UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

Eduardo Devidson Costa Bezerra

Tratamento de Incertezas no Processamento de Eventos Complexos para Internet das Coisas através da Teoria das Evidências de Dempster-Shafer

São Luís
2021

Eduardo Devidson Costa Bezerra

Tratamento de Incertezas no Processamento de Eventos Complexos para Internet das Coisas através da Teoria das Evidências de Dempster-Shafer

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como requisito parcial para a obtenção do grau de DOUTOR em Engenharia de Eletricidade com área de concentração em Ciência da Computação.

Orientador: Francisco José da Silva e Silva

Doutor - UFMA

São Luís

2021

Devidson Costa Bezerra, Eduardo

Tratamento de Incertezas no Processamento de Eventos Complexos para Internet das Coisas através da Teoria das Evidências de Dempster-Shafer / Eduardo Devidson Costa Bezerra. – São Luís, 2021.

139 f.

Orientador: Francisco José da Silva e Silva.

Impresso por computador (fotocópia).

Tese (Doutorado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia de Eletricidade. São Luís, 2021.

1. Internet das Coisas. 2. Incerteza no Processamento de Eventos Complexos. 3. Teoria de Dempster-Shafer. I. José da Silva e Silva, Francisco, orient. II. Título.

CDU

Eduardo Devidson Costa Bezerra

Tratamento de Incertezas no Processamento de Eventos Complexos para Internet das Coisas através da Teoria das Evidências de Dempster-Shafer

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Eduardo Devidson Costa Bezerra e aprovada pela comissão examinadora.

Aprovada em 23 de Novembro de 2021

BANCA EXAMINADORA

Francisco José da Silva e Silva (orientador)

Doutor - UFMA

Denivaldo Cicero Pavão Lopes

Doutor - UFMA

Fábio Moreira Costa

Doutor - UFG

Ginalber Luiz de Oliveira Serra

Doutor - IFMA

Luciano Reis Coutinho

Doutor - UFMA

Omar Andres Carmona Cortes

Doutor - IFMA

Dedico esta glória a Deus por iluminar meus pensamentos nos momentos mais difíceis, nos momentos em que mais precisei...

Agradecimentos

Primeiramente agradeço a Deus por me manter sereno em meio a tensão, firme em meio as dificuldades, perseverante em meio aos erros, mais forte e grato em meio aos acertos.

Agradeço ao professor Francisco Silva pelos ensinamentos e orientação até a conclusão deste trabalho. Agradeço por sua competência e conselhos nas horas em que nada parecia estar dando certo. Agradeço por me aceitar como seu aluno de doutorado, ser justo e confiar em mim nos momentos em que mais precisei.

Agradeço ao professor Luciano Coutinho por dedicar parte do seu tempo na construção desse trabalho. Agradeço por agregar ao trabalho o seu ponto de vista relevante através de boas e longas discussões.

Agradeço aos alunos e professores do Laboratório de Sistemas Distribuídos Inteligentes (LSDi/UFMA) pela companhia, discussões e apoio todos esses anos. Em especial, agradeço ao Ariel Teles pela parceria fundamental na construção e revisão do artigo científico.

Agradeço aos membros da banca examinadora, por aceitarem a missão de avaliar este trabalho.

Agradeço ao Programa de Pós-Graduação em Engenharia de Eletricidade (PPGEE/UFMA) e ao secretário administrativo Alcides, sempre disponível para atender e ajudar os alunos.

Agradeço aos meus colegas de trabalho da Superintendência de Tecnologia da Informação (STI/UFMA) pelo apoio durante todos esses anos de trabalho e estudo, em especial, Marcos Lauande, Neto e Anilton pela parceria. Agradecimento também especial ao Jorge Lucas pelo apoio técnico e discussões na elaboração do *framework*.

Agradeço aos meus pais Maria de Jesus e Reinaldo, meus irmãos, meus filhos Enzo e Maitê, que sempre entenderam o meu afastamento. Mesmo nas minhas ausências, eles sempre seguiram me apoiando de maneira irrestrita.

Agradeço especialmente à minha esposa Laura, por sua paciência nos meus momentos de reclusão. Nos dias ruins, ela sempre esteve ao meu lado me motivando a continuar. Por ela e meus filhos sempre me mantive resiliente para superar os desafios dessa difícil jornada e chegar até aqui.

In memoriam, gostaria de fazer um agradecimento especial ao professor Zair Abdelouahab.

Já é um vencedor quem sabe a dor de uma derrota enfrentar.

E a quem Deus prometeu nunca faltou, na hora certa o bom Deus dará...

Rodrigo Leite / Sergio Serafim

Resumo

A Internet das Coisas (IoT) surgiu a partir da proliferação de dispositivos móveis e objetos conectados, resultando na aquisição de fluxos de eventos periódicos de diferentes dispositivos e sensores. No entanto, tais sensores e dispositivos podem estar defeituosos, afetados por falhas, má calibração, produzindo em aplicações de IoT dados imprecisos e fluxos de eventos frequentemente não confiáveis. Em IoT, uma das técnicas mais proeminentes para análise de fluxos de eventos é o Processamento de Eventos Complexos (CEP). A incerteza em processamento de eventos é usualmente observada nos eventos primitivos (ex., leituras de sensores) e na sua propagação para os eventos complexos derivados (ex., situações de alto nível). Este trabalho investiga a identificação e o tratamento de incerteza em aplicações de IoT baseadas em CEP. Nesse viés, apresentamos a proposta *DST-CEP* que é uma abordagem que utiliza a Teoria Dempster-Shafer (TDS) para tratar incertezas. Por meio do uso da TDS, a solução pode combinar dados de sensores não confiáveis em situações conflitantes e detectar corretamente os resultados. A abordagem *DST-CEP* propõe um modelo arquitetural para tratar a incerteza nos eventos e sua propagação para os eventos complexos. Considerando o modelo arquitetural proposto, um framework *DST-CEP* foi implementado. Um estudo de caso é descrito e aplicado, utilizando a solução *DST-CEP* em um sistema multi-sensor de detecção de incêndio. A solução foi submetida para experimentos com um conjunto de dados de sensores reais e foi avaliada usando métricas de desempenho bem conhecidas. A solução alcança resultados promissores em relação às métricas *Accuracy, Precision, Recall, F-measure* e Curva ROC, mesmo combinando leituras conflitantes de sensores. *DST-CEP* demonstra ser adequada e flexível para lidar com as questões de incertezas levantadas nesta pesquisa.

Palavras-chaves: Internet das Coisas; Processamento de Eventos Complexos; Incerteza; Teoria de Dempster-Shafer.

Abstract

The Internet of Things (IoT) has emerged from the proliferation of mobile devices and objects connected, so resulting in the acquisition of periodic event flows from different devices and sensors. However, such sensors and devices can be faulty or affected by failures, have poor calibration, producing in IoT applications inaccurate data and frequently unreliable event flows. In IoT, a prominent technique for analyzing event flows is Complex Event Processing (CEP). Uncertainty in event processing is usually observed in primitive events (i.e., sensor readings) and rules that derive complex events (i.e., high-level situations). In this study, we investigate the identification and treatment of uncertainty in CEP-based IoT applications. In this vein, we propose the *DST-CEP*, an approach that uses the Dempster-Shafer Theory to treat uncertainties. By using this theory, our solution can combine unreliable sensor data in conflicting situations and detect correct results. *DST-CEP* has an architectural model for treating uncertainty in events and its propagation to comlex event. Considering the proposed architectural model, a *DST-CEP* framework was implemented. We describe a case study using the proposed approach in a multi-sensor fire outbreak detection system. We submit our solution to experiments with a real sensor dataset, and evaluate it using well-known performance metrics. The solution achieves promising results regarding Accuracy, Precision, Recall, F-measure, and ROC Curve, even when combining conflicting sensor readings. *DST-CEP* demonstrates to be suitable and flexible to deal with questions of uncertainty raised in this research..

Keywords: Internet of Things; Complex Event Processing; Uncertainty; Dempster-Shafer Theory.

Lista de Figuras

Lista de Tabelas

Sumário

1 Introdução

1.1 Contexto Geral

Este trabalho investiga a identificação e o tratamento de incerteza no processamento de eventos primitivos e sua propagação para os eventos complexos derivados a partir de dados de múltiplos sensores no ambiente de Internet das Coisas. Por sua vez, a **Internet das Coisas** (*Internet of Things* - **IoT**) surgiu a partir da proliferação de objetos ("coisas") e dispositivos móveis conectados, o que resulta em uma aquisição de fluxos de dados periódicos de diferentes dispositivos e sensores que precisam ser processados [60]. A IoT propõe a expansão da atual estrutura da Internet para uma rede de objetos interligados que não apenas colhem informações do ambiente mas que também interagem com o mundo físico para prestar serviços de transferência de informação, análises, aplicações e comunicações [54] [31]. São exemplos de algumas aplicações de IoT: *Smart Cities*, *Smart Factories*, *Smart Buildings*, *Smart Homes e Smart Cars* [1]. As plataformas de IoT demandam uma enorme quantidade de interações entre dispositivos que revelam alguns aspectos de Big Data, tais como volume, velocidade, variedade e veracidade (ou incerteza) dos dados. Entender o que está acontecendo nos ambientes de IoT e suas interações está se tornando uma tarefa cada vez mais difícil. Pensar em tais interações como **notificações de eventos** de modo a processá-los, pode apoiar a análise e a interpretação dessas informações.

Nos últimos anos, modelos de comunicação e processamento orientados a eventos têm sido amplamente difundidos, estudados pela comunidade acadêmica e aceitos pela indústria. Diversos domínios de aplicação têm se beneficiado na área de **processamento de eventos** envolvendo tanto aspectos de comunicação quanto de processamento baseados em eventos [23]. A ideia chave é explorar relacionamentos de temporalidade e causalidade entre os eventos para dar sentido a eles em tempo hábil. Isso revela oportunidades e/ou ameaças tão logo elas surjam ou podem servir para diagnosticar e executar decisões onde o tempo limitado é a principal restrição.

Eventos podem ser pensados como ocorrências únicas de interesse no tempo. Porém, identificar situações ou padrões que compreendem uma composição particular dessas ocorrências para obter informações significativas para sistemas, torna o processamento desses eventos, **complexo**. Assim, sistemas capazes de processar dados de forma eficiente para imediatamente reconhecer situações complexas de interesse enquanto elas ocorrem, define o **Processamento de Eventos Complexos** (**CEP** - *Complex Event Processing*) [15]. CEP pode ser visto como um paradigma de programação que suporta reações a fluxos de dados de eventos em tempo real[1], através de um conjunto de métodos e técnicas para realizar tal processamento [26] [45] [7]. Identificar ocorrências de interesse próximas do tempo real, frente a uma grande variedade de fluxos de eventos de diversas fontes é um dos requisitos essenciais em muitas das aplicações de IoT baseadas em processamento de eventos.

1.2 Caracterização do Problema

O processamento de eventos tornou-se usual em muitas aplicações de IoT do mundo real e a incerteza está inerente em tais aplicações de IoT orientadas a fluxos de eventos em tempo real. Ao considerar o processamento de eventos, as aplicações assumem que o fluxo de eventos é estável e os eventos são capturados conforme eles acontecem. Assume-se também que todos os canais de comunicação e sensores são confiáveis e o processamento de eventos **não** ocorre sobre dados incertos [75] [84]. Porém, em aplicações de IoT do mundo real, produtores de eventos geram fluxos de eventos que são frequentemente não confiáveis. Este é o caso quando tais aplicações têm que funcionar em ambientes físicos reais, intrinsecamente complexos e imprevisíveis com dados adquiridos de dispositivos, sensores e instrumentos de leitura que podem conter diferentes níveis de precisão definidos por diversos fabricantes. Além disso, nessas aplicações, as informações contidas nos fluxos de eventos podem conter imprecisões na origem do evento [8]. Por exemplo, sensores e dispositivos que emitem os eventos podem estar defeituosos ou acometidos por falhas, má calibração, produzindo dados imprecisos, o que configura o tipo de **incerteza na fonte do evento**, segundo a taxonomia de Wasserkrug et. al [82].

[1]Um curto período de tempo necessário para que os sistemas computacionais recebam dados e informações, e os disponibilizem imediatamente.

Em IoT, uma das técnicas mais proeminentes para a análise de fluxos de eventos provenientes de sensores, dispositivos e sistemas presentes em um espaço físico é o processamento de eventos complexos. Através de uma estrutura baseada em um conjunto de regras, tecnologias CEP permitem extrair informações sobre padrões de relacionamentos entre eventos simples (também denominados de eventos primitivos) e imediatamente derivar eventos complexos de mais alto nível (às vezes denominados de situações) [12] [3] [46]. Mais precisamente, cada regra é executada por um estágio intermediário de processamento CEP conhecido como *Event Processing Agent* (**EPA**). A partir da comunicação entre EPAs, através da interconexão entre seus terminais de entrada e saída tem-se uma rede de processamento de eventos (ou *Event Processing Network* - **EPN**) [25]. A **incerteza no processamento de eventos** costuma ser observada nos eventos (ex: dados ou leituras de sensores) e propagada para uma EPN que pode levar a resultados não confiáveis ou conclusões conflitantes. As fontes que produzem os eventos transmitidos podem incorporar imprecisões [8] [5] para o conteúdo do evento (atributos) [17] [70] [89] [61] [84] ou mesmo fornecer julgamento não confiável sobre uma ocorrência de eventos derivados [16] [79]. Os dados imprecisos propagados para o processamento de fluxos de eventos podem resultar em informações não verdadeiras ou ainda em um comportamento incorreto de sistemas. Em casos como rede de sensores, o fluxo de eventos pode ser ruidoso e não ser assumido como um fluxo válido e estável. Vários trabalhos de pesquisas foram realizados sobre o processamento de fluxos de eventos ruidosos [41] [74] [2] e processamento de eventos sob incerteza [84] [83]. No fluxo de eventos gerado através de sensores, é possível associar aos dados capturados níveis de incerteza, pois sensores podem prover dados erroneamente quando diante de situações para as quais não foram projetados.

Portanto, o **escopo deste trabalho** compreende a identificação e o tratamento da incerteza em aplicações de IoT baseadas em processamento de eventos, introduzida por dados de sensores não confiáveis, ou seja, a incerteza originada na fonte do evento primitivo; além disso, compreende também o tratamento da propagação da incerteza dos eventos primitivos que afetam diretamente o processamento de eventos derivados, ou seja, a incerteza dos eventos complexos. O escopo deste trabalho é considerado problema de pesquisa relevante na literatura [27] [75] [17] [84] [8] [5].

Considerando os problemas de incerteza levantados, cabe investigar as possíveis influências da incerteza na fonte do evento sobre os resultados do processamento e detecção dos eventos derivados. Segundo a taxonomia de Wasserkrug et al. [82], este tipo de incerteza é definida como a incerteza resultante da inferência do evento. Segundo Akila et al [3], a incerteza nos eventos primitivos degrada a precisão dos resultados (eventos complexos) significativamente em termos de falsos positivos e falsos negativos. Como consequência, o processamento de eventos "puro" que não considera incerteza, também não atende às expectativas de alcançar os objetivos e as capacidades de tomada de decisão em aplicações de missão crítica de diversos domínios sob condições de incerteza.

1.3 Hipótese de Pesquisa

Motivado pelos problemas de incerteza e as investigações iniciais, buscou-se uma teoria matemática para tratar os problemas de incerteza característicos do processamento de eventos em um ambiente de IoT. A Teoria de Dempster-Shafer (TDS) foi introduzida formalmente em 1976, através do trabalho de Glenn Shafer [68], baseado na extensão dos trabalhos de Arthur Dempster [19]. Também apresentada como **teoria das evidências** pelo fato de lidar com a sustentação de hipóteses baseada em evidências.

A relação entre uma evidência e uma hipótese corresponde à relação de causa e consequência, ou seja, uma evidência implica em uma hipótese ou um conjunto de hipóteses. As evidências podem ser modeladas como eventos que ocorreram ou podem ocorrer dentro de uma aplicação de IoT de processamento de eventos. A força de uma atribuição evidência-hipótese, ou a força dessa implicação é quantificada pela declaração de uma pessoa (especialista), estudo, organização, ou entidade (fonte de dados) que provê informação para um cenário de IoT. Estudos realizados evidenciam que a Teoria de Dempster-Shafer pode permitir representar e propagar valores de incerteza e, consequentemente, fornecer uma indicação de certeza sobre o resultado apresentado [19] [68]. Dessa forma, este trabalho visa explorar meios de utilizar a TDS para alcançar o tratamento de incerteza na origem e sua propagação em processamento eventos, provenientes de informações de sensores e dispositivos não confiáveis no ambiente de IoT. Neste viés, uma abordagem baseada na

TDS, especificamente projetada para representar e lidar com os problemas de incerteza descritos neste estudo, pode ser utilizada para lidar com a incerteza no processamento de eventos em aplicações de IoT do mundo real.

