off com 97% de precisão em um conjunto de dados real e com 96% de precisão em ambiente simulado. Em comparação com outros estudos, o método foi 95% mais rápido na identificação de ataques do tipo *On-Off* e também foi capaz de diferenciar nós atacantes de nós defeituosos.

O método introduzido é baseado em duas tecnologias: Aprendizado de Máquina e Janela Elástica Dinâmica (JED). O aprendizado de máquina auxila no reconhecimento de classes de recursos e cálculo da confiança, enquanto a JED proporciona uma análise aprofundada dos recursos IoT em um período de tempo flexível. Funcionando em arquitetura de computação em nuvem e sendo acessado através de protocolo CoAP, o método possui flexibilidade para atender vários tipos de arquiteturas, sistemas e recursos IoT, inclusive aqueles com baixo poder computacional.

De maneira integrada, o método fornece simultaneamente as arquiteturas IoT quatro funcionalidades relacionados a serviços e segurança: O reconhecimento e padronização de atributos semânticos, emissão de nota de confiança acerca dos objetos, identificação de atacantes do tipo OA e separação de nós atacantes de nós defeituosos.

A abordagem apresentada atende as necessidades dos requisitos técnicos de comunicação entre dispositivos heterogêneos, evoluindo o uso e entendimento da semântica das informações e atributos dos recursos para a integração e disponibilização dos serviços de IoT. O método inteligente e

dinâmico atua de forma diferente de outros métodos de mapeamento semânticos tradicionais que possuem deficiências no tratamento do dinamismo exigido pelas arquiteturas IoT, quer seja por utilizar tratativas manuais, tecnologias adaptadas como o *Service-Oriented Architecture* (SOA) ou mesmo pela ausência de padrões. Além de solucionar este problema, o método verifica o nível de confiança das informações utilizadas, prevenindo ataques à segurança.

Além de identificar ataques à confiança do tipo OA, o método é capaz de diferenciar, de forma inédita, nós atacantes daqueles que se encontram em um estado de mal funcionamento. A identificação desses nós permite que os administradores e serviços IoT executem as tarefas apropriadas para a proteção ou conserto destes recursos do sistema.

O método também foi capaz reconhecer automaticamente atributos semânticos nos metadados dos recursos IoT, com a finalidade de executar tarefas de padronização e mapeamento, facilitando seu entendimento e uso por usuários e outras aplicações diferentes dos planejados inicialmente. Esta funcionalidade permite também a integração de arquiteturas e soluções, aumentando a granularidade de objetos, sem demandar intervenção manual dos administradores.

Os dados disponibilizados pelos recursos IoT, como por exemplo as leituras observadas por sensores, são analisados pelo método introduzido em busca de semelhanças com outros dados confiáveis. Esta análise é realizada pela mensuração da distância da amostra analisada em relação função de decisão da classe relacionada. Assim, foi possível emitir uma nota (*score*) que pode ser utilizada por sistemas de gerenciamento de confiança IoT para a seleção dos recursos que irão prover informações para o serviço.

Os ataques do tipo OA comprometem a confiança de sistemas IoT provendo informações confiáveis e incorretas alternadamente. A JED implementada no método, permite a análise dos dados providos pelos objetos IoT em um período de tempo dinâmico, ajustado pela nota de confiança. Esta análise permite realizar uma verificação mais sensível para identificação de comportamentos anômalos.

Por último, a JED também suporta a identificação de nós defeituosos em um conjunto de nós atacantes. Esta atividade permite que os administradores e serviços IoT executem as tarefas apropriadas para a proteção ou conserto dos recursos do sistema.

Os resultados obtidos na implantação do método proposto confirmam a possibilidade de realizar a integração flexível e confiável de objetos IoT, através do reconhecimento e análise semântica dos atributos e dados dos recursos IoT. Desta forma, o alcance do objetivo principal dessa tese

permitiu validar a hipótese considerada que: "Um método baseado em aprendizado de máquina e análise temporal das informações de atributos relacionados a objetos IOT pode habilitar a integração de sistemas, ao mesmo tempo em que colabora para o gerenciamento da confiança e diferencia nós atacantes de defeituosos".

O desenvolvimento desta pesquisa também permitiu o alcance de alguns objetivos específicos. O extenso levantamento dos trabalhos relacionados propiciou a compreensão do estado da arte dos modelos utilizados pelas arquiteturas de IoT atuais disponíveis, ao que concerne a integração semântica, suas limitações e soluções. Os experimentos conduzidos neste trabalho permitiram identificar um conjunto de tecnologias e métodos apropriadas para o desenvolvimento da solução eficiente e auto-configurável, capaz de reconhecer semanticamente atributos e identificar ataques a confiança. O método proposto foi validado utilizando dados reais obtidos de várias *SmarCities*, como também em ambiente de simulação, tendo sua efetividade medida por métricas apropriadas de precisão e *recall*.

A IoT, por definição, oferece serviços de informação conectando uma variedade de dispositivos com características heterogêneas, com diferentes capacidades computacionais e padrões de projetos, muitas vezes incompatíveis entre si. A integração confiável de recursos é um desafio que precisa ser superado para o oferecimento de melhores serviços aos usuários. Os resultados apresentados neste trabalho contribuem com o desenvolvimento da IoT por intermédio de um método automatizado para o reconhecimento de semântico de recursos IoT, permitindo que o gerenciamento da confiança identifique atacantes e recursos defeituosos.

Para trabalhos futuros, objetiva-se o aumento da precisão geral do método usando outros conjuntos de dados para treinamento, assim como a utilização de outros classificadores baseados em aprendizagem profunda (*Deep Learning*). Também pretende-se usar a janela elástica dinâmica para identificar outros tipos de ataques relacionados com confiança da IoT, como ataques oportunista à serviços, *ballot-stuffing*, *bad-mouthing* e ataques de auto-promoção. Além disso, busca-se evoluir o método com outras capacidades de *Trust*, como a composição e a propagação das notas de confiança, além de técnicas de fusão de dados, de modo a deixá-lo o mais abrangente possível para as necessidades de confiança em arquiteturas dinâmicas de Internet das Coisas.