Logo, a **hipótese de pesquisa** deste trabalho é:

A Teoria de Dempster-Shafer possibilita desenvolver uma abordagem para adequadamente modelar e tratar os problemas de incerteza originados de dados de sensores não confiáveis e, especificamente para aplicações de IoT baseadas em processamento de eventos, lidar com a incerteza na origem dos eventos e sua propagação para os eventos derivados.

1.4 Objetivos

O objetivo geral dessa pesquisa é investigar o tratamento de incerteza no processamento de fluxos de eventos em aplicações de Internet das Coisas e contribuir com uma solução explorando a Teoria de Dempster-Shafer (TDS) para modelar os problemas de incerteza na fonte do evento e a propagação da incerteza para os eventos complexos. Para tanto, consideram-se os seguintes objetivos específicos:

- Desenvolver uma abordagem para identificar, modelar e tratar adequadamente a incerteza nos eventos e sua propagação para os eventos complexos em uma aplicação de IoT;
  - Construir um modelo de representação formal de eventos de modo a suportar parâmetros de incerteza da fonte produtora do evento;
  - Projetar e implementar um modelo arquitetural com base matemática sólida e desenvolvido em uma plataforma CEP;
  - Traduzir as leituras e a precisão dos sensores em valores de incerteza associados aos dados observados que servirão de base para o uso adequado da Teoria de Dempster-Shafer;
  - Integrar o modelo de incerteza do evento ao processamento de eventos, manipulando-os de forma consistente.
- Utilizar métricas de performance do estado da arte para avaliar o desempenho da solução para o tratamento de incerteza em uma aplicação de IoT baseada em processamento de eventos.

1.5 Metodologia de Pesquisa

A metodologia de pesquisa adotada nessa tese de doutorado pressupõe as seguintes etapas:

1. Sobre a pesquisa.

   O primeiro passo para realização deste trabalho foi a pesquisa exploratória restringindo-se à elaboração de diversos aspectos gerais do problema e definição de objetivos. Em seguida, uma pesquisa descritiva permitiu fazer uma análise minuciosa e descritiva do objeto de estudo, definição do problema e aprofundamento do assunto [86]. Neste sentido, as metodologias de pesquisa exploratória e descritiva envolveram o levantamento bibliográfico, o que permitiu uma explanação crítica e científica sobre o tema desta pesquisa.

2. Identificação do problema em aberto.

   A partir de um levantamento bibliográfico detalhado sobre incerteza no processamento de eventos complexos, mais especificamente em cenários e aplicações de IoT, buscou-se o estado da arte e a linha de pesquisa deste trabalho. Portanto, foi possível definir o problema em aberto alinhado com as temáticas de pesquisa do laboratória LSDi (Laboratório de Sistemas Distribuídos Inteligentes).

3. Elaboração da hipótese de pesquisa.

   Com base em estudo sobre um conjunto de trabalhos relacionados, exposição dos problemas de incerteza característicos do processamento de eventos em ambientes de IoT, além de evidências encontradas na TDS como um possível caminho para solucionar parte dos problemas levantados nesta pesquisa, elaborou-se a hipótese de pesquisa.

4. Concepção e desenvolvimento da solução.

   Nesse ponto procurou-se conceber uma solução inovadora e diferenciada em relação aos trabalhos encontrados na literatura. Para provar a hipótese, deu-se início à concepção e desenvolvimento da proposta de solução DST-CEP para abordar o problema identificado. A abordagem DST-CEP define a representação formal do evento e um modelo arquitetural. Tal proposta de modelo arquitetural realiza a modelagem da incerteza nos eventos e a propagação para os eventos

complexos sob a perspectiva dos elementos e funções da Teoria de Dempster-Shafer. Um framework DST-CEP foi desenvolvido refletindo a implementação do modelo arquitetural. Finalmente, um estudo de caso é realizado considerando o desenvolvimento de uma aplicação de IoT, que por sua vez, faz o uso do framework DST-CEP.

5. Avaliação da solução proposta.

   Na sequência, avaliou-se a abordagem DST-CEP sob vários aspectos. Primeiro, com o objetivo de investigar os benefícios do tratamento de incerteza com a abordagem DST-CEP, a aplicação de detecção de incêndio apresentada no estudo de caso foi submetida para o processamento de um conjunto de dados (*dataset*) coletados de sensores reais. Uma *baseline* de comparação foi montada para explorar a análise do processamento das regras CEP sem considerar incerteza para fins de comparação com os resultados DST-CEP que consideram incerteza. Além disso, foram exploradas na *baseline* abordagens probabilísticas frente à abordagem DST-CEP. Ao final, métricas de performance foram utilizadas na avaliação, como *Accuracy*, *Precision*, *Recall* e *F-Measure*, além de curvas ROC (*Receiver Operation Characteristic*) e AUC (*Area Under Curve*).

1.6 Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma:

- O **Capítulo 2** exibe uma fundamentação teórica sobre Internet das Coisas, Processamento de Eventos Complexos em IoT, a Teoria de Dempster-Shafer e uma discussão das teorias relacionadas para tratar incerteza;
- O **Capítulo 3** exibe uma revisão da literatura, e de forma ilustrativa, apresenta um mapa de tópicos dos assuntos relacionados com o tema desta pesquisa. Apresenta diferentes abordagens para lidar com o problema desta pesquisa, e ao final, realiza um comparativo entre os trabalhos relacionados;
- O **Capítulo 4** apresenta a abordagem *DST-CEP* proposta. Para isso, um estudo de caso com a apresentação de um cenário de IoT com múltiplos sensores ilustra os detalhes de implementação, processamento, cálculos e aplicação da abordagem;

- O **capítulo 5** apresenta a implementação do framework *DST-CEP* considerando o modelo arquitetural *DST-CEP*. Inicialmente é apresentado um estudo de caso de uma aplicação de IoT de processamento de eventos sob incerteza. Em seguida, são apresentados a modelagem e os detalhes de implementação do framework.

- O **capítulo 6** apresenta avaliações experimentais da solução proposta nesta pesquisa utilizando um *dataset* de sensores reais e várias métricas de performance. O capítulo também apresenta uma análise e discussão dos resultados.

- O **capítulo 7** descreve as conclusões, trabalhos futuros e contribuições desta tese de doutorado.

2 Fundamentação Teórica

Este capítulo apresenta uma introdução à Internet da Coisas e ao Processamento de Eventos Complexos para entendimento do contexto em que este trabalho de pesquisa está situado. Além disso, uma introdução sobre a Teoria de Dempster-Shafer é apresentada, bem como sua relação com a Teoria de Probabilidade e com as Redes Bayesianas, todas no cenário de tratamento de incerteza. Toda a fundamentação das teorias citadas é de essencial conhecimento para o entendimento da solução proposta neste trabalho.

2.1 Processamento de Eventos Complexos em IoT

O processamento orientado a eventos tem sido amplamente difundido pela comunidade acadêmica e aceito pela indústria, sendo que vários domínios de aplicação têm se beneficiado do processamento baseados em eventos. Luckham (2001) [46] define eventos como registros imutáveis da ocorrência de uma ação ou mudança de estado de um sistema, e o processamento de eventos como um método de rastreamento e análise de fluxos de informações (dados) sobre coisas que acontecem (eventos) que derivam uma conclusão. Em Luckham (2011) [45], um evento é apresentado como algo que acontece ou é rotulado como acontecendo, e também está relacionado com o processamento de ocorrências em tempo real.

O processamento de eventos em aplicações de IoT explora em tempo hábil, relacionamentos de temporalidade, causalidade e semântica entre os eventos com o objetivo de revelar oportunidades e/ou ameaças tão logo elas surjam, bem como servir para diagnosticar e executar decisões em tempo limitado. Este fato torna o processamento de eventos requisito essencial em muitas aplicações de IoT que têm o objetivo de detectar ocorrências ou situações de interesse próximas do tempo real. Além disso, o processamento de eventos deve oferecer para ambientes e aplicações de IoT uma visibilidade e meios de processamento de uma grande variedade de dados (ex., leituras de diversos sensores) em tempo real.

Como foi observado, um dos requisitos fundamentais em aplicações de IoT para o emprego de tecnologias CEP é o processamento crítico de eventos onde o prazo de execução para uma tarefa não deve ser violado. Vale ressaltar, a partir deste requisito, outra característica importante que difere a tecnologia CEP de outros sistemas gerenciadores de fluxos de dados e deve trazer benefícios para IoT. CEP armazena as consultas contínuas que são executadas à medida que os dados fluem sobre as consultas. Ou seja, ao invés de armazenar os dados, CEP se concentra em analisar e processar continuamente os dados enquanto eles passam, usando para isso, consultas armazenadas. Cada consulta contínua utiliza uma ou mais primitivas CEP em tempo real, como por exemplo, *filter* e *negation* [63]. Uma sequência ordenada de eventos através do tempo define um fluxo de eventos (*Event Stream*) que é a principal fonte de entrada para as consultas contínuas. As consultas contínuas permitem combinar primitivas CEP de tempo real para reagir, processar e derivar outros eventos de mais alto nível.

Ambientes de IoT demandam uma infinidade de fluxos de informações, colaborações humanas baseadas em computador, negócios eletrônicos, interações com agentes de software, sensores e atuadores, ou seja, uma grande quantidade de dados gerados continuamente. As plataformas de IoT devem processar uma enorme quantidade de fluxos de interações e têm se beneficiado do uso de estruturas de processamento de eventos para consumir fluxos de informação de várias fontes e derivar situações. Um exemplo de tal estrutura é a Rede de Processamento de Eventos (**EPN** - *Event Processing Network*) como ilustra a Figura 2.1. Na EPN, os eventos são criados por produtores (*producers*), que são entidades (por exemplo, sensores e aplicações clientes) que geram ocorrências de interesse em uma aplicação de IoT. O fluxo de eventos é o resultado de uma sequência de eventos criada e enviada por *producers*. O *workflow* CEP processa continuamente essa sequência de eventos de entrada, analisa e manipula tais eventos. Em seguida, têm-se eventos derivados na saída que são entregues para entidades consumidoras (*consumers*) (por exemplo, aplicações de monitoramento). Essas saídas geralmente representam notificações sobre situações detectadas. Mais precisamente, é possível identificar hierarquias de eventos, onde cada consulta contínua é executada por um estágio intermediário de processamento CEP conhecido como *Event Processing Agent* (**EPA**). A partir da comunicação entre EPAs, produtores e consumidores, através da interconexão entre

seus terminais de entrada e saída, tem-se finalmente, uma rede de processamento de eventos (EPN) [25], que é empregada em muitas aplicações de IoT.

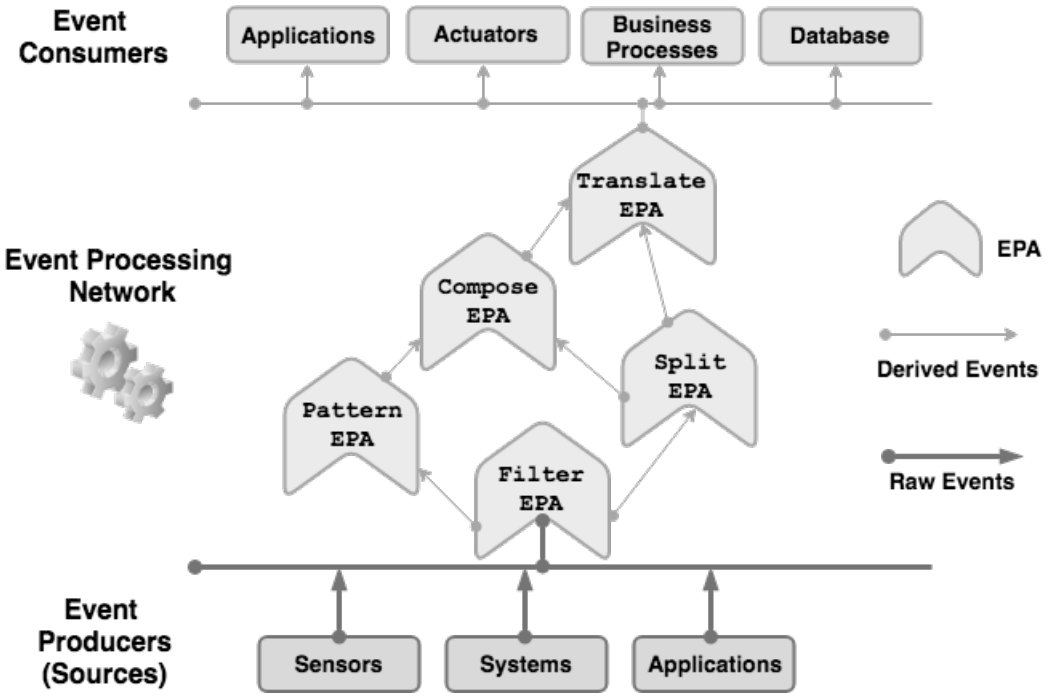


Figura 2.1: Visão geral da EPN.

2.2 Teoria de Dempster-Shafer

A Teoria de Dempster-Shafer (TDS) foi introduzida formalmente por Glenn Shafer [68] baseada na extensão dos trabalhos de Arthur Dempster [19]. A TDS, também denominada **teoria das evidências**, apresenta um conjunto funções e elementos que abordam a ideia da sustentação de hipóteses baseada em evidências.

Segundo estudos da TDS [73], é possível acreditar em uma hipótese se ela concordar com uma percepção, embora possa existir um distanciamento entre as percepções e a realidade, o que torna necessária a noção de "Evidência", que pode ser forte ou fraca em relação a uma determinada hipótese. Para Shafer [68], não é esperado que exista uma relação objetiva entre uma dada evidência e uma dada hipótese em que se determine um grau numericamente preciso. Nem que tal grau, nesta relação evidência-hipótese, possa sempre ser determinado. Ao invés disso, pressupõe-se que havendo verificado percepções que constituem um corpo de evidências, poderá ser anunciado um número que represente o grau que se julga que a evidência sustenta

uma dada hipótese, e consequentemente, o grau de crença que se deseja atribuir a esta hipótese. Os elementos e as funções da TDS serão apresentados ao longo das próximas seções.

2.2.1 Quadro de Discernimento

A Teoria de Dempster-Shafer assume inicialmente um Quadro de Discernimento (**FoD** do inglês, *Frame of Discernment*), que é um conjunto de hipóteses primitivas, por exemplo (h_1, h_2 e h_3), sobre algum domínio do problema ou ambiente de interesse. Assim, um FoD é representado por Θ e consiste do conjunto de elementos do ambiente de interesse $\Theta = \{h_1, h_2, h_3\}$. Dado o conjunto de hipóteses primitivas, todos os subconjuntos formados pelos elementos de $\Theta = \{h_1, h_2, h_3\}$ dão origem a 2^Θ possibilidades de combinação de hipóteses, que podem ser interpretadas como possíveis novas hipóteses de interesse, como segue:

$$2^\Theta = \{\{h_1, h_2, h_3\}, \{h_1, h_2\}, \{h_1, h_3\}, \{h_2, h_3\}, \{h_1\}, \{h_2\}, \{h_3\}, \emptyset\} \quad (2.1)$$

Geralmente, o conjunto vazio ($\emptyset$) não é considerado porque sempre corresponde à resposta falsa, não tem elementos. Embora o conjunto vazio sempre será um subconjunto de Θ, ele não será listado como um subconjunto válido no restante do trabalho.

Para ilustrar a noção de quadro discernimento, hipóteses e evidências, assume-se o exemplo de uma *Smart Home* em IoT, equipada com múltiplos sensores, que visa detectar a atividade atual de uma pessoa em determinado ambiente da casa. Em um único ambiente podem ocorrer várias atividades e cada atividade é determinada pela combinação das notificações de diversos sensores. O quadro de discernimento neste cenário é constituído pelas seguintes hipóteses de atividades:

$$\Theta = \{h_1, h_2, h_3\}$$

As hipóteses de atividades podem ser inferidas a partir das notificações de sensores (evidências). Por exemplo, dados os três sensores (S_1, S_2, S_3), a atividade $\{h_1\}$ é inferida quando os sensores S_1 e S_2 são disparados. Ou ainda, a atividade $\{h_2\}$ é inferida a partir das notificações de S_2 e S_3. Se somente o sensor S_2 for disparado, isso

implica em uma evidência para que as atividades h_1 ou h_2 possam estar ocorrendo, dessa forma uma indicação para a hipótese $\{h_1, h_2\}$ formada por um subconjunto de Θ. Embora cada subconjunto de Θ possa ser considerado uma hipótese, em determinados domínios é usual que nem todos os subconjuntos possíveis sejam de interesse.

2.2.2 Função de Massa

Na TDS, o valor de crença para cada elemento de 2^{Θ} (ou cada hipótese) é representada por uma função chamada **função de massa**, às vezes denominada **massa básica de crença**. A função de massa é a fonte inicial ou atribuição básica da TDS que indica quão fortemente uma evidência suporta (ou "sustenta") determinada hipótese [87]. Com base nas evidências, para indicar a crença em determinada hipótese, a TDS associa um número no intervalo [0, 1] que mede o quanto uma evidência suporta uma hipótese. Este número é o valor de **massa (m)** atribuído pela função de massa que realiza um mapeamento das crenças atribuídas às hipóteses. Portanto, dada uma evidência inicial, a função massa calcula o valor massa que tal evidência suporta uma hipótese ou um conjunto de hipóteses de um determinado domínio.

Vale ressaltar que a teoria não se concentra no ato de julgamento, cálculo ou mecanismo pelo qual o valor da massa é determinado. Em vez disso, a TDS concentra-se em **combinar** valores de massas distintos atribuídos para diferentes hipóteses, onde tais massas são baseadas em evidências de múltiplas fontes.

Para ilustrar uma distribuição de massas, retomando o exemplo da *Smart Home*, assume-se que a aplicação deve requerer dos sensores a localização do usuário em determinado ambiente da casa para inferir uma atividade. Porém, o usuário é detectado em vários cômodos devido ao seu movimento ou à precisão limitada dos sensores, o que gera incerteza no processo de identificação da atividade atual do usuário. Ao considerar essa situação, e com base nas evidências, a indicação dos valores de massa são: para as hipóteses das atividades h_2 ou h_3, a massa é de 0.2, podendo-se usar a seguinte **representação**[1]: $m(h_2, h_3) = 0.2$. A partir da mesma fonte

[1]Por questões de facilitar a escrita e compreensão das notações, será adotado ao longo do texto que, por exemplo, a escrita $m(h_2, h_3)$ deve representar a mesma notação $m(\{h_2, h_3\})$. Ou ainda, que a escrita $m(h_1)$ representa a mesma notação $m(\{h_1\})$.

de evidências (sensor), há também uma indicação de maior massa para a hipótese da atividade h_1, ou seja, o valor $m(h_1) = 0.6$.

Vale observar que, nos capítulos subsequentes, será discutido com mais profundidade o cálculo dos valores e distribuição de massas a partir dos dados de sensores não confiáveis. Porém, para fins de esclarecimento da TDS nesta seção, assume-se valores de massa fictícios a partir das leituras de sensores.

Por definição, a Teoria de Dempster-Shafer (na Equação 2.2) exige que a partir de uma fonte, a soma das massas atribuídas às hipóteses de interesse seja igual a 1. Formalmente, se H_i representa qualquer elemento de 2^Θ, então $m : 2^\Theta \rightarrow [0, 1]$ se satisfaz:

$$\begin{aligned} m(\emptyset) &= 0 \\ \sum_{H_i \in 2^\Theta} m(H_i) &= 1 \end{aligned} \tag{2.2}$$

A quantidade $m(H_i)$ é entendida como a medida de parte da massa (ou parte da crença total) que é atribuída exclusivamente para H_i, onde H_i é qualquer elemento de 2^Θ. A condição $m(\emptyset) = 0$ reflete o fato de que nenhuma crença deve ser atribuída para ($\emptyset$), na Equação 2.2 pode-se observar que a soma de todas as massas atribuídas às hipóteses H_i pertencentes a 2^Θ seja igual a 1 (atribuição básica total = 1):

Entretanto, no exemplo da *Smart Home*, a soma das massas atribuídas às hipóteses $m(h_1) = 0.6$ e $m(h_2, h_3) = 0.2$ não é igual a um. O restante da massa (cujo valor é 0.2) não deve ser atribuída a outra hipótese caso não exista evidência que justifique tal obrigação. A TDS define que a crença não atribuída às hipóteses restantes por falta de evidências deve ser atribuída à hipótese que representa o domínio do problema, na TDS representada por Θ, e não aos elementos restantes. Portanto, o valor de massa que "sobra" devido à falta de evidências, ignorância ou informações desconhecidas deve ser atribuído à hipótese Θ, ou seja, $m(\Theta) = 0.2$.

Por convenção, o valor de massa que sobra, após terem sido atribuídas as massas aos subconjuntos próprios de Θ, é chamada de **crença não atribuída**. Sabe-se que as evidências existentes suportam as hipóteses com os valores $m(h_1) = 0.6$ e $m(h_2, h_3) = 0.2$. A hipótese $\Theta = \{h_1, h_2, h_3\}$ é constituída de todas as hipóteses primitivas do domínio do problema e também representa a hipótese de incerteza da

fonte de informação. Assim, a crença não atribuída ou massa restante (0.2) é então atribuída à hipótese Θ, que representa incerteza da fonte de informação ou todas as hipóteses de atividades no cenário apresentado da *Smart Home*.

Logo, a atribuição de massa básica de crença inicial segue:

$$m(h_1) = 0.6$$
$$m(h_2, h_3) = 0.2$$
$$m(\Theta) = 0.2$$

2.2.3 Regra de Combinação de Dempster

Assim que evidências adicionais se tornam disponíveis, é possível combiná-las para produzir uma melhor estimativa sobre as hipóteses. Por exemplo, se múltiplas fontes provêm novas notificações ou evidências sobre as hipóteses do quadro de discernimento, diversas fontes podem também registrar massas indicando a ocorrência de diferentes hipóteses. Nessa situação, a **Regra de Combinação de Dempster** permite calcular novos valores de crença sobre as hipóteses e que representam um consenso baseado na combinação de valores de massas (evidências) de diversas fontes.

Continuando o exemplo da *Smart Home*, a Teoria Dempster-Shafer deve permitir combinar diferentes evidências através da sua regra de combinação. Dadas as massas atribuídas no início do exemplo, designadas por m_1:

$$m_1(h_1) = 0.6$$
$$m_1(h_2, h_3) = 0.2$$
$$m_1(\Theta) = 0.2$$

A partir de um segundo conjunto de leituras de um diferente sensor, tem-se uma segunda atribuição de massas (m_2) para as seguintes hipóteses de atividades:

$$m_2(h_1, h_3) = 0.1$$
$$m_2(h_1) = 0.6$$
$$m_2(\Theta) = 0.3$$

Dadas as duas atribuições de massas distintas m_1 e m_2, a regra de combinação de Dempster realiza a **soma ortogonal** das atribuições para produzir uma nova massa que representa um consenso das evidências originais e possivelmente conflitantes[2], neste caso indicada pela notação $m_1 \oplus m_2$. Formalmente, a soma ortogonal ($\oplus$) é definida pela soma dos produtos das massas das hipóteses (X e Y) de duas fontes, cuja interseção $X \cap Y = H_i$, como segue:

$$m_1 \oplus m_2(H_i) = \frac{1}{k} \cdot \sum_{X \cap Y = H_i} m_1(X) \cdot m_2(Y) \tag{2.3}$$

A nova massa normalizada atribuída a H_i, é a soma ortogonal das massas dividida pelo fator de normalização $\boldsymbol{k}$. O fator de normalização $\boldsymbol{k}$ é o resultado da subtração de um menos a soma do produto das massas atribuídas aos conjuntos vazios ($X \cap Y = \emptyset$):

$$k = 1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y) \tag{2.4}$$

Eventualmente, se ao utilizar a equação 2.3 não ocorrem casos de interseção nula ($X \cap Y = \emptyset$), então a equação 2.3 se resume na equação a seguir:

$$m_1 \oplus m_2(H_i) = \sum_{X \cap Y = H_i} m_1(X) \cdot m_2(Y) \tag{2.5}$$

Considerando no exemplo as massas m_1, m_2 e as equações acima, as novas massas (m_3) são:

[2]Conflitantes nesse contexto quer dizer evidências contra e a favor de uma mesma hipótese de um domínio.

$$
\begin{aligned}
m_3(\Theta) &= 0.0682 \\
m_3(h_2, h_3) &= 0.0682 \\
m_3(h_1, h_3) &= 0.0227 \\
m_3(h_3) &= 0.0227 \\
m_3(h_1) &= 0.8182
\end{aligned}
$$

Vale observar que o detalhamento numérico dos cálculos destes resultados é apresentado na seção 4.2.2. Os resultados encontrados mostram que o maior valor de crença apresentado é $m_3(h_1) = 0.8182$, ou seja, a atividade mais provável e mais assertiva a ser considerada dentre as evidências e as hipóteses analisadas é a atividade h_1.

Portanto, será investigado como a Teoria de Dempster-Shafer poderá ser empregada no processo de análise de incertezas presentes em ambientes de IoT, considerando o processamento de dados de sensores não confiáveis. A TDS poderá configurar elemento fundamental para a análise de eventos complexos a partir dos dados, capturados dos sensores, denominados de evidências e na definição da hipótese mais assertiva dentro de um domínio Θ envolvido.

2.3 Teoria de Probabilidades para Tratar Incerteza

A Teoria de Dempster-Shafer oferece uma alternativa à Teoria de Probabilidade tradicional para a representação e o raciocínio sobre informações incertas. Dessa forma, esta seção apresenta inicialmente alguns conceitos básicos de probabilidades e em seguida limitações da Teoria de Probabilidades para tratar incerteza. Finalmente, limitações da Teoria de Probabilidades frente à Teoria de Dempster-Shafer são apresentadas.

2.3.1 Conceitos Básicos de Probabilidades

Um modelo probabilístico é uma descrição matemática de uma situação de incerteza. Todo modelo probabilístico envolve um processo subjacente chamado de **experimento** ou estudo de um **fenômeno aleatório**, que é qualquer observação cujo resultado não seja conhecido com certeza. Mesmo se o experimento for repetido

várias vezes sob condições semelhantes, apresenta resultados imprevisíveis sujeitos a incerteza. Por exemplo, jogar uma moeda uma ou várias vezes, lançar um dado, etc [21] [13].

Todo experimento aleatório possui a característica de ser possível listar um conjunto de todos os possíveis resultados, denominado de **espaço amostral**, e denotado pela letra grega ômega (Ω). Um exemplo clássico de espaço amostral é o lançamento de um dado comum, representado pelo conjunto Ω = {1, 2, 3, 4, 5, 6}. Já o **evento** é qualquer subconjunto do espaço amostral. Por exemplo, dado o lançamento simultâneo de **dois** dados, o espaço amostral é representado pelo conjunto de todas as possibilidades de faces voltadas para cima [55]:

$$\Omega = \{(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(2,1),(2,2),(2,3),...,(6,6)\} \quad (2.6)$$

Dentro desse espaço amostral alguns **eventos** podem ser de interesse, por exemplo:

- **Evento** A: soma é igual a 12. $A = \{(6,6)\}$
- **Evento** B: a soma é menor que 4. $B = \{(1,1),(1,2),(2,1)\}$

Probabilidade [40] [13] é um valor numérico que representa uma chance, uma eventualidade ou uma possibilidade de que um determinado evento venha a acontecer. Dado um experimento e um espaço amostral Ω, o objetivo da probabilidade é atribuir a cada evento A um número $P(A)$, chamado de probabilidade do evento A, que dará uma medida da chance de A ocorrer, como ilustra o modelo probabilístico da Figura 2.2.

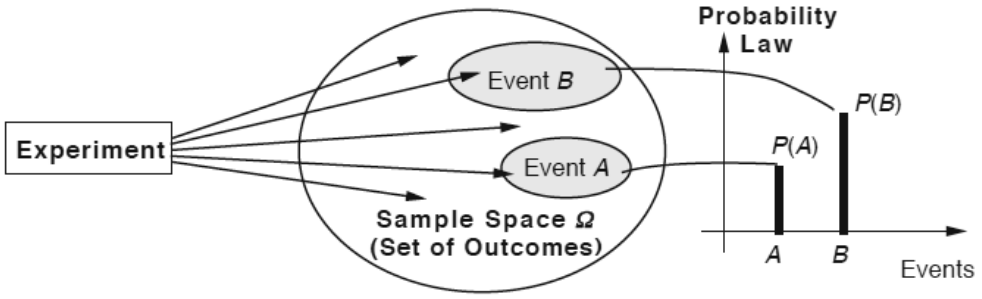


Figura 2.2: Elementos de um modelo probabilístico [13].

Para completar o modelo probabilístico, deve-se introduzir uma lei de probabilidade. Intuitivamente, isso especifica a "probabilidade" de qualquer

resultado, ou de qualquer conjunto de resultados possíveis (um evento). A lei de probabilidade atribui a cada evento A, um número $P(A)$, satisfazendo os seguintes **Axiomas de Probabilidade**:

1. $P(A) \geq 0$, para qualquer evento A, onde a chance de ocorrência do evento A deve ser **não negativa**;

2. $P(\Omega) = 1$, o espaço amostral contém todos os resultados possíveis. A probabilidade do espaço amostral inteiro Ω é igual a 1;

3. Se A e B são dois eventos mutuamente exclusivos, então a probabilidade da sua união satisfaz $P(A \cup B) = P(A) + P(B)$. Se o espaço de amostra tiver um número infinito de elementos mutuamente exclusivos $A_1, A_2, ...$, então a probabilidade de sua união satisfaz $P(A_1 \cup A_2 \cup ...) = P(A_1) + P(A_2) + ...$ (**axioma de aditividade, Fig. 2.3**).

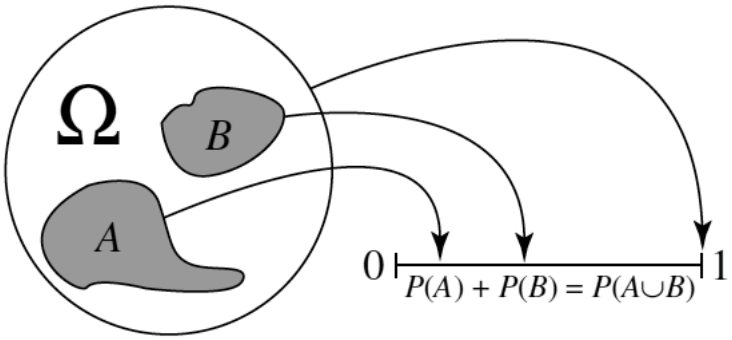


Figura 2.3: Axioma de Aditividade.

Existem diferentes abordagens ou definições para obter o valor da probabilidade de um evento.

Probabilidade Clássica

A abordagem clássica é usada quando cada resultado no espaço amostral tem a mesma probabilidade de ocorrer, ou seja, se o espaço amostral é equiprovável. Neste caso, a probabilidade é baseada no conhecimento prévio do fenômeno observado. Por exemplo, a probabilidade de ocorrência de um evento A, indicada por P(A), é dada por:

$$P(A) = \frac{n^o \text{ de elementos do evento } A}{n^o \text{ de elementos do espaço amostral}}$$

No exemplo do lançamento de dois dados, dado um evento A = "obter valores iguais nos dois dados" = {(1,1), (2,2), (3,3), (4,4), (5,5), (6,6)}, a probabilidade da ocorrência deste evento é:

$$P(A) = \frac{6}{36} = \frac{1}{6} \cong 0,1666$$

Probabilidade Frequentista

A probabilidade frequentista de um evento consiste em repetir o experimento aleatório, muitas vezes, anotando a frequência com que o evento de interesse ocorre. Os resultados são baseados em dados observados e não no conhecimento prévio do fenômeno envolvido. De acordo com a Lei dos grandes números [52], à medida que um experimento é repetido mais e mais vezes, a probabilidade teórica (frequência relativa) de um evento tende à sua probabilidade real. Assim, a probabilidade de ocorrência de um evento A, indicada por P(A), é dada por:

$$P(A) = \frac{n^o \textit{ de ocorrências do evento } A}{n^o \textit{ de repetições do experimento}}$$

Probabilidade Subjetiva

A abordagem frequentista de probabilidade aplica-se a fenômenos ou processos com o requisito de que possam ser repetidos, pelo menos conceitualmente, sob as mesmas condições. Entretanto, muitas situações não podem ser repetidas sob condições iguais. Em tais situações, é também desejável atribuir probabilidades numéricas a vários eventos. Por exemplo, médicos algumas vezes atribuem probabilidades subjetivas à expectativa de vida para pessoas com câncer. Previsão do tempo é um outro exemplo de probabilidades subjetivas. Desse modo, a probabilidade subjetiva baseia-se em experiências do passado, opinião ou análise pessoal e é especialmente útil na tomada de decisões, quando estas não puderem ser determinadas pela probabilidade clássica ou frequentista [55].