Referências bibliográficas

1 NORDRUM, Amy. *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*. 2016. 1 p. Disponível em: <https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>.

2 ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo; DIEE, A. The Internet of Things: A survey. *Computer Networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. ISSN 13873326.

3 FATHY, Yasmin; BARNAGHI, Payam; TAFAZOLLI, Rahim. Large-Scale Indexing , Discovery and Ranking for the Internet of Things (IoT). Association for Computing Machinery (ACM), v. 9, n. 4, p. 1–67, 2010. Disponível em: <http://epubs.surrey.ac.uk/844698/>.

4 FERRAG, Mohamed Amine; MAGLARAS, Leandros A.; JANICKE, Helge; JIANG, Jianmin. Authentication Protocols for Internet of Things: A Comprehensive Survey. *Security and Communication Networks*, Hindawi, v. 2017, p. 1–41, 11 2016. ISSN 1939-0114. Disponível em: <http://arxiv.org/abs/1612.07206>.

5 KHAN, Rafiullah; KHAN, Sarmad Ullah; ZAHEER, Rifaqat; KHAN, Shahid. Future internet: the internet of things architecture, possible applications and key challenges. In: IEEE. *Frontiers of Information Technology (FIT), 2012 10th International Conference on*. [S.l.], 2012. p. 257–260.

6 YANG, Zhihong; YUE, Yingzhao; YANG, Yu; PENG, Yufeng; WANG, Xiaobo; LIU, Wenji. Study and application on the architecture and key technologies for IOT. In: IEEE. *2011 International Conference on Multimedia Technology*. 2011. p. 747–751. ISBN 978-1-61284-771-9. Disponível em: <http://ieeexplore.ieee.org/document/6002149/>.

7 WANG, Wei; DE, Suparna; TOENJES, Ralf; REETZ, Eike; MOESSNER, Klaus. A comprehensive ontology for knowledge representation in the internet of things. In: IEEE. *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012*. [S.l.], 2012. p. 1793–1798. ISBN 9780769547459. ISSN 2324-898X.

8 CHUI, M.; LÖFFLER, M.; ROBERTS, R. The Internet of Things. In: IEEE. *Communications & Strategies*. [S.l.], 2010. v. 5, p. 6. ISBN 9781522521044.

9 AL-FUQAHA, Ala; GUIZANI, Mohsen; MOHAMMADI, Mehdi; ALEDHARI, Mohammed; AYYASH, Moussa. Internet of Things : A Survey on Enabling Internet of Things : A Survey on Enabling Technologies , Protocols and Applications. *Ieee*, IEEE, v. 17, n. November, p. 2347–2376, 2015.

10 SONG, Jaeseung; KUNZ, Andreas; SCHMIDT, Mischa; SZCZYTOWSKI, Piotr. Connecting and managing M2M devices in the future internet. *Mobile Networks and Applications*, Springer, v. 19, n. 1, p. 4–17, 2014. ISSN 1383469X.

11 KHRIYENKO, Oleksiy; TERZIYAN, Vagan; KAIKOVA, Olena. User-assisted semantic interoperability in internet of things. In: *The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'12)*. [S.l.: s.n.], 2012. p. 104–110. ISBN 9781612082363.

12 XU, Li Da; HE, Wu; LI, Shancang. Internet of Things in Industries : A Survey Internet of Things in Industries : A Survey. *Industrial Informatics, IEEE Transactions on*, IEEE, v. 10, n. November, p. 2233–2243, 2014.

13 AMIR, Mohammed; HU, Y.Fun; PILLAI, Prashant; CHENG, Yongqiang; BIBIKS, Kirils. Interaction Models for Profiling Assets in an Extensible and Semantic WoT Framework. In: VDE. *Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on*. [S.l.], 2013. p. 1–5. ISBN 9783800735297. ISSN 21540225.

14 ThingSpeak. *Public Channels on ThingSpeak. ThingSpeak is the open IoT platform with MATLAB analytics.* 2017. Disponível em: <https://thingspeak.com/channels/public>.

15 RUTA, Michele; SCIOSCIA, Floriano. Framework and architecture for the Semantic Web of Things. In: *ICSC*. [S.l.: s.n.], 2014. p. 1–5.

16 PFISTERER, Dennis; ROMER, Kay; BIMSCHAS, Daniel; KLEINE, Oliver; MIETZ, Richard; TRUONG, Cuong; HASEMANN, Henning; KROLLER, Alexander; PAGEL, Max; HAUSWIRTH, Manfred; others. SPITFIRE: toward a semantic web of things. *Communications Magazine, IEEE*, v. 49, n. 11, p. 40–48, 2011. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6069708\%5Cnhttp://ieeexplore.ieee.org/Xplore/cookiedetectresponse.jsp?reload=true\%5Cnhttp://www.iti.uni-luebeck.de/fileadmin/user_upload/Paper/IEEEComMag.pdf>.

17 HUR, Kangho; CHUN, Sejin; JIN, Xiongnan; LEE, Kyong-Ho. Towards a Semantic Model for Automated Deployment of IoT Services across Platforms. In: IEEE. *2015 IEEE World Congress on Services*. 2015. p. 17–20. ISBN 978-1-4673-7275-6. ISSN 2378-3818. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7196498>.

18 BLACKSTOCK, Michael; LEA, Rodger. IoT interoperability: A hub-based approach. In: IEEE. *2014 International Conference on the Internet of Things (IOT)*. 2014. p. 79–84. ISBN 9781479951543. Disponível em: <http://dx.doi.org/10.1109/iot.2014.7030119>.

19 BIASE, Laisa C.C. De; CALCINA-CCORI, Pablo C.; SILVA, Flavio S.C.; ZUFFO, Marcelo K. The semantic Mediation for the Swarm: An adaptable and organic solution for the Internet of Things. In: IEEE. *2017 IEEE International Conference on Consumer Electronics, ICCE 2017*. [S.l.], 2017. p. 78–79. ISBN 9781509055449.

20 DAVOUDPOUR, Maryam; MASOUMI, Arman; SADEGHIAN, Alireza; RAHNAMA, Hossein. A formal ontology alignment for canthings (context aware network for the connected things). In: IEEE. *Proceedings - 2014 8th International Conference on Next Generation*

Mobile Applications, Services and Technologies, NGMAST 2014. [S.l.], 2014. p. 175–180. ISBN 9781479950737. ISSN 2161-2889.

21 XU, Yuan; ZHANG, Chunhong; JI, Yang. An upper-ontology-based approach for automatic construction of IOT ontology. *International Journal of Distributed Sensor Networks*, Hindawi Publishing Corporation, v. 2014, 2014. ISSN 15501477.

22 HEYVAERT, Pieter; DIMOU, Anastasia; VERBORGH, Ruben; MANNENS, Erik; WALLE, Rik Van De. Towards a uniform user interface for editing mapping definitions. In: *CEUR Workshop Proceedings*. [S.l.: s.n.], 2015. v. 1472. ISSN 16130073.

23 MING, Zhou; HONG, Fan; YAN, Ma. Semantic annotation method of IOT middleware. In: IEEE. *Proceedings of the 2013 International Conference on Intelligent Control and Information Processing, ICICIP 2013*. [S.l.], 2013. p. 495–498. ISBN 9781467362481.

24 YAQOOB, I.; AHMED, E.; HASHEM, I.; ABDELMUTTLIB, A.; GUIZANI, M. Internet of Things Architecture : Recent Advances, Taxonomy, Requirements, and Open Challenges. *IEEE Wireless Communications* •, IEEE, v. 24, n. June, p. 10–16, 2017.

25 SICARI, S.; RIZZARDI, A.; GRIECO, L. A.; COEN-PORISINI, A. Security, privacy and trust in Internet of things: The road ahead. *Computer Networks*, v. 76, p. 146–164, 2015. ISSN 13891286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128614003971>.

26 WEBER, Rolf H. Internet of Things - New security and privacy challenges. *Computer Law and Security Review*, v. 26, n. 1, p. 23–30, 1 2010. ISSN 02673649. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0267364909001939>.

27 GYRARD, Amelie; SERRANO, Martin; ATEMEZING, Ghislain A. Semantic web methodologies, best practices and ontology engineering applied to Internet of Things. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015. p. 412–417. ISBN 978-1-5090-0366-2. Disponível em: <http://ieeexplore.ieee.org/document/7389090/>.

28 KOVATSCH, Matthias; HASSAN, Yassin N.; MAYER, Simon. Practical semantics for the Internet of Things: Physical states, device mashups, and open questions. In: *2015 5th International Conference on the Internet of Things (IOT)*. IEEE, 2015. p. 54–61. ISBN 978-1-4673-8056-0. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7356548>.

29 BARNAGHI, Payam; WANG, Wei; HENSON, Cory; TAYLOR, Kerry. Semantics for the Internet of Things: Early Progress and Back to the Future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, v. 8, n. 1, p. 1–21, 2012. Disponível em: <https://www.igi-global.com/article/content/70584>.

30 YAN, Zheng; ZHANG, Peng; VASILAKOS, Athanasios V. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, v. 42, p. 120–134, 2014. ISSN 10958592. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804514000575>.

31 CHEN, Ing Ray; GUO, Jia; BAO, Fenye. Trust Management for SOA - based IoT and Its Application to Service Composition. *Journal of Machine Learning Research*, IEEE, v. 9, n. 3, p. 482–495, 2014.

32 KOTIS, Konstantinos; ATHANASAKIS, Iraklis; VOUROS, George A. Semantically enabling IoT trust to ensure and secure deployment of IoT entities. *International Journal of Internet of Things and Cyber-Assurance*, v. 1, n. 1, p. 3, 2018. ISSN 2059-7967. Disponível em: <http://www.inderscience.com/link.php?id=90158>.

33 NITTI, Michele; GIRAU, Roberto; ATZORI, Luigi. Trustworthiness management in the social internet of things. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 26, n. 5, p. 1253–1266, 5 2014. ISSN 10414347. Disponível em: <http://ieeexplore.ieee.org/document/6547148/>.

34 GUO, Jia; CHEN, Ing Ray; TSAI, Jeffrey J.P. A survey of trust computation models for service management in internet of things systems. *Computer Communications*, Elsevier, v. 97, p. 1–14, 2017. ISSN 01403664.

35 LEE, Jong Hyouk; KIM, Hyoungshick. Security and Privacy Challenges in the Internet of Things [Security and Privacy Matters]. *IEEE Consumer Electronics Magazine*, v. 6, n. 3, p. 134–136, 7 2017. ISSN 21622256. Disponível em: <http://ieeexplore.ieee.org/document/7948855/>.

36 BARRIO, Manuel Gértrudix; GARCÍA, Sergio Álvarez; FERNÁNDEZ, Mario Rajas. Open Data en aplicaciones móviles: nuevos modelos para la información de servicio. *Fonseca, Journal of Communication*, Ediciones Universidad de Salamanca (España), v. 12, n. 12, p. 117–131, 2016. ISSN 2172-9077.

37 BALESTRINI, M.; DIEZ, T.; MARSHALL, P.; GLUHAK, A.; ROGERS, Y. IoT Community Technologies: Leaving Users to Their Own Devices or Orchestration of Engagement? *EAI Endorsed Transactions on Internet of Things*, v. 1, n. 1, p. 150601, 2015. ISSN 2414-1399. Disponível em: <http://eudl.eu/doi/10.4108/eai.26-10-2015.150601>.

38 MISRA, Prasant; SIMMHAN, Yogesh; WARRIOR, Jay. Towards a Practical Architecture for the Next Generation Internet of Things. *arXiv preprint arXiv:1502.00797*, 2015. Disponível em: <http://arxiv.org/abs/1502.00797>.

39 MIORANDI, Daniele; SICARI, Sabrina; PELLEGRINI, Francesco De; CHLAMTAC, Imrich. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, v. 10, n. 7, p. 1497–1516, 9 2012. ISSN 15708705. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1570870512000674>.