2.3.2 Relações da Teoria de Probabilidades e a Incerteza

Avanços em termos de poder computacional permitiram aos sistemas análises mais complexas no domínio da incerteza, porém aplicar apenas **uma** estrutura

matemática, como a teoria de probabilidade tradicional, se torna uma limitação para representar todo o escopo da incerteza.

A natureza dual da incerteza é descrita com as seguintes definições de Helton [32]:

- **Incerteza Aleatória**: resulta do fato de que um sistema pode se comportar de maneira aleatória, também conhecida como: incerteza irredutível ou incerteza objetiva. A incerteza aleatória não está relacionada apenas com algo ou fenômeno que não se conhece, mas também que não se pode conhecer, ou seja, é incognoscível. Por exemplo, por mais que se queira saber se vai chover daqui a um ano na Filadélfia, por mais que se consulte os maiores metereologistas e se tente superar as médias sazonais, sempre será um problema irremediavelmente nebuloso, com uma incerteza que é impossível, mesmo em teoria, de eliminar;

- **Incerteza Epistêmica**: resulta da falta de conhecimento sobre um sistema, porém que pode ser explorada. Tal incerteza está relacionada a algo que não se conhece, mas teoricamente, se tiver meios pode ser conhecido, ou seja, é cognoscível. Por exemplo, analistas ou especialistas podem com habilidade entender, medir e descrever um sistema sob investigação, ou seja, a incerteza epistêmica possui a propriedade de ser passível de análises. Também conhecida como: incerteza redutível ou incerteza subjetiva. Isso por que esta fonte de incerteza pode, a princípio, ser reduzida pelo crescimento no conhecimento sobre um sistema, através de estudo suficiente ou conhecimento dos especialistas.

Limitações da Teoria de Probabilidades para tratar Incerteza

Tradicionalmente, a teoria da probabilidade tem sido usada para caracterizar os dois tipos de incerteza. A **incerteza aleatória** tem sido tratada usando a **abordagem frequentista** de probabilidade. No entanto, a teoria da probabilidade tradicional é limitada para lidar com a **incerteza epistêmica** em determinadas situações. Por exemplo, a aplicação de alguns métodos probabilísticos para lidar com a incerteza epistêmica (subjetiva) exige que um analista tenha informações sobre a probabilidade de todos os eventos. Quando esta informação não está disponível é usada com frequência a função de distribuição uniforme, onde todos os eventos

simples para os quais a probabilidade não é conhecida são igualmente prováveis em um dado espaço amostral (justificado por Laplace - *Principle of Insufficient Reason* [65]). Para ilustrar tal situação, considera-se uma falha no sistema em que há três componentes possíveis (A, B, C) que poderiam ter causado esta falha. Um especialista em confiabilidade do componente A atribui uma probabilidade 0,3 de falha desse componente. Este especialista não sabe nada sobre as outras duas fontes potenciais de falha (Componentes B e C). Uma análise probabilística tradicional poderia atribuir uma probabilidade de falha de 0,35 a cada um dos dois componentes restantes (B e C). Esta seria uma afirmação muito precisa e perigosa sobre a probabilidade de falha desses dois componentes em face da **completa ignorância** em relação a esses componentes, por parte do especialista [67].

Dentro dessa reflexão, o axioma da aditividade onde a soma de todas as probabilidades deve ser igual a 1, satisfazendo propriedades específicas de eventos complementares, força a conclusão de que o conhecimento de um evento implica necessariamente no conhecimento do seu complemento. Neste caso, a probabilidade de ocorrência de um evento é traduzida no conhecimento da probabilidade de que esse evento não ocorra. No exemplo anterior, se um especialista acredita que um sistema pode falhar devido a um componente em particular com probabilidade de 0,3, significa, necessariamente, que o especialista acredita que o sistema **não** falhará com probabilidade 0,7. Isso configura o desafio de modelar qualquer incerteza associada à crença subjetiva de um especialista [67].

Embora a teoria de probabilidades possa ser apropriada para modelar os eventos aleatórios associados à incerteza aleatória, as restrições levantadas abrem precedentes para questionar sua aplicação em situações como:

1. Quando há pouca informação para avaliar uma probabilidade;

2. Quando a informação é imprecisa, incompleta, conflitante ou não específica.

Vale ressaltar que as definições formais da probabilidade não explicitam as situações (ou condições) especiais em que elas são fundamentadas, como por exemplo, a existência de um espaço amostral previamente conhecido (probabilidade clássica) ou uma longa sequência de repetições de um experimento (probabilidade frequentista). Quando um fenômeno de interesse não se encaixa nestas condições especiais, os

conceitos e as definições de probabilidades não são suficientes para modelar todas as situações de incerteza, em especial, a incerteza subjetiva. Ou seja, a **probabilidade** passa então a ser insuficiente, se considerada apenas em seu sentido clássico, por modelar apenas a incerteza aleatória ou objetiva. Entra em cena o termo "crença", onde tanto as incertezas objetivas quanto às subjetivas são modeladas [69].

Relações da Teoria de Probabilidades com a TDS

Para Shafer em [69], probabilidade é o grau racional de crença. Ou seja, a probabilidade de um evento $P(A)$ é o grau com que deve-se acreditar que A irá acontecer, ou o grau com o qual as evidências **suportam** (sustentam) o acontecimento de A. Porém, cabe a pergunta se existe um **grau numérico preciso** de que as evidências suportam a ocorrência de A? Segundo Shafer [69], proponentes recentes da interpretação do "suporte", admitem que é difícil medir graus de suporte, mas eles estão convencidos de que as evidências dão suporte para as crenças. Outro ponto a ser considerado, quando não é possível caracterizar a incerteza com uma medida precisa (probabilidade precisa), é razoável considerar uma medida para um intervalo ou um conjunto, como ocorre na TDS. [67].

Adicionalmente, a TDS foi estabelecida para a representação e o raciocínio sobre informações incertas, imprecisas e incompletas [71]. Dempster propôs uma nova forma de lidar com a incerteza por causa das seguintes limitações da teoria da probabilidades, discutidas na seção anterior:

- A dificuldade de representar ignorância. Na teoria da probabilidade, a ignorância é representada através da atribuição de probabilidades iguais a todos os eventos anteriores e é cercada de dificuldades e limitações;
- A exigência de crença em um evento e também a sua negação, com a soma igual a um. Dempster afirmou que, em muitas situações a evidência que suporta uma hipótese não deve necessariamente diminuir a crença para todas as outras restantes [20]. Na TDS, não há nenhuma exigência de que a crença não comprometida com uma dada proposição deve estar comprometida com a sua negação. Isso faz com que a alocação total de crença possa variar de acordo com a extensão do conhecimento sobre as hipóteses do problema.

Finalmente, algumas vantagens da TDS são resumidas por Liu et al. [42], onde destaca-se que a TDS tem a capacidade de modelar a informação de uma maneira flexível sem a necessidade de uma probabilidade ser atribuída a cada elemento de um conjunto. A TDS fornece um mecanismo conveniente e simples para a combinação de duas ou mais evidências sob certas condições. Pode modelar a ignorância explicitamente e rejeita da lei da aditividade (axioma) ao considerar a crença em subconjunto de hipóteses.

2.4 Redes Bayesianas para tratar Incerteza

Essa seção tem como objetivo introduzir conceitos básicos da teoria de Redes Bayesianas, bem como seu uso e limitações para tratar incerteza. Ao final da seção são apresentadas as relações das Redes Bayesianas com a Teoria de Dempster-Shafer.

2.4.1 Conceitos Básicos de Redes Bayesinas

Redes Bayesianas (RB), também chamadas de redes de crenças (*belief networks*), redes probabilísticas ou redes causais, são tradicionalmente entendidas como um modelo gráfico de variáveis e suas relações para um problema específico. Sua representação é baseada em um grafo acíclico direcionado (*Directed Acyclic Graph* - DAG) [33]. Uma RB é constituída de nós e arcos. Os **nós** representam as variáveis da rede e propriedades relevantes de um determinado sistema, e os **arcos** direcionados (*links*) representam a dependência condicional entre pares de nós [64] [58]. Existe uma hierarquia de nós dentro da rede onde os termos pai e filho são usados para referenciar a relação de dependência direta entre dois nós por meio do arco que os conecta. O nó de onde o arco parte é designado nó pai e o nó onde o arco chega com sua ponta é designado nó filho.

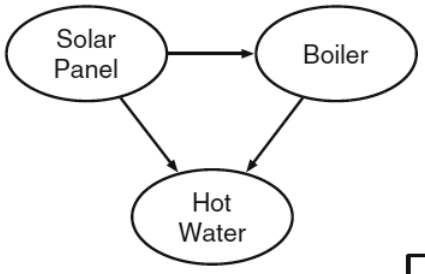
Outro elemento importante dentro da estrutura de Redes Bayesianas é a tabela de probabilidade condicional (*Conditional Probability Table* - CPT). Trata-se da exibição dos parâmetros de probabilidade condicional da variável sendo condicionada a seu(s) pai(s). Ou seja, para cada uma das variáveis e seus cruzamentos condicionais,

tem-se uma CPT que explica numericamente a chance da cada evento ocorrer dadas as variáveis anteriores.

A Figura 2.4 ilustra um exemplo de uma Rede Bayesiana que modela a disponibilidade de água quente usando três variáveis [17]: ***Solar Panel, Boiler*** e ***Hot Water***, respectivamente painel solar, caldeira e água quente. Cada variável pode assumir dois valores (`ON, OFF`). As duas linhas entrando no nó água quente representam uma dependência causal com ambos painel solar e caldeira. Esta dependência é quantificada na tabela associada à água quente da seguinte maneira: a disponibilidade de água quente é certa quando tanto a caldeira quanto o painel solar estão ligados (`ON`), e altamente provável, ou seja, (P(`ON`) = 0.9) quando um deles está `ON` e o outro está `OFF`. Quando ambos, a caldeira e o painel solar estão `OFF`, é altamente provável que **não** haja água quente (P (`OFF`) = 0.9).

Solar Panel	
ON	OFF
0.6	0.4

	Boiler	
Solar Panel	ON	OFF
ON	0.2	0.8
OFF	0.9	0.1



		Hot Water	
Solar Panel	Boiler	ON	OFF
ON	ON	1.0	0.0
ON	OFF	0.9	0.1
OFF	ON	0.9	0.1
OFF	OFF	0.1	0.9

Figura 2.4: Exemplo de Rede Bayesiana [17]

Vale observar que as Redes Bayesianas (Figura 2.4) são compostas de duas partes complementares: uma parte qualitativa e a outra quantitativa.

Na **parte qualitativa** são levadas em consideração as relações gerais entre as variáveis de interesse, em termos da relevância (ou dependência) de uma variável para outra. O resultado é uma representação gráfica que captura as dependências condicionais de uma forma qualitativa, isto é, não numérica [18]. Observa-se no exemplo (Figura 2.4) que a rede inclui uma dependência causal entre o painel solar e a caldeira que modela a presença de um controlador para ligar ou desligar a caldeira

(`ON/OFF`) dependendo do estado do painel solar. Assim, um arco ligando as variáveis A e B, na seguinte forma $A \rightarrow B$, indica que a variável B é a consequência e a variável A é a causa, e estas apresentam uma relação de dependência, resumidas na regra "se A então B". Por outro lado, se não houver um arco ligando duas variáveis então se assume que essas variáveis são independentes [78].

Na **parte quantitativa** da RB, para os arcos na representação gráfica são atribuídos números representando probabilidades condicionais para cada nó que têm pai. Essas probabilidades são essencialmente probabilidades subjetivas, pois dependem do julgamento pessoal, ou da consulta a usuários e especialistas para se chegar as estimativas subjetivas das probabilidades condicionais. Para os nós que **não** têm pai ou para o nó raiz (*root node*) são atribuídas as probabilidades *a priori* calculadas utilizando a teoria da probabilidade a partir dos valores já elicitados do domínio [18] [53]. Por exemplo (Figura 2.4), o estado do nó raiz painel solar não depende ou possui relação causal com qualquer outra variável. Assim, tabela de painel solar é preenchida com a probabilidade *a priori* de que o painel solar esteja funcionando corretamente.

Finalmente, uma RB pode ser usada para inferir dependências condicionais entre varáveis e valores esperados para uma variável, dada a distribuição *a priori* de valores na rede.

2.4.2 Limitações de Redes Bayesianas para tratar Incerteza

Redes Bayesianas são um modelo matemático que captura o conhecimento do domínio e as incertezas envolvidas. Porém, construir uma RB pode ser custoso e difícil pois pode envolver grande consumo de tempo e, em muitos casos, o próprio domínio não é bem conhecido. Além disso, julgar relações qualitativas entre as variáveis de um domínio é a parte mais simples da construção da rede. Porém pode-se encontrar dificuldades em quantificar as probabilidades associadas. Assim, considera-se que a construção da **parte qualitativa** de uma RB é factível, enquanto a parte desafiadora é avaliar as probabilidades das variáveis que formam a **parte quantitativa** [57].

Geralmente, a RB é extraída a partir do conhecimento de um especialista sobre o domínio de uma aplicação. Entretanto, pesquisas vêm sendo realizadas

buscando construir uma RB utilizando algoritmos capazes de estimar os valores das probabilidades dos nós da rede a partir de conjuntos de dados (*datasets*). As probabilidades condicionais da rede e as probabilidades *a priori* não precisam ser conhecidas previamente, ou seja, podem ser aprendidas usando técnicas de amostragem estatística ou abordagens de aprendizado supervisionado [53].

Segundo Pearl [58], uma RB requer que probabilidades condicionais numéricas completas sejam especificadas entre os nós. Esse requisito se torna inviável quando há dificuldade na quantificação das probabilidades, dado que a presença de um nó influencia diretamente na especificação da probabilidade condicional de seus nós filhos. Dependendo do nível de conhecimento do problema por parte do especialista e da disponibilidade de informações sobre o domínio em análise, poderia ser difícil a obtenção das probabilidades de toda a rede.

2.4.3 Relações das Redes Bayesianas com a TDS

A Teoria de Dempster-Shafer (TDS) trata a representação de incertezas de forma semelhante à Redes Bayesianas. No entanto, na TDS, o raciocínio é feito com medidas de crença, que são obtidas por meio de funções de crença. Uma Rede Bayesiana pode ser pensada como uma base de conhecimento que representa explicitamente crenças e as relações sobre elementos de um sistema. O propósito de tal base de conhecimento é inferir alguma crença ou eventos em um sistema [53].

Propagação da Incerteza para os resultados (outputs)

Em geral, uma probabilidade de interesse é representada pela saída (*output*) de uma rede Bayesiana. Isso por que, na maioria dos domínios de aplicação, uma rede bayesiana é usada para fazer diagnósticos ou previsões baseadas na probabilidade de interesse computada. A robustez do resultado de uma RB representada por uma probabilidade de interesse, no entanto, **não** representa a robustez de toda a RB, uma vez que há incertezas e imprecisões quanto às possíveis probabilidades atribuídas aos nós da rede, que por sua vez afetam a probabilidade de interesse de saída.

Segundo Shafer [68], conhecendo todas as probabilidades, então seguramente pode-se adotá-las como graus de crença. Caso não se conheça todas

as probabilidades, a TDS permite expressar a ignorância parcial e total de forma extremamente adequada, ao contrário da abordagem Bayesiana que expressa a ignorância parcial atribuindo-se crença à negação da hipótese, e quanto a ignorância total dividindo-se o total de crença entre as hipóteses presentes (eventualmente atribuindo mais crença do que realmente possuem) [76].

Dessa forma, como afirma Shafer em [68]: "Desde que não se requeira que expressemos nossa evidência como uma certeza, a TDS permite-nos construir descrições de raciocínio provável que são mais modestas que descrições Bayesianas e mais fidedignas à forma humana de pensar."