40 KITCHENHAM, Barbara. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. TR/SE-0401, p. 28, 2004. ISSN 13537776. Disponível em: <http://csnotes.upm.edu.my/kelasmaya/pgkm20910.nsf/0/715071a8011d4c2f482577a700386d3a/$FILE/10.1.1.122.3308[1].pdf\%5Cnhttp://tests-zingarelli.googlecode.com/svn-history/r336/trunk/2-Disciplinas/MetodPesquisa/kitchenham_2004.pdf>.

41 WEBSTER, Jane; WATSON, Richard T. *Analyzing the Past to Prepare for the Future : Writing a Literature Review ANALYZING THE PAST TO PREPARE FOR THE FUTURE : WRITING A*. [S.l.]: JSTOR, 2016. xiii–xxiii p.

42 WOHLIN, Claes. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: ACM. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*. 2014. p. 1–10. ISBN 9781450324762. ISSN 09505849. Disponível em: <http://dl.acm.org/citation.cfm?doid=2601248.2601268>.

43 WANG, Wei; DE, Suparna; CASSAR, Gilbert; MOESSNER, Klaus. An experimental study on geospatial indexing for sensor service discovery. *Expert Systems with Applications*, Pergamon, v. 42, n. 7, p. 3528–3538, 5 2015. ISSN 09574174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S095741741400757X>.

44 PERERA, Charith; ZASLAVSKY, Arkady; CHRISTEN, Peter; COMPTON, Michael; GEORGAKOPOULOS, Dimitrios. Context-aware sensor search, selection and ranking model for internet of things middleware. In: *Proceedings - IEEE International Conference on Mobile Data Management*. IEEE, 2013. v. 1, p. 314–322. ISBN 978-0-7695-4973-6. ISSN 15516245. Disponível em: <http://ieeexplore.ieee.org/document/6569153/>.

45 LE-PHUOC, Danh; QUOC, Hoan; PARREIRA, Josiane Xavier; HAUSWIRTH, Manfred. The linked sensor middleware–connecting the real world and the semantic web. *Semantic Web Challenge*, n. April 2005, p. 1–8, 2011.

46 CoRE Working Group. *CoRE Resource Directory*. 2015. 1–28 p. Disponível em: <https://tools.ietf.org/html/draft-ietf-core-resource-directory-04>.

47 SANCHEZ, Luis; MUÑOZ, Luis; GALACHE, Jose Antonio; SOTRES, Pablo; SANTANA, Juan R.; GUTIERREZ, Veronica; RAMDHANY, Rajiv; GLUHAK, Alex; KRCO, Srdjan; THEODORIDIS, Evangelos; PFISTERER, Dennis. SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks*, Elsevier B.V., v. 61, p. 217–238, 3 2014. ISSN 13891286. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S1389128613004337>.

48 BONINO, Dario; PASTRONE, Claudio; SPIRITO, Maurizio. Towards a Federation of Smart City Services. *International Conference on Recent Advances in Computer Systems (RACS 2015)*, n. Racs 2015, p. 163–168, 2015.

49 KILJANDER, J; D'ELIA, A; MORANDI, F; HYTTINEN, P; TAKALO-MATTILA, J; YLISAUKKO-OJA, A; SOININEN, J.-P.; CINOTTI, T S. Semantic Interoperability Architecture for Pervasive Computing and Internet of Things. *Access, IEEE*, IEEE, v. 2, p. 856–873, 2014. ISSN 2169-3536.

50 JARA, Antonio J.; LOPEZ, Pablo; FERNANDEZ, David; CASTILLO, Jose F.; ZAMORA, Miguel A.; SKARMETA, Antonio F. Mobile digcovery: Discovering and interacting with the world through the Internet of things. *Personal and Ubiquitous Computing*, Springer-Verlag, v. 18, n. 2, p. 323–338, 2014. ISSN 16174909.

51 CIRANI, Simone; DAVOLI, Luca; FERRARI, Giorgio; LÉONE, Rémy; MEDAGLIANI, Paolo; PICONE, Marco; VELTRI, Luca. A scalable and self-configuring architecture for service discovery in the internet of things. *Internet of Things Journal, IEEE*, IEEE, v. 1, n. 5, p. 508–521, 2014.

52 HACHEM, Sara; TEIXEIRA, Thiago; ISSARNY, Valérie. Ontologies for the internet of things. In: ACM. *Proceedings of the 8th Middleware Doctoral Symposium on - MDS '11*. New York, NY, USA: ACM, 2011. (MDS '11), p. 1–6. ISBN 9781450310727. ISSN 9781450310727. Disponível em: <http://dl.acm.org/citation.cfm?doid=2093190.2093193>.

53 BARNAGHI, Payam; PRESSER, Mirko. Publishing linked sensor data. In: CEUR-WS. ORG. *CEUR Workshop Proceedings*. [S.l.], 2010. v. 668, p. 1–16. ISBN 9781424466221. ISSN 16130073.

54 KOTIS, Konstantinos; KATASONOV, Artem. Semantic interoperability on the Web of things: The semantic smart gateway framework. In: IEEE. *Proceedings - 2012 6th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2012*. [S.l.], 2012. p. 630–635. ISBN 9780769546872. ISSN 19473532.

55 CHEN, X. Y.; LI, G. Y. Semantic collaboration and sensing as a service in semantic web of things. *International Journal on Smart Sensing and Intelligent Systems*, v. 9, n. 2, p. 998–1028, 2016. ISSN 11785608.

56 BRÖRING, Arne; MAUÉ, Patrick; JANOWICZ, Krzysztof; NÜST, Daniel; MALEWSKI, Christian. Semantically-enabled sensor Plug & Play for the Sensor Web. *Sensors*, Molecular Diversity Preservation International, v. 11, n. 8, p. 7568–7605, 2011. ISSN 14248220.

57 BONOMI, Flavio; MILITO, Rodolfo; ZHU, Jiang; ADDEPALLI, Sateesh. Fog computing and its role in the Internet of things. In: ACM. *ACM MCC workshop on Mobile cloud computing*. [S.l.], 2012. p. 13–16.