TDS mais Flexível

A análise Bayesiana é focada na atribuição de probabilidades a cada proposição (ou hipótese) individual de um conjunto de hipóteses mutuamente exclusivas. Como alternativa, a TDS atribui crenças para hipóteses ou combinação de hipóteses de forma independente. No exemplo da *smart home*, dados de sensores podem indicar três atividades, A1, A2 ou A3. Uma abordagem Bayesiana pode atribuir probabilidades individualmente a A1, A2, A3 como {0.1, 0.2, 0.7}. Enquanto a TDS pode atribuir probabilidade a cada uma das oito possibilidades $\{\{A1, A2, A3\}, \{A1, A2\}, \{A1, A3\}, \{A2, A3\}, \{A1\}, \{A2\}, \{A3\}, \emptyset\}$. Isso permite uma flexibilidade extra para a TDS, que pode ser considerada uma generalização da abordagem Bayesiana. Observa-se que uma abordagem Bayesiana pode assumir a probabilidade da atividade A1 ou A2, P(A1) ou P(A2), que pela soma estaria no intervalo 0.2 a 0.3. Por outro lado, a TDS poderia atribuir para P(A1) ou P(A2) um valor que excedesse a soma das atribuições individuais para (A1) e (A2). Outra possibilidade é que a TDS suporta a situação de não ocorrência de atribuições individuais de $\{A1\}$ e $\{A2\}$, independente do valor atribuído para $\{A1, A2\}$. Na abordagem Bayesiana, para alterar a atribuição $\{A1, A2\}$, seria necessário alterar as atribuições $\{A1\}$ e $\{A2\}$ individualmente, enquanto a TDS permite alterar o valor de $\{A1, A2\}$ sem alterar as atribuições individuais de $\{A1\}$ ou $\{A2\}$.

Outro ponto a ser considerado é que na teoria de Dempster-Shafer, as funções de crença obtidas a partir das informações disponíveis podem assumir várias formas, incluindo as distribuições de probabilidades convencionais que

possuem alguns problemas e limitações para tratar incerteza já discutidos nas seções anteriores. O conceito de uma função de crença da TDS torna possível representar a evidência mais fielmente de acordo com a qualidade da evidência e sua origem, além disso é muito mais flexível do que as distribuições de probabilidade convencionais para expressar evidências. Na abordagem Bayesiana, qualquer tipo de evidência, independentemente da sua qualidade, deve ser representada na forma de probabilidade convencional. Assim, a TDS é um método mais flexível quanto à representação de hipóteses.

Vale ressaltar também limitações que podem ser observadas na TDS. Um comportamento contra-intuitivo pode ser observado na regra de combinação de Dempster quando as evidências a serem combinadas possuem distribuições de valores de crença altamente conflitantes e divergentes. Por exemplo, uma primeira fonte de informação apresenta a distribuição de crença A = 0.99 e B = 0.01, enquanto uma segunda fonte apresenta C = 0.99 e B = 0.01. Neste caso, o elemento do quadro de discernimento não intuitivo (elemento B) recebe maior valor de crença ao final da combinação. Este problema se apresenta quando há uma assimetria conflitante de valores das evidências coletadas. Para solucionar este problema, uma medida de peso de conflito calculada através do fator de normalização da combinação permite detectar e desconsiderar evidências altamente conflitantes.

2.5 Síntese

Nesse capítulo, primeiramente, os conceitos de Internet das Coisas (IoT) e o Processamento de Eventos Complexos (CEP) foram vistos. Em seguida, foi apresentada a Teoria de Dempster-Shafer (TDS) e ressaltadas as vantagens da TDS para o tratamento de incerteza. Essa afirmativa foi reforçada pelo conteúdo abordado na sequência, que discutiu as desvantagens da Teoria de Probabilidades e Redes Bayesianas para tratar incerteza, bem como as limitações das duas teorias frente à TDS. Assim, a TDS pode ser empregada no tratamento de incertezas presentes em ambientes de IoT, considerando o processamento de dados de sensores não confiáveis.

3 Trabalhos Relacionados

Neste capítulo são apresentadas as principais pesquisas, abordagens e técnicas da literatura relacionadas com o tratamento de incerteza no processamento de eventos complexos. Inicialmente é apresentado um mapeamento dos trabalhos relacionados a partir do estudo de três *surveys* que abordam o tema. Um mapa de tópicos é elaborado para cada *survey* e em seguida é apresentado um mapa geral de tópicos para estruturar todo o conhecimento sobre o assunto. A partir do mapa de tópicos, são estabelecidos critérios para a seleção de oito trabalhos relacionados que são abordados individualmente com ênfase nos objetivos, tipos de incertezas abordadas, pontos relevantes da solução, síntese dos resultados e limitações. Vale observar que buscou-se selecionar também trabalhos recentes e fora do escopo dos trabalhos verificados nos *surveys* estudados. Ao final do capítulo são estabelecidos critérios de comparação entres os oito trabalhos, além de uma tabela comparativa que ilustra a relação dos itens de comparação com cada trabalho selecionado e o fechamento do capítulo apresenta uma discussão sobre os principais problemas e limitações das abordagens.

3.1 Mapeamento dos Trabalhos Relacionados

A forma como a incerteza é tratada em CEP depende de especificidades dos sistemas, dos tipos de incertezas relacionadas, além de problemas e domínios específicos de aplicações. Lidar com todos os casos possíveis de incerteza em CEP em um único método parece improvável, de modo que cada trabalho se concentra em suas próprias restrições. Consequentemente, a literatura oferece uma ampla variedade de abordagens, não apenas da perspectiva técnica mas também conceitual. Assim, foram selecionados três extensos *surveys*: Flouris et al. [27], Akila et al. [3] e Alevizos et al. [6] que fornecem uma visão geral sobre as principais pesquisas e técnicas existentes para realizar o tratamento de incerteza em CEP.

A partir de cada *survey* foi elaborado um **mapa de tópicos**, que é uma padronização ISO/IEC 13250:2003[12] para a representação do conhecimento e organização de estruturas complexas de informações [59]. De maneira particular, o mapa de tópicos estrutura todo o conhecimento sobre um assunto com o objetivo de relacionar e mesclar tópicos relevantes e adjacentes a partir de várias fontes.

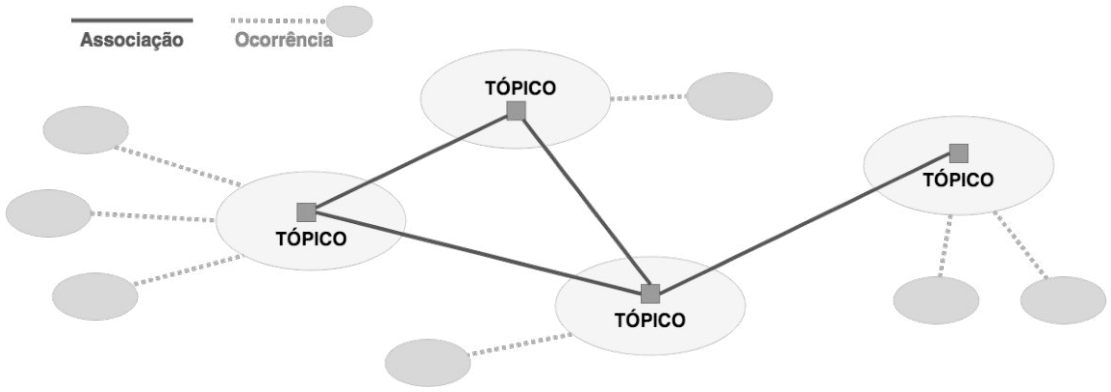


Figura 3.1: Mapa de Tópicos

Como ilustra a Figura 3.1, o mapa de tópicos é composto pelos seguintes elementos [28]:

- Tópico: representa qualquer conceito, assunto ou termo que representa uma ideia. Quando agrupados, formam um conjunto de tópicos de conhecimento para o domínio em questão;
- Associação: representa a relação entre os tópicos de conhecimento;
- Ocorrência: informações que são pertinentes e relevantes de alguma forma para um determinado tópico de conhecimento.

Dessa forma, são apresentados a seguir três *surveys* destacando os tópicos relacionados ao domínio de incerteza em CEP.

[1]Revisada recentemente para ISO/IEC 13250-2:2006 e ISO/IEC 13250-3:2007, o padrão especifica o modelo do mapa de tópicos, define uma estrutura abstrata e de interpretação dos mapas de tópicos.

[2]O padrão ISO/IEC 13250:2003 define regras para mesclar mapas de tópicos e o conjunto de ocorrências (assuntos) fundamentais do domínio em questão.

3.1.1 *Issues in complex event processing: Status and prospects in the Big Data era* [27]

O primeiro *survey*, de Flouris et al. 2017 [27], tem como foco levantar os principais problemas de pesquisas relacionados com CEP. No entanto, somente os conceitos restritos a incerteza foram identificados e destacados deste trabalho, como ilustra a Figura 3.2 construída a partir dessa fonte. As elipses representam os tópicos e ocorrências relevantes dentro domínio de incerteza em CEP, as linhas relacionam todo o conhecimento apresentado. Observa-se que no mapa de tópicos deste trabalho são estruturados os tipos de modelos dos eventos processados no fluxo, os tipos de incertezas associadas ao modelo de eventos probabilísticos e os impactos deste modelo no processamento CEP.

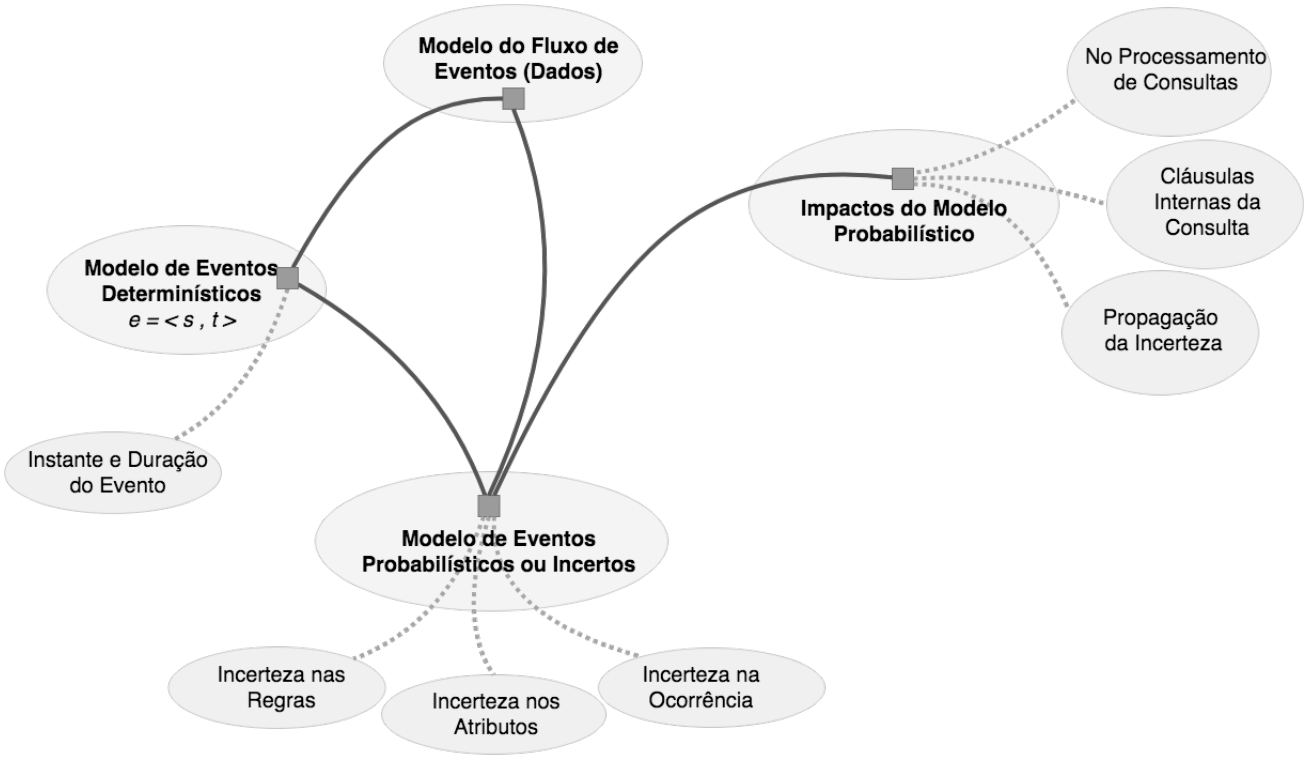


Figura 3.2: Mapa de Tópicos de Incerteza em CEP, a partir de Flouris et al. 2017 [27]

O **modelo de eventos determinísticos** é caracterizado pela ideia de um evento ser visto como uma "tupla" que representa a ocorrência de um evento de interesse. Neste modelo, a definição do evento e, dada pela tupla $e =< s, t >$ (como apresentado em [66]), possui o conjunto s de atributos com informações que indicam o significado do evento, além dos atributos de tempo t constituídos pelo tempo (ou instante) de ocorrência do evento. Já o **modelo de eventos probabilísticos** herda a

estrutura de representação de eventos determinísticos $e = < s, t >$, mas considera na sua representação que as fontes que produzem os eventos podem incorporar imprecisões para um sistema CEP [8] [5]. Nesse sentido, um conjunto de trabalhos considera que uma fonte imprecisa afeta os dados do evento ocasionando **incerteza no conteúdo do evento** [17] [84] [89]. Neste caso, o modelo de eventos probabilísticos representa seus atributos s ou t acompanhados de valores de probabilidades ou distribuições de probabilidades (mais detalhes apresentados na seção 3.2.3). Para outro grupo de trabalhos [79] [16], uma fonte imprecisa pode ocasionar um julgamento falho sobre a ocorrência de um evento. Neste caso, o modelo probabilístico representa a **incerteza na ocorrência de um evento** pela tupla $< e, p_e >$ onde p_e significa a probabilidade de ocorrência do evento e. Além disso, existe a **incerteza nas regras** ou **padrões** que podem ser probabilísticos quando exige-se o processamento de eventos incertos [17] [84] [83] [85].

Neste *survey* é discutido o impacto da **incerteza no processamento de consultas CEP**. Por exemplo, no caso da incerteza na ocorrência do evento e cuja probabilidade de ocorrência é representada por $< e, p_e >$, no caso complementar $< \neg e, 1 - p_e >$, observa-se a probabilidade de não ocorrência do evento. Supondo uma consulta que envolva vários tipos de eventos, cada qual com duas instâncias probabilísticas, tem-se uma maior quantidade de verificações na consulta, devido as instâncias probabilísticas. No caso do modelo de eventos determinísticos, uma quantidade menor de verificações na consulta são realizadas. Um conjunto de trabalhos emprega este modelo de probabilidade de ocorrência para realizar a avaliação de consultas sobre o fluxo de eventos probabilísticos [16] [61] [17] [79].

Outro conjunto de trabalhos apresenta técnicas para evitar a sobrecarga de verificações nas consultas. Uma forma é utilizar **cláusulas da própria linguagem de consulta CEP** para manipulação de eventos probabilísticos. Por exemplo, uma forma é utilizar a cláusula HAVING incorporada à consulta para verificar um limiar de "confiança" ou probabilidade, neste caso filtrando eventos altamente improváveis. Essa técnica de "poda" (ou corte) baseada em predicados é vista como uma otimização e explicitamente levada em consideração no conjunto de trabalhos [89] [79] [16].

Outro aspecto considerado são as técnicas de **propagação de incerteza**, neste caso quando admite-se incerteza (através de valores de probabilidade) em eventos primitivos, exige-se a definição de um mecanismo que mapeia ou quantifica

tais valores de incerteza para o evento complexo derivado (propagação). Diferentes abordagens de propagação de incerteza podem afetar a detecção do evento complexo. Por exemplo, em algumas situações os eventos de entrada são dependentes (a probabilidade do primeiro evento tem influência sobre a probabilidade do segundo) ou em algumas situações os eventos são independentes (a probabilidade do primeiro evento não interfere ou tem relação com a probabilidade do segundo). As situações de dependência entre eventos afetam severamente a complexidade do processamento. A linha de trabalhos de Wasserkrug et al. [81] [84] [83] [85] assume a independência de eventos, pelo menos até o nível dos eventos primitivos (detalhes na seção 3.2.1). Um outro grupo de trabalhos emprega abordagens *Markovianas* [62] [79] [61] para propagar incerteza entre eventos (detalhes na seção 3.5.1). Outro grupo de trabalhos adota Redes Bayesianas como parte da solução para propagar incerteza em CEP [17] [83] [85] (detalhes na seção 3.2.3).