58 SEHGAL, V K; PATRICK, A; SONI, A; RAJPUT, L. Smart human security framework using internet of things, Cloud and fog computing. In: *Advances in Intelligent Systems and Computing*. Springer, 2015. v. 321, p. 251–263. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84921403194&partnerID=40&md5=84773c19bee7d3b11cb7cc712e5cd675>.

59 LEE, Wangbong; NAM, Kidong; ROH, Hak-Gyun; KIM, Sang-Ha. A gateway based fog computing architecture for wireless sensors and actuator networks. In: IEEE. *2016 18th International Conference on Advanced Communication Technology (ICACT)*. 2016. p. 1–1. ISBN 978-8-9968-6506-3. Disponível em: <http://ieeexplore.ieee.org/document/7423331/>.

60 GIORDANO, Andrea; SPEZZANO, Giandomenico; VINCI, Andrea. Smart agents and fog computing for smart city applications. In: SPRINGER. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.], 2016. v. 9704, p. 137–146. ISBN 9783319395944. ISSN 16113349.

61 GYRARD, Amelie; DATTA, Soumya Kanti; BONNET, Christian; BOUDAOUD, Karima. Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation. In: IEEE. *Proceedings - 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, OBD 2015*. [S.l.], 2015. p. 9–16. ISBN 9781467381031.

62 XIAO, Guangyi; GUO, Jingzhi; MEMBER, Senior; GONG, Zhiguo. User Interoperability With Heterogeneous IoT. *IEEE Transactions on Industrial Informatics*, IEEE, v. 10, n. 2, p. 1486–1496, 2014.

63 MALIK, Kaleem Razzaq; AHMAD, Tauqir; FARHAN, Muhammad; ULLAH, Farhan; AMJAD, Kashif; KHALID, Shehzad. Multiagent Semantical Annotation Enhancement Model for IoT-Based Energy-Aware Data. *International Journal of Distributed Sensor Networks*, v. 2016, 2016. ISSN 15501477.

64 CHEN, Ing Ray; BAO, Fenye; GUO, Jia. Trust-Based Service Management for Social Internet of Things Systems. *IEEE Transactions on Dependable and Secure Computing*, IEEE, v. 13, n. 6, p. 684–696, 11 2016. ISSN 15455971. Disponível em: <http://ieeexplore.ieee.org/document/7097037/>.

65 SAIED, Yosra Ben; OLIVEREAU, Alexis; ZEGHLACHE, Djamal; LAURENT, Maryline. Trust management system design for the Internet of Things: A context-aware and multi-service approach. *Computers and Security*, Elsevier, v. 39, n. PART B, p. 351–365, 11 2013. ISSN 01674048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404813001302>.

66 NAMAL, Suneth; GAMAARACHCHI, Hasindu; LEE, Gyu Myoung; UM, Tai Won. Autonomic trust management in cloud-based and highly dynamic IoT applications. In: IEEE. *Proceedings of the 2015 ITU Kaleidoscope: Trust in the Information Society, K-2015 - Academic Conference*. [S.l.], 2016. p. 1–8. ISBN 9789261158217. ISSN 18498558.

67 MENDOZA, Carolina V L; KLEINSCHMIDT, João H. Mitigating on-off attacks in the internet of things using a distributed trust management scheme. *International Journal of Distributed Sensor Networks*, Hindawi Publishing Corporation, v. 2015, 2015. ISSN 15501477. Disponível em: <http://journals.sagepub.com/doi/pdf/10.1155/2015/859731>.

68 LI, Wenjia; SONG, Houbing; ZENG, Feng. Policy-based Secure and Trustworthy Sensing for Internet of Things in Smart Cities. *IEEE Internet of Things Journal*, p. 1–1, 2017. ISSN 23274662. Disponível em: <http://ieeexplore.ieee.org/document/7959573/>.

69 YU, Yang; JIA, Ziyan; TAO, Weige; XUE, Bo; LEE, Changhoon. An Efficient Trust Evaluation Scheme for Node Behavior Detection in the Internet of Things. *Wireless Personal Communications*, Springer US, v. 93, n. 2, p. 571–587, 3 2017. ISSN 1572834X. Disponível em: <http://link.springer.com/10.1007/s11277-016-3802-y>.

70 BOUSTANIFAR, Fariba; MOVAHEDI, Zeinab. A Trust-Based Offloading for Mobile M2M Communications. In: *Proceedings - 13th IEEE International Conference on Ubiquitous Intelligence and Computing, 13th IEEE International Conference on Advanced and Trusted Computing, 16th IEEE International Conference on Scalable Computing and Communications, IEEE Internationa*. IEEE, 2017. p. 1139–1143. ISBN 9781509027705. Disponível em: <http://ieeexplore.ieee.org/document/7816971/>.

71 HELLAOUI, Hamed; BOUABDALLAH, Abdelmadjid; KOUDIL, Mouloud. TAS-IoT: Trust-Based Adaptive Security in the IoT. In: *Proceedings - Conference on Local Computer*

Networks, LCN. IEEE, 2016. p. 599–602. ISBN 9781509020546. ISSN 0742-1303. Disponível em: <http://ieeexplore.ieee.org/document/7796850/>.

72 ABDERRAHIM, Oumaima Ben; ELHEDHILI, Mohamed Houcine; SAIDANE, Leila. CTMS-SIOT: A context-based trust management system for the social Internet of Things. In: *2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017*. IEEE, 2017. p. 1903–1908. ISBN 9781509043729. Disponível em: <http://ieeexplore.ieee.org/document/7986378/>.

73 ZHANG, Tong; YAN, Lisha; YANG, Yuan. *Trust evaluation method for clustered wireless sensor networks based on cloud model*. Springer US, 2016. 1–21 p. Disponível em: <http://link.springer.com/10.1007/s11276-016-1368-y>.

74 SONY, Swathy M; SASI, Swapna B. On - off attack management based on trust. In: *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*. IEEE, 2016. p. 1–4. ISBN 978-1-5090-4556-3. Disponível em: <http://ieeexplore.ieee.org/document/7916760/>.