3.1.2 *Complex Event Processing over Uncertain Events: Techniques, Challenges, and Future Directions* [3]

O *survey* de Akila et al. 2016 [3] enfatiza a importância do tratamento de incerteza em CEP e destaca limitações observadas nas abordagens CEP relacionadas a incapacidade de definir, modelar e propagar a incerteza e a imprecisão de aplicações nesse contexto. Cita-se que na presença de incerteza nos eventos, é importante investigar como transformar em tempo real, os dados dos eventos incertos adequados para o consumo de aplicações finais de usuários e terceiros. Além disso, no âmbito das consultas CEP, a expressividade da especificação das consultas se torna limitada em função da capacidade dos sistemas CEP em processar somente eventos determinísticos e a partir de fontes confiáveis. Finalmente destaca-se que, na presença de incerteza, existem limitações nas abordagens e modelos, para inferir e capturar a correlação entre dados incompletos e imprecisos de aplicações em ambientes reais, o que torna essencial incorporar a incerteza na forma de eventos probabilísticos.

Portanto, como ilustra a Figura 3.3, este *survey* tem um foco inicial em incerteza nos dados, apresenta alguns modelos que lidam com esse tipo de incerteza e como tais modelos podem apoiar o processamento CEP sob incerteza. Além disso, o *survey* aborda individualmente Sistemas CEP, Linguagens de Consultas de Eventos e

Eventos Incertos. Entretanto, observa-se que essas áreas principais se inter-relacionam ao longo do texto, o que resulta em subconjuntos dessas relações. Logo, como ilustra a Figura 3.3, apenas os subconjuntos em comum (texto em azul) que têm interseção com **Eventos Incertos** foram destacados desse *survey*.

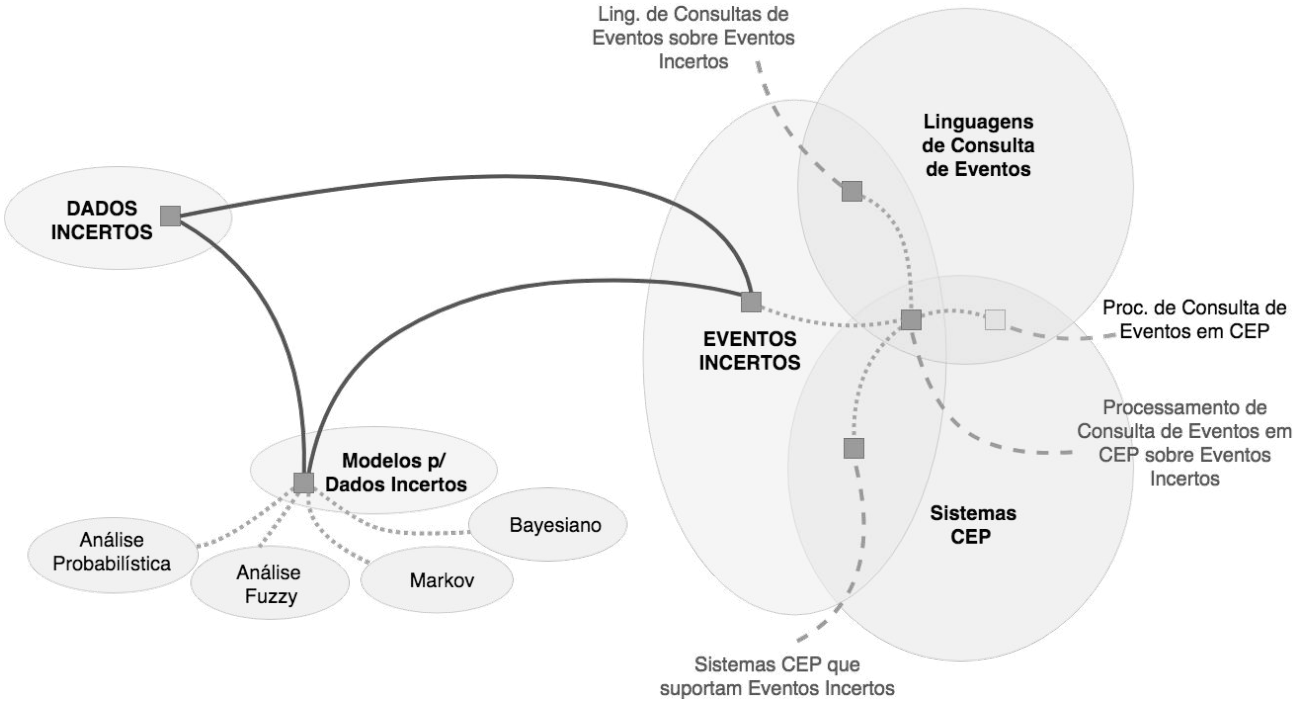


Figura 3.3: Mapa de Tópicos de Incerteza em CEP, a partir de Akila et al. 2016 [3]

A incerteza nos dados (ou dados incertos) é definida como a falta de precisão ou conhecimento incompleto da informação que leva a uma situação complexa para descrever um possível resultado. Também definida como informações insuficientes, imprecisas e vagas que dificultam o processo de tomada de decisão em um sistema complexo. A análise da incerteza envolve o processo de medir, reconhecer e minimizar todos os tipos de incertezas, enquanto o gerenciamento de dados incertos concentra-se principalmente em técnicas de coleta, modelagem, representação e consulta, de dados e eventos incertos.

A representação de dados incertos depende de um modelo para capturá-los. Os **modelos para dados incertos** influenciam a precisão e o desempenho das aplicações, e tentar equilibrar um limiar entre usabilidade e expressividade do modelo depende da necessidade e requisitos da aplicação. Os modelos a seguir discutem algumas das técnicas usadas no gerenciamento de dados incertos que criam uma base sólida para o processamento de eventos incertos. O **modelo de dados probabilísticos** especifica uma estrutura matemática para quantificar a incerteza e atribuir valores de

probabilidades associados aos atributos. Por exemplo, uma linha de trabalhos desses modelos estende o modelo relacional para representar as probabilidades associadas aos atributos, descrevendo-os como determinísticos ou probabilísticos, e esta inclusão exige linguagens descritivas mais ricas e um conjunto de novos operadores relacionais que estendem os operadores convencionais. Wasserkrug et al. [81] propõe um modelo probabilístico e ilustra o cálculo de probabilidades relevantes (detalhes na seção 3.2.1).

Os **modelos e análise *fuzzy*** são baseados em Lógica Fuzzy (LF) e Teoria de Conjuntos Fuzzy, que são estruturas matemáticas aptas para lidar com a imprecisão e a indeterminação (*vagueness*[3]) dos dados. A implementação da lógica *fuzzy* permite mecanismos de representação para melhorar a flexibilidade de lidar com conceitos linguísticos de dados e avaliar termos linguísticos (um caso prático na literatura é avaliar a temperatura para os termos: quente, frio, morno, baixa, alta, etc.). Alguns trabalhos utilizam formulações *fuzzy* para reconhecimento de atividades e comportamento humano, afim de superar limitações como a falta de leituras de sensores, sobreposição de atividades realizadas ao mesmo tempo, gerenciamento de imprecisões e dados incompletos[4] [22] [35].

Um **modelo baseado na rede de Markov** é um modelo gráfico não direcionado que representa probabilidades sobre sequências de observações, e o estado atual de uma variável observada é independente de todos os outros estados anteriores. Recentemente em Rince et al. 2018 [62] um modelo baseado em redes de Markov foi proposto para tratar incerteza em CEP com o objetivo de estimar a probabilidade de reconhecimento de um evento complexo em um período razoável de tempo a partir de um fluxo de dados potencialmente errôneo. Vale observar que os modelos markovianos possuem um custo computacional elevado incompatível com as restrições impostas pelos domínios de aplicações CEP, o que exige propostas de extensões ou combinações com outras técnicas para lidar com este problema no modelo (detalhes seção 3.5.1). Um **modelo baseado em redes Bayesianas** é um modelo de grafo acíclico direcionado que permite múltiplas variáveis e suas relações são definidas dependendo do problema específico (como visto na seção 2.4). Este

[3] *Vagueness* surge a partir do uso de termos não claros que podem criar interpretações erradas e confusas.

[4] Incompleto nesse contexto são fluxos de dados onde alguns eventos, que formam o fluxo, não são disparados quando na verdade deveriam disparar.

modelo também é chamado de rede de crença, rede probabilística ou rede causal devido à dependência entre eventos especificada em cada nó da rede pela tabela de probabilidade condicional. O modelo Bayesiano é amplamente utilizado em uma variedade de soluções para tratamento de incerteza em CEP, como observado na linha de propostas de Wasserkrug et al. [81] [83] [85] até trabalhos como Cugola et al. [17] (mais detalhes seção 3.2).

Neste *survey* é interessante observar como o autor inter-relaciona as três áreas divididas em **Sistemas CEP**, **Linguagens de consultas de eventos** e **Eventos Incertos**, todas ilustradas na Figura 3.3. Em cada intersecção entre essas principais áreas, existe um grupo de trabalhos relacionados. Assim, vale destacar como essas áreas se relacionam sob a perspectiva dos modelos de incerteza apresentados e citar alguns trabalhos dentro dessas intersecções.

Sistemas CEP com suporte para Eventos Incertos representam uma necessidade amplamente reconhecida pela comunidade CEP e uma linha significativa de pesquisa resultando em vários mecanismos de manipulação de eventos incertos. O processamento de eventos incertos em [36] estende o sistema CEP SASE modificando a abordagem baseada em autômatos para otimizar o custo de computação e o tempo de resposta sobre fluxos probabilísticos. Já em aplicações de tecnologia RFID, uma proposta de modelos probabilísticos extrai eventos complexos a partir de fluxos de dados RFID. Neste caso, o PEEX (Probabilistic Event EXtractor) é um sistema de extração de informações de alto nível a partir de eventos probabilísticos e lida com erros e ambiguidades de dados nesse cenário [37]. O CEP2U (Complex Event Processing under Uncertainty) é a proposta do Cugola et al. [17] para lidar com a incerteza da fonte do evento e a propagação da incerteza a partir de eventos primitivos, cuja implementação estende a *engine* T-REX (detalhes seção 3.2.3).

As **Linguagens de Consultas de Eventos sobre Eventos Incertos** precisam ser estendidas para modelar e capturar as incertezas subjacentes, e poucas linguagens de consulta foram estendidas com sucesso para manipular eventos incertos. O modelo CEP2U [17] trata a incerteza em eventos e regras, além de utilizar os recursos básicos da linguagem CEP TESLA para integrar o modelo proposto. O gerenciamento da incerteza é apoiado pela construção de redes Bayesianas a partir da definição de regras. A SWRL (Semantic Web Rule Language) é uma linguagem flexível e expressiva que inclui operadores com conjunções, negações, sequências e repetições, além de

suportar a manipulação de incerteza [47]. PEL (Probabilistic Event Language) permite definir a composição de eventos probabilísticos seguindo a sintaxe SQL e possui alguns construtores para identificar a ordem dos eventos [37].

O **Processamento de Consultas de Eventos em CEP sobre Eventos Incertos** tem seu resultado afetado, tanto pela natureza probabilística dos eventos, quanto pela incerteza nas regras e a propagação da incerteza. Assim, o processamento de consultas sobre fluxos de eventos probabilísticos se torna um desafio. Logo, são apresentadas algumas técnicas dentro dessa linha. CEP2U [17] integra as características de incerteza à linguagem TESLA para o suporte ao processamento de consultas. Wasserkrug et al. [85] propõe um componente que processa as notificações de eventos coletadas de diferentes fontes para descobrir padrões de interesse, com suporte a incerteza. Outra linha de trabalhos lida com o desafio de processar e detectar padrões de sequência de eventos com incerteza temporal ou *time stamps* imprecisos. Em [39], a imprecisão de dados sequenciais, ou uma sequência de localizações capturadas a partir de RFID/GPS é tratada como um fluxo probabilístico. Neste trabalho, o autor define um modelo baseado na sequência de eventos com suporte semântico para consultas de padrões e incertezas temporais e espaciais (localização).

3.1.3 *Probabilistic Complex Event Recognition: A Survey* [6]

O *survey* de Alevizos et al. 2017 [6] destaca as aplicações de reconhecimento de eventos complexos (ou CER - *Complex Event Recognition*) que exibem vários tipos de incertezas, variando de fluxos de dados incompletos e errôneos, até padrões de eventos complexos não precisos. Para o autor, a maioria dos sistemas CER (também entendidos como sistemas CEP) não lidam com a incerteza e essa necessidade está gradualmente sendo reconhecida e se tornando uma importante linha de pesquisa e desenvolvimento para o CEP (importância enfatizada em Cugola e Margara [30]).

Como ilustra a Figura 3.4, o propósito desse *survey* é apresentar uma visão geral das abordagens e métodos existentes para tratamento de incerteza em CEP. Destacam-se os tipos de incertezas encontradas em aplicações CEP, além de algumas abordagens e métodos para tratá-las. O processamento de consultas sobre dados incertos em fluxos probabilísticos é abordado em outras pesquisas [3] [80], por isso o autor não incluiu neste *survey*.

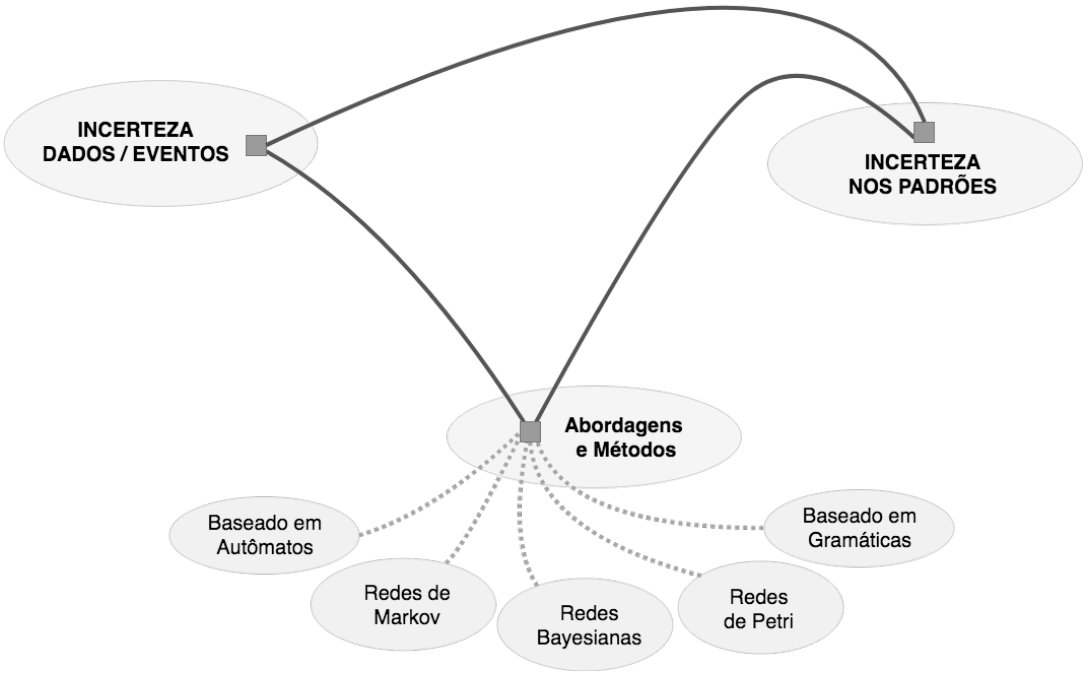


Figura 3.4: Mapa de Tópicos de Incerteza em CEP, a partir de Alevizos et al. 2017 [6]

Reconhecer e entender os tipos de incerteza é tarefa crucial para qualquer sistema CEP que pretende lidar com o problema. Neste *survey* a **incerteza nos dados** se estende para a **incerteza nos eventos**. Logo, as descrições foram unificadas como se segue (Figura 3.4). O fluxo de eventos que provê dados de entrada para uma *engine* CEP pode conter incertezas, por exemplo, à medida que os dados (também denominados de evidências) podem estar incompletos ou serem perdidos por vários motivos, seja devido à falha de sensores em reportar um evento decorrente do mal funcionamento do *hardware* ou devido à precisão limitada dos sensores. Em alguns casos, mesmo com o *hardware* funcionando corretamente, certas características do ambiente monitorado podem causar distorções ao longo da comunicação, evitando que os eventos sejam registrados. De maneira geral, o autor sugere que os eventos do fluxo de entrada com incerteza inerente, possam ser acompanhados com valores de probabilidades, como apresentado também na seção 3.1.1 em Flouris et al. [27].