75 ZHOU, L; CHAO, HC. Multimedia traffic security architecture for the internet of things. *IEEE Network*, IEEE, v. 25, n. 3, p. 35–40, 2011. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5772059>.

76 KOTHMAYR, Thomas; SCHMITT, Corinna; HU, Wen; BRÜNIG, Michael; CARLE, Georg. DTLS based security and two-way authentication for the Internet of Things. *Ad Hoc Networks*, Elsevier, v. 11, n. 8, p. 2710–2723, 2013. ISSN 15708705.

77 ŢIPLEA, Ferucio Laurenţiu. A lightweight authentication protocol for RFID. In: IEEE. *Communications in Computer and Information Science*. [S.l.], 2014. v. 448 CCIS, p. 110–121. ISBN 9783662448922. ISSN 18650929.

78 TURKANOVIĆ, Muhamed; BRUMEN, Boštjan; HÖLBL, Marko. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Networks*, Elsevier, v. 20, p. 96–112, 2014.

79 DU, Wenliang; DENG, Jing; HAN, Y S; K, Varshney P. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. *Proceedings of the 10th ACM conference on Computer and communications security*, ACM, v. 8, n. 2, p. 42–51, 2003. Disponível em: <http://portal.acm.org/citation.cfm?id=948118>.

80 BAO, Fenye; CHEN, Ing-Ray. Dynamic trust management for internet of things applications. In: ACM. *Proceedings of the 2012 international workshop on Self-aware internet of things - Self-IoT '12*. 2012. p. 1. ISBN 9781450317535. Disponível em: <http://dl.acm.org/citation.cfm?doid=2378023.2378025>.

81 HERNÁNDEZ-RAMOS, José Luis; BERNAL, Jorge; JOSE, Bernabe; HERNANDEZ, Luis; ANTONIO, Ramos. TACIoT : Multidimensional Trust-aware Access Control system for the Internet of Things. *Soft Computing*, Springer, v. 20, n. August, p. 1763–1779, 2015.

82 RAFEY, Sherif Emad Abdel; ABDEL-HAMID, Ayman; EL-NASR, Mohamad Abou. CBSTM-IoT: Context-based social trust model for the Internet of Things. In: IEEE. *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*. 2016. p. 1–8. ISBN 978-1-5090-1743-0. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7496623>.

83 PEDREGOSA, Fabian; VAROQUAUX, Gaël; GRAMFORT, Alexandre; MICHEL, Vincent; THIRION, Bertrand; GRISEL, Olivier; BLONDEL, Mathieu; PRETTENHOFER, Peter; WEISS, Ron; DUBOURG, Vincent et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, v. 12, n. Oct, p. 2825–2830, 2011.

84 CHANG, Chih-Chung; LIN, Chih-Jen. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, ACM, v. 2, n. 3, p. 1–27, 4 2011. Disponível em: <http://dl.acm.org/citation.cfm?doid=1961189.1961199>.

85 LI, Zhiyong; TIAN, Ye; LI, Ke; YANG, Wei. Reject inference in credit scoring using Support Vector Machines. *Expert Systems with Applications*, Elsevier, v. 74, p. 105–114, 2017. ISSN 09574174. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S095741741730012X>.

86 KUMAR, Santosh; GAO, Xiaoying; WELCH, Ian; MANSOORI, Masood. A Machine Learning Based Web Spam Filtering Approach. In: IEEE. *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. 2016. p. 973–980. ISBN 978-1-5090-1858-1. ISSN 1550445X. Disponível em: <http://ieeexplore.ieee.org/document/7474194/>.

87 LIN, Wen-Hui; WANG, Ping; TSAI, Chen-Fang. Face recognition using support vector model classifier for user authentication. *Electronic Commerce Research and Applications*, Elsevier, v. 18, p. 71–82, 2016. ISSN 15674223. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S1567422316300011>.

88 BAN, Tao; TAKAHASHI, Takeshi; GUO, Shanqing; INOUE, Daisuke; NAKAO, Koji. Integration of Multi-modal Features for Android Malware Detection Using Linear SVM. In: IEEE. *Proceedings - 11th Asia Joint Conference on Information Security, AsiaJCIS 2016*. [S.l.], 2016. p. 141–146. ISBN 9781509022854.

89 WANG, Ping; CHAO, Kuo-Ming; LIN, Hsiao-Chung; LIN, Wen-Hui; LO, Chi-Chun. An Efficient Flow Control Approach for SDN-Based Network Threat Detection and Migration Using Support Vector Machine. In: IEEE. *2016 IEEE 13th International Conference on e-Business Engineering (ICEBE)*. 2016. p. 56–63. ISBN 978-1-5090-6119-8. Disponível em: <http://ieeexplore.ieee.org/document/7809901/>.

90 SANTANDER, City of. *SmartSantander DataSet*. 2016. Disponível em: <https://data.lab.fiware.org/dataset?q=santander&sort=score+desc\%2C+metadata_modified+desc&tags=santander>.

91 THERMI, City of. *Thermi DataSet*. 2016. Disponível em: <http://www.people-project.eu/portal/index.php?option=com-content\%5C&view=article\%5C&id=60:about-thermi\%5C&catid=44\%5C&Itemid=18>.

92 CITIZEN, Team Smart. *Smart Citizen Devices*. 2015. Disponível em: <https://smartcitizen.me/devices>.

93 TRENTINO, City of. *Trentino DataSet*. 2016. Disponível em: <https://data.lab.fiware.org/organization/trentino>.

94 AARHUS, City of. *Weather Data for the City of Aarhus in Denmark*. 2016. Disponível em: <http://iot.ee.surrey.ac.uk:8080/datasets.html>.

95 FIWARE. *FIWARE Lab is a working instance of FIWARE available for experimentation*. 2017. Disponível em: <https://cloud.lab.fiware.org>.

96 SAITTA, Lorenza; NERI, Filippo. Learning in the "real world". *Machine Learning*, Springer, v. 30, n. 2-3, p. 133–163, 1998.