A **incerteza nos padrões** pode ocorrer devido à complexidade ou falta de conhecimento de um domínio. Em determinadas situações, pode ser impossível capturar exatamente todas as condições que um padrão deve satisfazer tornando-o incompleto. Optar pela definição mais geral de um padrão em vez de tentar

determinar todas as suas condições específicas pode ser mais fácil de implementar. Além disso, não verificar várias condições pode ser mais eficiente quanto ao custo computacional e promover ganho de desempenho. Entretanto um padrão mais amplo pode se tornar mais problemático quanto à precisão dos resultados.

Neste *survey* foram identificadas as seguintes classes de abordagens e métodos decorrentes da necessidade de operar fluxos de eventos frente aos problemas de incerteza descritos acima.

Métodos baseados em Autômatos concentram-se no reconhecimento de sequências de eventos, onde alguns desses eventos podem estar relacionados pelos seus atributos ou sequência de ocorrências. Tais métodos parecem adequados para o CEP, já que os eventos de entrada são geralmente na forma de fluxos/sequências de eventos, similar às cadeias de caracteres reconhecidas em autômatos. Nas versões probabilísticas dos métodos baseados em autômatos, os eventos incertos são acompanhados de valores de probabilidade quanto à sua ocorrência e/ou atributos, e tais eventos probabilísticos são usados para determinar a probabilidade do evento complexo. Por exemplo, no trabalho de Kawashima et al. [36] uma árvore é construída com os eventos que acionam as transições de estado. Ao percorrer a árvore, a sequência de eventos simples permite identificar o evento complexo e capturar sua probabilidade de maneira direta através da multiplicação dos valores probabilísticos dos eventos simples considerados.

Métodos que utilizam **modelos gráficos probabilísticos** para lidar com a incerteza em CEP são divididos em duas principais classes: **Redes de Markov** (modelos não direcionados) e **Redes Bayesianas** (modelos direcionados). Tais modelos fornecem flexibilidade em relação à representação das probabilidades (nos atributos, ocorrência dos eventos e regras) que podem ser codificadas no processamento e lidam com restrições de dependências entre os eventos. Mais detalhes das duas abordagens são evidenciados nas seções subsequentes.

Métodos baseados em Redes de Petri têm sido propostos para reconhecer eventos complexos. Redes de Petri possuem a particularidade de permitir modelar sistemas assíncronos e não determinísticos. Como ilustra a Figura 3.5, a representação gráfica de uma rede de Petri é formada por dois componentes: um ativo chamado de transição (barra) e outro passivo denominado lugar (círculo), atividade ou recurso.

Os lugares equivalem às variáveis de estado (pode possuir marcas ou *tokens*) e as transições correspondem às ações realizadas pelo sistema (evento) [51].

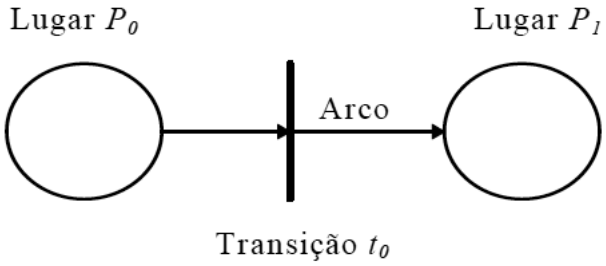


Figura 3.5: Elementos da Rede de Petri [51]

Alguns métodos têm empregado extensões probabilísticas das redes de Petri, por exemplo para reconhecer atividades em [38] [4]. Um evento complexo é representado na rede por eventos simples (com valores de certeza) e possíveis restrições (ex. temporais) codificados como condições de transição entre os nós da rede. A transição de um estado para outro é associada com um valor de probabilidade. A soma das probabilidades de todas as transições de um estado específico para todos os estados possíveis deve ser igual a 1 e são atribuídas probabilidades iguais a todos essas transições. Dada uma sequência de eventos, pode ser identificada a probabilidade de mudança para um novo estado (marcas) ou a probabilidade de uma sequência específica de eventos que representa um evento complexo.

Métodos baseados em Gramáticas têm focado em abordagens sintáticas que convertem um fluxo de eventos simples para um fluxo de símbolos de entrada, onde regras podem ser aplicadas para o reconhecimento de eventos complexos. Para levar em conta a incerteza, em [84] são atribuídas probabilidades para cada produção (ou sequência) da gramática, nesse ponto uma vantagem seria conseguir a probabilidade de "*matches* parciais" de um evento complexo, o que seria útil para realizar previsão de eventos que possam ocorrer.

3.1.4 *Mapa Geral de Tópicos do Domínio de Incerteza em CEP*

Observou-se nos *surveys* que a representação do conhecimento sobre incerteza em CEP não é absoluta, isto é, um mesmo domínio foi modelado sob diferentes aspectos, visões e características, de forma individualizada em cada

trabalho. Um ponto que torna o mapa de tópicos especial é a integração de informações e mapas permitindo uma combinação única de características.

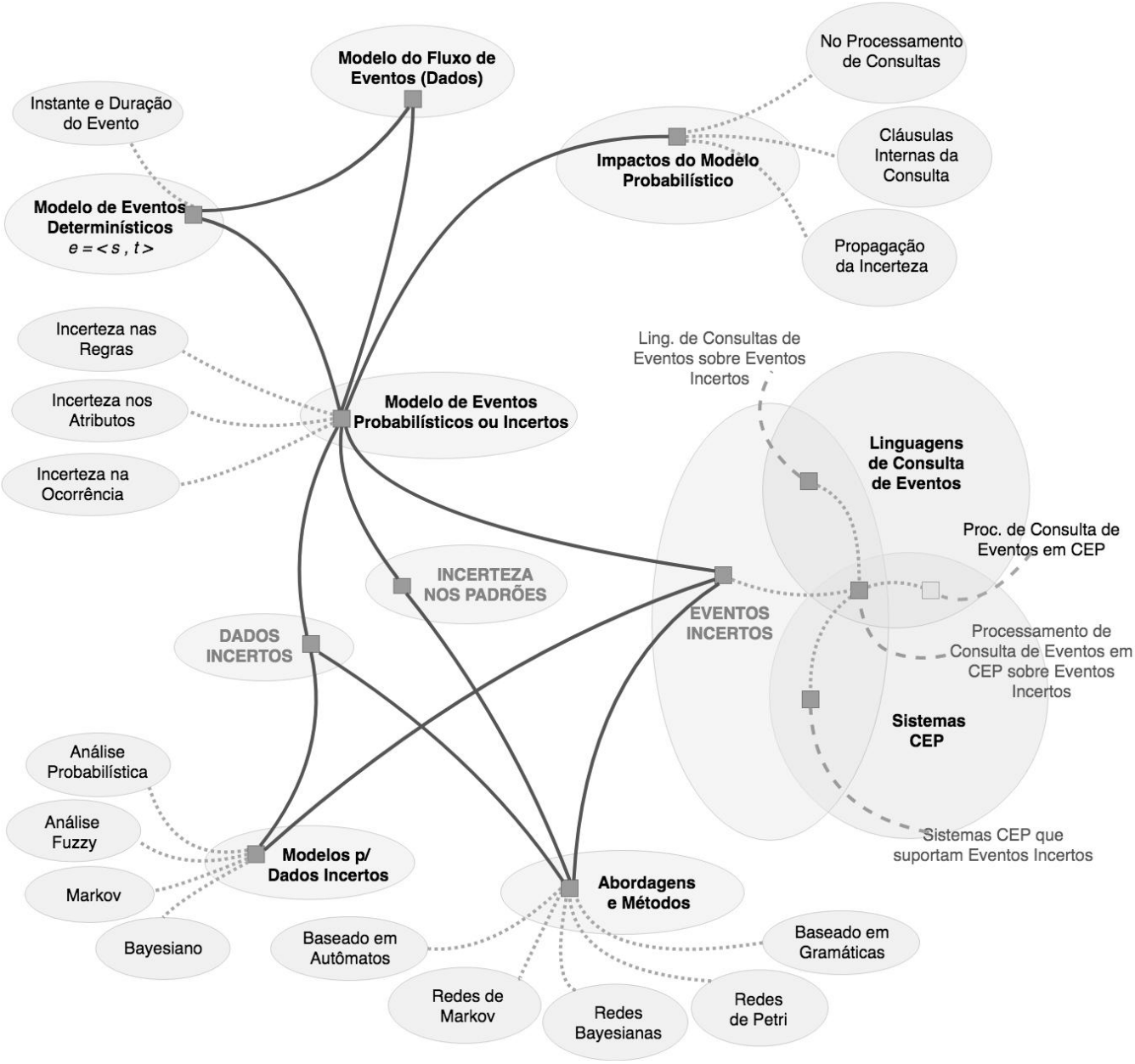


Figura 3.6: Mapa Geral de Tópicos do Domínio de Incerteza em CEP.

Logo, como ilustra a Figura 3.6, um mapa geral de tópicos do domínio de incerteza em CEP foi elaborado com o objetivo de conhecer e entender como se relacionam os principais conceitos, problemas e abordagens sobre o tema e o domínio desta pesquisa. A fusão (ou *merging*) de mapas de tópicos apresenta as seguintes vantagens [59]: a união de características originais para fornecer uma visão unificada do todo (domínio), integração de ”ilhas” de informação até então desconectadas dentro de um domínio e acesso consolidado a todas as informações relacionadas. Além disso,

o *merging* de mapas suporta problemas de denominação e identificação global de termos, e problemas de sinônimos e homônimos, onde respectivamente, um tópico (ou assunto) pode possuir vários nomes e um mesmo nome pode ser empregado para diferentes assuntos. Observa-se no mapa (Fig. 3.6) que os termos principais do tema (fonte vermelha) foram unificados e relacionados de forma particular com os tópicos apresentados nos *surveys*.

O conhecimento adquirido com base no mapa geral de tópicos permitiu a seleção dos trabalhos relacionados, que formam um conjunto de abordagens fundamentadas na Teoria de Probabilidades, Redes Bayesianas, Lógica Fuzzy, Cadeia de Markov e Teoria de Dempster-Shafer. Cada trabalho é apresentado a seguir, com ênfase nos seus **objetivos**, **tipos de incertezas** abordados, pontos relevantes da **solução**, síntese dos **resultados** e algumas **limitações** identificadas.

3.2 Abordagens fundamentadas na Teoria de Probabilidade e em Redes Bayesianas

Nesta seção são apresentados trabalhos que possuem como fundamentação teórica a Teoria de Probabilidade e Redes Bayesianas.

3.2.1 *Model for Reasoning with Uncertain Rules in Event Composition Systems* [81] (2005)

Consoante com o esforço desta pequisa até onde se conhece, o primeiro modelo proposto para lidar com a incerteza no processamento de eventos é descrito por Wasserkrug et al. [81] e este modelo foi estendido em [84], [83] e [85]. O trabalho inicial de Wasserkrug et al. [81] propõe uma representação formal dos eventos e da composição de eventos, possibilitando a inferência (raciocínio) de eventos complexos mesmo sob o cenário de incerteza. São considerados dois tipos de incerteza neste trabalho. A primeira é a incerteza causada pela imprecisão de informações sinalizadas pela fonte do evento, por exemplo, sensores defeituosos ou imprecisos. O segundo tipo de incerteza é inerente às relações entre eventos. Por exemplo, um sistema que monitora a compra e venda de ações é obrigado a responder automaticamente na

ocorrência de negociações ilegais de ações. Nesse caso, por mais que se consiga relacionar evidências que indiquem negociações fraudulentas, o sistema não pode ter certeza se a negociação ilegal de fato ocorreu (não determinismo). O melhor que pode ser alcançado é uma "suspeita" baseada na quantificação de alguma medida de probabilidade em relação à ocorrência do evento de negociação ilegal. Vale observar que este trabalho de Wasserkrug et al. [81], usa como base a teoria de probabilidade para representar as duas incertezas associadas e para quantificar a probabilidade de ocorrência dos eventos derivados na solução.

A solução propõe inicialmente um **modelo de representação do evento**, por exemplo, um evento que cota o preço de $100 para uma ação da empresa IBM no horário 10:45h é representado por $E = \{10:45; IBM; 100\}$. Além disso, o sistema pode atribuir probabilidades para os eventos ocorridos da seguinte forma, a probabilidade do evento E ter ocorrido é de 0.3, representado por $Pr(E = \{10:45; IBM; 100\}) = 0.3$ ou por simplificação, $Pr(E = \{Occurred\}) = 0.3$. A probabilidade do evento não ter ocorrido, neste caso, é $Pr(E = \{notOccurred\}) = 0.7$.

O trabalho de Wasserkrug et al. [81] define também um **modelo de representação da regra** $r =< sel_r, pattern_r, eventType_r, prob_r >$ constituído de vários elementos que possibilitam respectivamente a definição de consultas, padrões, tipos de eventos e a definição de um valor de probabilidade da regra. Para ilustrar os elementos do modelo de representação de regras no cenário do mercado de ações, toma-se como exemplo uma regra r usada para reconhecer uma operação ilegal de negociação de ações que significa a venda de uma ação, seguida pela compra da mesma ação, pelo mesmo cliente em um curto espaço de tempo. Inicialmente a regra r é composta pelo elemento $sel_r = (\varepsilon_1 \in stockSell,\ \varepsilon_2 \in stockPurchase)$, que representa a seleção de quaisquer dois eventos, o primeiro pertencente ao tipo $stockSell$ (venda de ação) e o segundo pertencente ao tipo $stockPurchase$ (compra de ação). Além disso, uma regra pode ser composta por um padrão, como segue no exemplo:

$$pattern_r = (\varepsilon_1.occT \leq \varepsilon_2.occT \leq \varepsilon_1.occT + 5) \wedge$$
$$(\varepsilon_1.stockTicker = \varepsilon_2.stockTicker) \wedge (\varepsilon_1.customerID = \varepsilon_2.customerID)$$

O padrão $pattern_r$ é aplicado sobre os elementos (ε_1 e ε_2) selecionados por sel_r, e que deve obedecer as seguintes restrições: $occT$ representa a relação temporal entre os eventos obedecendo uma ordem cronológica do tempo de ocorrência

dos eventos. Em seguida, no $pattern_r$ são verificados os tipos dos eventos ($\varepsilon_1.stockTicker = \varepsilon_2.stockTicker$) e as identificações dos clientes ($\varepsilon_1.customerID = \varepsilon_2.customerID$).

Finalmente, uma regra r pode receber um valor de probabilidade $prob_r = 0.7$, refletindo que a venda de uma ação, seguida pela compra da mesma ação, pelo mesmo cliente em um curto espaço de tempo, recebe uma indicação de probabilidade 0.7 de negociação suspeita.

Com base nos modelos acima, uma Rede Bayesiana (RB) é construída à medida que os eventos ocorrem. Nesse passo da solução, o objetivo é calcular as probabilidades de ocorrências de interesse com base nas probabilidades definidas pela RB construída. Resumidamente, cada nó da rede é composto por um evento (ex: E_1) que adicionalmente recebe os valores de probabilidade de ocorrência e não ocorrência, por exemplo $Pr(E_1 = \{occurred\}) = 0.6$ e $Pr(E_1 = \{notOccurred\}) = 0.4$. À medida que novos eventos ocorrem (E_2, E_3, ...), é verificado se os eventos relacionados obedecem a regra estabelecida segundo o modelo de regras. Se a regra é satisfeita, a probabilidade do evento derivado E_r é atribuída e o mesmo é adicionado à RB, como ilustra a Figura 3.7 [81].

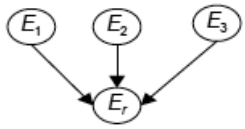


Figura 3.7: Rede Bayesiana Construída [81]

Os resultados deste trabalho apresentam modelos de representação formais (do evento e da regra) para a composição de eventos em situações de incerteza. A representação das incertezas inerentes aos eventos e a composição de eventos utiliza como base a teoria de probabilidades e redes Bayesianas.

3.2.2 *Efficient Processing of Uncertain Events in Rule-Based Systems [85]* (2012)

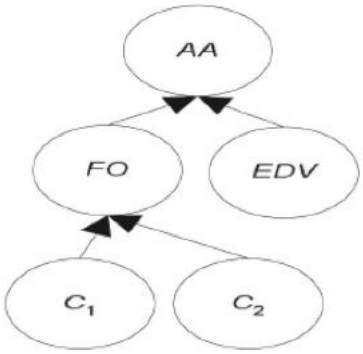
O seguinte trabalho de Wasserkrug et al. [85] estendido de [81], tem como objetivo fornecer um mecanismo eficiente e preciso para raciocínio sobre eventos incertos. Dentre os focos deste trabalho, o primeiro é prover eficiência na derivação

de eventos sob incerteza considerando uma grande quantidade de eventos recebidos. O segundo foco está nas probabilidades associadas aos eventos, que devem ser corretamente capturadas e representadas.