97 SEHGAL, Anuj. Using the Contiki Cooja Simulator. *Computer Science, Jacobs University Bremen Campus Ring*, v. 1, p. 28759, 2013. Disponível em: <http://cnds.eecs.jacobs-university.de/courses/iotlab-2013/cooja.pdf>.

98 JOACHIMS, Thorsten. *Text categorization with Support Vector Machines: Learning with many relevant features*. Springer, 1998. 137–142 p. ISSN 03436993. ISBN 978-3-540-64417-0, 978-3-540-69781-7. Disponível em: <http://link.springer.com/10.1007/BFb0026683>.

99 TRAN, Thanh N.; DRAB, Klaudia; DASZYKOWSKI, Michal. Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, Elsevier, v. 120, p. 92–96, 1 2013. ISSN 0169-7439. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169743912002249>.

Apêndice A

Código fonte da aplicação para acesso aos dados de uma *SmartCity* no FI-Ware

```
1  ######################################################
2  ## Carregamento das bibliotecas padrao utilizadas ##
3  ######################################################
4
5  import requests, json, getpass
6
7
8  ########################################
9  ## Conexao com a plataforma FI-Ware ##
10 ########################################
11
12 TOKENS_URL = "http://cloud.lab.fi-ware.org:4730/v2.0/tokens"
13
14 USER = input("FIWARE Lab Username: ")
15 PASSWORD = getpass.getpass("FIWARE Lab Password: ")
16 PAYLOAD = "{\"auth\": {\"passwordCredentials\": {\"username\":\""+USER+"\",
       \"password\":\""+PASSWORD+"\"}}}"
17 HEADERS =  {'content-type': 'application/json'}
18 URL = TOKENS_URL
19
20 RESP = requests.post(URL, data=PAYLOAD, headers=HEADERS)
21 print()
```

```
22 print ("FIWARE Oauth2.0 Token: "+RESP.json()["access"]["token"]["id"])
23 print()
24 print ("Token expires: "+RESP.json()["access"]["token"]["expires"])
25 print()
26
27 TOKEN = "Ewif2KqvL6nvqubMNHrOrftoCLnUS8"
28
29
30 ################################################
31 ## Leitura de valores da SmartCity Santander ##
32 ################################################
33
34 SERVER = "http://orion.lab.fiware.org:1026/v2/entities/urn:smartsantander:
       testbed:3332"
35
36 HEADERS = {'accept': 'application/json','X-Auth-Token' : TOKEN}
37 DATA = requests.get(SERVER, headers=HEADERS)
38
39 print()
40 print ("* Status Code: "+str(DATA.status_code))
41 print ("* Response: ")
42 print (DATA.json())
43
44 DATA.json().keys()
45 DATA.json()['temperature']['value']
46
47 ########################
48 ## Fim do aplicativo ##
49 ########################
```

Apêndice B

Código fonte de implementação da Janela Elástica Dinâmica

```
1
2  ######################################################
3  ## Carregamento das bibliotecas padrao utilizadas ##
4  ######################################################
5
6  import sys
7  import matplotlib.pyplot as plt
8  import matplotlib.font_manager
9  import matplotlib.patches as mpatches
10 import pandas as pd
11 from sklearn.manifold import TSNE
12 from sklearn.externals import joblib
13 from scipy import sparse
14 matplotlib inline
15
16 from sklearn.feature_extraction.text import HashingVectorizer
17 from sklearn import svm
18 from sklearn import metrics
19 import numpy as np
20
21 from sklearn.model_selection import train_test_split
22 from sklearn.model_selection import ShuffleSplit
```

```
23 from sklearn.model_selection import learning_curve
24
25 import time
26
27
28 ##############################################################
29 ## Funcoes auxiliares para a visualizacao dos resultados ##
30 ##############################################################
31
32 def plot(vetor,feat,nomes,titulo,p):
33     tsne = TSNE(n_components=2, perplexity=p, random_state=0)
34     np.set_printoptions(suppress=True)
35     vetor = tsne.fit_transform(vetor)
36
37     plt.figure(figsize=(20, 20))
38
39     color = []
40     for i in feat:
41         color.append(lista_tipos[i][2])
42
43     plt.scatter(vetor[:, 0], vetor[:, 1], s=200, c=color)
44
45     #out_patch = mpatches.Patch(color='m', label='Outlier')
46     temp_patch = mpatches.Patch(color='b', label='Temperature')
47     hum_patch = mpatches.Patch(color='g', label='Humidity')
48     light_patch = mpatches.Patch(color='y', label='Light')
49     press_patch = mpatches.Patch(color='r', label='Pressure')
50     sound_patch = mpatches.Patch(color='c', label='Sound')
51     #wrong_patch = mpatches.Patch(color='k', label='Incorrect')
52     #plt.legend(handles=[out_patch,temp_patch,hum_patch,light_patch,
           press_patch,sound_patch,wrong_patch])
53     plt.legend(handles=[temp_patch,hum_patch,light_patch,press_patch,
           sound_patch])
54
55
56     for row_id in range(0, len(nomes)):
57         target_word = nomes.iloc[row_id]
```

```
58          x = vetor[row_id, 0]
59          y = vetor[row_id, 1]
60          plt.annotate(target_word, (x,y))
61
62      plt.title(titulo)
63      plt.show()
64
65
66  def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None, n_jobs=1,
        train_sizes=np.linspace(.1, 1.0, 5)):
67
68      plt.figure()
69      plt.title(title)
70      if ylim is not None:
71          plt.ylim(*ylim)
72      plt.xlabel("Training examples")
73      plt.ylabel("Score")
74      train_sizes, train_scores, test_scores = learning_curve(estimator, X, y,
            cv=cv, n_jobs=n_jobs, train_sizes=train_sizes)
75      train_scores_mean = np.mean(train_scores, axis=1)
76      train_scores_std = np.std(train_scores, axis=1)
77      test_scores_mean = np.mean(test_scores, axis=1)
78      test_scores_std = np.std(test_scores, axis=1)
79      plt.grid()
80
81      plt.fill_between(train_sizes, train_scores_mean - train_scores_std,
            train_scores_mean + train_scores_std, alpha=0.1, color="r")
82      plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
            test_scores_mean + test_scores_std, alpha=0.1, color="g")
83      plt.plot(train_sizes, train_scores_mean, 'o-', color="r", label="Training
            score")
84      plt.plot(train_sizes, test_scores_mean, 'o-', color="g", label="Cross-
            validation score")
85
86      plt.legend(loc="best")
87      plt.show()
88
```

```
89
90 ########################################################################
91 ## Inicializacao de variaveis e Carregamento de bases para avaliacao ##
92 ########################################################################
93
94 lista_tipos = [['0','Outlier','m'],['1','Temperature','b'],['2','Humidity','g
       '],['3','Light','y'],['4','Pressure','r'],['5','Sound','c'],['6','
       Incorrect','k']]
95
96 qtd_sensores = 2000
97
98
99 ##############################################
100 ## Pre-processamento das bases de avaliacao ##
101 ##############################################
102
103 sensores = pd.read_csv('things_speak_dataset_200000_v2.csv', names = ["Sensor
       ", "Read", "Type", "TypeName"])
104 #sensores = sensores.drop_duplicates(['Sensor'])
105 #sensores = sensores.drop_duplicates(['Read'])
106 sensores = sensores.dropna(how='any')
107
108 sensores = sensores.convert_objects(convert_numeric=True).dropna()
109 #warnings.filterwarnings(action='ignore', category=FutureWarning, module='
       convert_objects')
110 #sensores = pd.to_numeric(sensores['Read']).dropna()
111 sensores = sensores[sensores.Type != 0]
112 sensores = sensores[:qtd_sensores]
113
114 X_train, X_test = train_test_split(sensores, test_size=0.25)
115
116 vectorizer = HashingVectorizer(analyzer='char_wb')
117 X_Train_Dict = X_train['Sensor']
118 X_Train_Dict = vectorizer.transform(X_Train_Dict)
119 X_train_read = sparse.csr_matrix(np.array(X_train['Read']))
120 X_Train_Dict = sparse.hstack([X_Train_Dict, X_train_read.T])
121
```

```
122
123 ####################################
124 ## Inicializacao do Classificador ##
125 ####################################
126
127 inicio = time.time()
128 #print("Tempo de treinamento: "+str('{0:.2f}'.format(time.time()-inicio))+'s
        ')
129 clf = svm.LinearSVC(dual=True)
130
131 clf.fit(X_Train_Dict, X_train['Type'])
132 #X_train_linear = clf.decision_function(X_Train_Dict)
133
134 print("Tempo para treino: "+str('{0:.2f}'.format(time.time()-inicio))+'s')
135 #print(X_train_linear[:5])
136 #plot(X_train_linear, X_train['Type'], X_train['Sensor'],"Names and Reads
        trained",50)
137
138 estimator = clf
139 title = "Learning Curves: " + str(estimator)
140 cv = ShuffleSplit(n_splits=10, test_size=0.2, random_state=0)
141
142 plot_learning_curve(estimator, title, X_Train_Dict, X_train['Type'], ylim
        =(0.7, 1.01), cv=cv, n_jobs=4)
143
144
145 ########################################################################
146 ## Implementacao da Janela Elastica Dinamica, conforme Algoritimo 1 ##
147 ########################################################################
148
149 y_pred = []
150 y_true = []
151 things = {}
152 things_slide_window = {}
153
154 dataset = test_broken_dataset
155 sw_alpha = float(-400)
```

```
156 sw_init = 0
157
158 def slide_window(last_sw,dec_func,alpha_value):
159     return (last_sw - (dec_func))
160
161 for i in range(len(dataset)):
162     thing_id = int(dataset[i][0])
163     pred = int(clf.predict(dataset[i][1]))
164     defu = float(clf.decision_function(dataset[i][1]))
165
166 for i,j,k in dataset:
167     thing_id = i
168     pred = int(clf.predict(j))
169     defu = float(clf.decision_function(j))
170
171     if defu < sw_alpha:
172             trust_score = False #low
173     else:
174             trust_score = True #high
175
176     print(thing_id,things,thing_id in things)
177
178     if (thing_id in things) and (things_slide_window[thing_id] > 0):
179
180         last_pred = int(things[thing_id])
181         last_sw = things_slide_window[thing_id]
182         things_slide_window[thing_id] = slide_window(last_sw,defu,sw_alpha)
183
184         if ((pred == -1 and last_pred == -1) and (trust_score == False)):
185             things[thing_id] = 0
186
187         elif pred != last_pred and trust_score == True:
188             things[thing_id] = -1
189
190         elif pred != last_pred and trust_score == False:
191             things[thing_id] = last_pred
192
```

```
193     if things[thing_id] !=  k:print(i,"sw:",things[thing_id],"last:",
            last_pred,"pred:",pred,trust_score,defu,things_slide_window[thing_id],
            case)
194
195     else:
196         things[thing_id] = int(pred)
197
198         if pred == -1 and trust_score == False: # broken
199             things[thing_id] = 0
200
201     things_slide_window[thing_id] = slide_window(sw_init,defu,sw_alpha)
202
203     y_pred.append(things[thing_id])
204     y_true.append(k)
205
206 print("precision_score: {}".format(metrics.precision_score(y_true, y_pred,
        average='weighted')))
207 print("recall_score: {}".format(metrics.recall_score(y_true, y_pred, average=
        'weighted')))
208 print("f1_score: {}".format(metrics.f1_score(y_true, y_pred, average='
        weighted')))
209 print(things)
210
211 ids = []
212 y_pred = []
213 y_true = []
214
215 for i in things:
216     ids.append(i)
217     y_pred.append(things[i])
218
219     if i <= 40:
220         y_true.append(1)
221     elif i > 40 and i <= 45:
222         y_true.append(-1)
223     else:
224         y_true.append(0)
```

```
225
226 ########################
227 ## Fim do aplicativo ##
228 ########################
```