O cenário apresentado neste trabalho pertence ao domínio de sistemas de vigilância de surtos epidêmicos e ataques bioterroristas. Para identificar e quantificar um surto ou a severidade de um ataque, as fontes de informação usadas são bancos de dados com registros de vendas de medicamentos, atendimentos em hospitais e conhecimento especialista. Considera-se o seguinte exemplo, o aumento na venda de medicamentos para gripe em farmácias, aumento nas chamadas para atendimento de emergência, aumento nos registros de ausências escolares e aumento nas queixas de tosse de pessoas nos departamentos de emergência podem servir como indicadores para um surto epidêmico. Portanto, responder rapidamente a tal surto implica no rápido reconhecimento se os problemas anteriores ao surto ocorreram, o que configura uma tarefa difícil já que não existem indicações diretas de tais problemas. Este fato ilustra a incapacidade de determinar com certeza se um evento realmente ocorreu, dada a informação disponível. Nesse sentido, o desafio é derivar eventos com base nas fontes de dados que geralmente fornecem informações insuficientes (incompletas) para determinar a ocorrência de eventos complexos. Dessa forma, o primeiro tipo de incerteza envolvida neste trabalho é a **incerteza na derivação dos eventos**, que decorre da inabilidade de derivar eventos com certeza a partir da informação disponível (incerteza na ocorrência do evento derivado). O segundo tipo é a **incerteza na fonte do evento**, resultante de informações imprecisas ou incompletas fornecidas pela fonte do evento.

A solução de Wasserkrug et al. [85] adota os mesmos modelos de representação do evento e da regra do trabalho anterior de Wasserkrug et al. [81]. Além disso, um algoritmo é usado para a derivação de eventos incertos, que em suma funciona da seguinte maneira: dado um conjunto de eventos em um tempo t, é construída uma rede Bayesiana que representa corretamente as probabilidades no instante t. Os novos eventos têm a probabilidade calculada usando a rede Bayesiana. Para ilustrar tal situação, a Figura 3.8 descreve a parte qualitativa de uma rede Bayesiana que define um espaço de probabilidade em cinco variáveis: C_1, C_2, FO, EDV e AA, representadas como nós. Nos quais C_1 e C_2 (*Counter sales of cough medication*) correspondem aos eventos de aumento nas vendas de medicamento

para tosse em farmácias, FO (*Flu Outbreak*) corresponde o evento de surto de gripe, EDV (*Emergency Deparment Visits*) corresponde ao evento de um grande volume de entradas no departamento de emergência com queixas respiratórias e AA (*Anthrax Attack*) evento de ataque de antraz (bioterrorista). Os arcos indicam as dependências probabilísticas diretas entre os nós da rede.



	AA=true	*AA=false*
FO=true, EDV=true	0.3	0.7
FO=false, EDV=true	0.8	0.2
FO=true, EDV=false	0.05	0.95
FO=false, EDV=false	0.1	0.9

Figura 3.8: Exemplo gráfico da Rede Bayesiana (esquerda) e uma CPT (à direita) [85]

A parte quantitativa da rede é a tabela à direita na Figura 3.8 ou CPT (*Conditional Probability Table*), que define as probabilidades de cada variável com base nas dependências probabilísticas. A tabela mostra a relação quantitativa entre AA, FO e EDV da seguinte forma: para cada possibilidade de FO e EDV, a tabela fornece a probabilidade condicional de cada valor de AA. Por exemplo: $Pr(AA = true|FO = false, EDV = true) = 0.8$, cuja leitura indica que a probabilidade de um ataque de antraz ser verdadeiro é de 80%, dado que o surto de gripe é falso e o aumento nos atendimentos de emergência com queixas respiratórias é verdadeiro. Assim, com a chegada de um novo evento, as regras são verificadas, eventos são selecionados satisfazendo a semântica da regra e a rede é concluída, a probabilidade no momento t é calculada com base nas probabilidades descritas por toda a rede. Além disso, um algoritmo baseado em amostragens também é sugerido como mecanismo de aproximação das probabilidades dos eventos derivados com foco na performance da solução. A ideia é fornecer uma estimativa mais precisa de probabilidade à medida que o número de amostragens aumenta (vide detalhes [85]).

O resultado do trabalho apresenta como contribuição a introdução de um mecanismo formal e *framework* para gerenciar e derivar eventos sob condições de incerteza. A solução mostrou performance comparável a um sistema de processamento de eventos determinísticos.

3.2.3 *Introducing uncertainty in complex event processing: model, implementation, and validation [17]* (2015)

Um dos trabalhos mais contundentes para lidar com o problema da incerteza no processamento de eventos complexos é o trabalho do Cugola et al. [17], que propõe um modelo para lidar com a incerteza, além de validar sua solução, denominada CEP2U (*Complex Event Processing under Uncertainty*). A motivação de Cugola et al. [17] é justificada pela investigação das abordagens que utilizam *engines* CEP e têm frequentemente foco na performance e baixo *delay* (atraso) do processamento de eventos. Este fato tem levado à modelagem de regras mais simplificadas para capturar os problemas e a complexidade em cenários reais de aplicações onde os eventos ocorrem. O autor destaca que uma das principais limitações de tais abordagens CEP é a incapacidade de considerar, modelar e propagar a incerteza que existe na maioria das suas aplicações. Assim, as incertezas consideradas neste trabalho são de dois tipos. O primeiro é a **incerteza do dado originada na fonte produtora do evento**, como por exemplo, o erro introduzido por um conjunto de sensores distribuídos que medem temperatura e umidade em uma grande área para realizar a previsão do tempo. O segundo tipo é a **incerteza nas regras que derivam os eventos complexos**, ou seja, imprecisão nas regras, que não refletem completamente o comportamento do ambiente monitorado. Por exemplo, uma regra que detecta um evento de incêndio assumindo como um dos seus critérios a presença de fumaça. Tal regra eventualmente pode gerar falsos positivos se um fumante, próximo a um sensor de detecção de fumaça, acidentalmente disparar um alerta de incêndio.

A solução proposta neste trabalho é aplicada no cenário de túneis de veículos, que são constantemente monitorados por vários tipos de sensores para detectar possíveis problemas, como por exemplo obstruções. *Tunnel Ventilation System* (TVS) deve garantir a segurança nos túneis detectando eventos de mal funcionamento (***TVS Malfuncioning***) com base em sensores de temperatura, concentração de oxigênio, além da análise de setores específicos do túnel (km), intervalos de tempos e tráfego de veículos no túnel. Por exemplo, um evento de mal funcionamento ocorre quando a concentração de oxigênio for menor que 18% e a temperatura maior que 30°C no mesmo setor do túnel, nos últimos 5 min.

CEP2U modela incerteza nos eventos usando a teoria de probabilidade e para cada evento considera duas formas de incerteza:

Incerteza no Conteúdo do Evento - relacionada aos valores dos atributos do evento que são afetados por algum grau de incerteza, por exemplo causada pela imprecisão que afeta os sensores. Neste caso, para cada atributo é associado um erro de medição ($X_i' = X_i + \epsilon_i$), onde X_i é o valor real medido na ausência de erro e ϵ_i é a medida de erro. CEP2U assume que a distribuição de probabilidade (pdf) de ϵ_i é conhecida e depende, por exemplo, do ruído nas fontes ou imprecisões na medição dos sensores. Vale observar que o erro de medição pode ser provido pela fonte. Por exemplo, a estimativa de erro do sensor pode ser conhecida ou informada pelo fabricante, e tal taxa pode ser anexada às notificações dos eventos pelo próprio sensor. Ou ainda espera-se que especialistas do domínio possam prover tais erros de medições e que a informação seja integrada às notificações dos eventos antes do processamento.

Incerteza na Ocorrência do Evento - modelada através de uma estimativa da probabilidade de ocorrência do evento. Quando a ocorrência é assumida como certa, tem probabilidade com valor um (1) e quando a ocorrência não é certa, tem uma probabilidade menor que um. CEP2U assume que todos os eventos primitivos tenham ocorrido (probabilidade 1).

Para ilustrar as incertezas descritas, segue a notificação de um evento de temperatura: **Temp@13 %1 (km = <16.2, U(-1, 1)>, value = <31.8, N(0,1)>)**, onde o evento **Temp** no instante **13s** ocorreu (**%1**, prob. 1 indica certeza na ocorrência), com valor **31.8°C** e no setor **16.2 km**. Ambos os atributos **km** e **value** têm associadas as incertezas modeladas através das distribuições de seus erros de medição, uma uniforme (U) para **km** e a outra normal (N) para **value**.

Propagação da Incerteza dos Eventos - as incertezas nos eventos primitivos são propagadas para os eventos derivados. Logo, Cugola et al. [17] lida com a propagação da incerteza da seguinte forma. Inicialmente é considerada uma regra simplificada (**Rule R1**) que detecta um evento de mal funcionamento toda vez que um evento **Temp** é detectado, satisfazendo as duas condições a seguir, (km < 17.1) e (valor > 30).

Assim **Rule R1: define TVS_Malfun( ) from Temp(km<17.1 and value>30)**.

Quando um evento **Temp** ocorre no sistema, é calculada a probabilidade de cada restrição ser satisfeita a partir das distribuições de probabilidade de cada atributo. Por exemplo, dados a regra e o evento **Temp** recebido com os seguintes valores medidos e erros associados:

$Evento:$ **Temp@13 %1 (km = <16.2, U(-1, 1)>, value = <31.8, N(0,1)>)**

$Regra:$ **define TVS_Malfun( ) from Temp(km<17.1 and value>30)**

a probabilidade para que a primeira restrição na regra acima seja satisfeita é: P(X_{km} < 17.1), onde X_{km} é o valor real e desconhecido do atributo **km** e X'_{km} é o valor observado com a incerteza associada (ϵ_{km}). Sabe-se que:

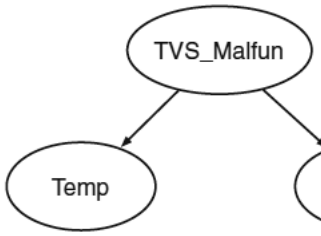
$$\begin{aligned} X'_{km} &= X_{km} + \epsilon_{km} \\ X_{km} &= X'_{km} - \epsilon_{km} \sim U(16.2 - 1, 16.2 + 1) \\ X_{km} &= U(15.2, 17.2) \end{aligned}$$

Assim, a probabilidade de X_{km} ser menor que 17.1, ou $P(X_{km} < 17.1)$, corresponde ao cálculo de $P(U(15.2, 17.2) < 17.1) = 0.9$. Similarmente para o atributo **valor**, tem-se: $X_{value} = X'_{value} - \epsilon_{value}$. A probabilidade de que o valor seja maior que 30, ou $P(X_{value} > 30)$, é equivalente ao cálculo de $P(N(31.8, 1) > 30) = 0.964$ (para detalhes dos cálculos, vide [17]).

Finalmente, CEP2U assume que os valores dos diferentes atributos são independentes. Com isso, a probabilidade geral de que o evento **Temp** satisfaça ambas as restrições na regra é o produto da probabilidade de cada restrição ser satisfeita, ou seja, $0.9 \cdot 0.964 = 0.868$. Como essas são as únicas restrições na regra, essa também é a probabilidade de ocorrência do evento TVS_Malfun gerado e que o CEP2U associa a este evento derivado.

Incerteza nas Regras - CEP2U modela a incerteza derivada das regras usando Rede Bayesiana (RB), escolhida devido à sua maneira de representar as dependências entre conceitos e relação causal entre eventos primitivos e derivados. Na prática, CEP2U traduz automaticamente uma regra em uma RB correspondente que é uma rede inicialmente simples. Esta RB é refinada e "enriquecida" por um especialista do domínio. Ou seja, neste passo a RB é atualizada com informações ou detalhes relevantes que podem influenciar a ocorrência dos eventos na rede e evitar deduções erradas. Por fim, as probabilidades dos eventos derivados são computadas considerando as informações inseridas na rede.

Para ilustrar a situação da tradução e o refinamento de uma RB a partir de uma regra, assume-se um evento **TVS_Malfun** detectado por uma Regra R2 que é disparada quando ocorre a observação de alta temperatura e baixa concentração de oxigênio. A Figura 3.9 ilustra uma RB resultante da descrição da regra. O processo de **tradução** define como a ocorrência de um evento derivado (**TVS_Malfun**) pode ser detectada a partir da observação ou composição de um ou mais eventos primitivos. Ou seja, modela uma dependência causal entre os eventos derivados e os eventos primitivos.



	Temp	
TVS_Malfun	YES	NO
YES	1.0	0.0
NO	0.0	1.0

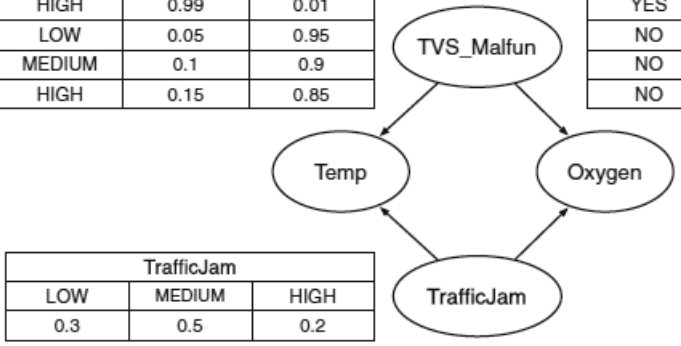
	Oxygen	
TVS_Malfun	YES	NO
YES	1.0	0.0
NO	0.0	1.0

Figura 3.9: Rede Bayesiana gerada a partir da regra [17]

CEP2U traduz a regra automaticamente para a RB (Fig. 3.9) que inclui um nó para cada evento (primitivo ou composto) e um arco (uma relação causal) conectando o evento composto a cada evento primitivo. A tradução automática assume que o evento **TVS_Malfun** determina a ocorrência dos eventos primitivos (**Temp** e **Oxygen**) com probabilidade igual a 1.

		Temp	
TVS_Malfun	TrafficJam	YES	NO
YES	LOW	0.94	0.06
YES	MEDIUM	0.96	0.04
YES	HIGH	0.99	0.01
NO	LOW	0.05	0.95
NO	MEDIUM	0.1	0.9
NO	HIGH	0.15	0.85

		Oxygen	
TVS_Malfun	TrafficJam	YES	NO
YES	LOW	0.95	0.05
YES	MEDIUM	0.97	0.03
YES	HIGH	0.99	0.01
NO	LOW	0.08	0.92
NO	MEDIUM	0.12	0.88
NO	HIGH	0.16	0.84



TrafficJam		
LOW	MEDIUM	HIGH
0.3	0.5	0.2

Figura 3.10: Enriquecimento da Rede Bayesiana gerada [17]

O passo de **enriquecimento** permite ao especialista do domínio editar a RB, ou seja, modificar a direção dos arcos, inserir novos nós e suas dependências causais em relação aos eventos da rede. A Figura 3.10 ilustra a RB inicial enriquecida. Neste caso, foi constatado pelo especialista que o engarrafamento (**TrafficJam**) no túnel é uma fator externo que causa o aumento da temperatura e baixa concentração de oxigênio. Dessa forma, um novo nó foi inserido representando tal fator e conectado aos eventos **Temp** e **Oxygen**. Em função do novo nó, as tabelas de probabilidades de **Temp** e **Oxygen** foram modificadas levando em consideração a presença de congestionamentos. **TrafficJam** pode assumir três valores: HIGH, MEDIUM e LOW; e a distribuição *a priori* desses valores foi adicionada à rede. Tais distribuições podem ser modificadas com o tempo, por exemplo, para melhor representar situações de cenários variáveis, como tendências sazonais.

Os resultados deste trabalho são avaliados sob dois aspectos no tratamento de incerteza em CEP. O primeiro é a avaliação do custo computacional da sobrecarga introduzida pelo gerenciamento de incerteza em CEP, que não será discutido nesta seção (vide detalhes [17]). O segundo aspecto é a avaliação dos benefícios de lidar com a incerteza como valor agregado ao evento. Para o autor, CEP2U é fácil de usar porque permite que especialistas capturem facilmente e em separado os tipos de incertezas a partir dos eventos e das regras, assim reduzindo a complexidade do gerenciamento das incertezas. A incerteza nas regras é modelada utilizando redes Bayesianas e o enriquecimento da rede permite incluir fatores externos de incerteza que não são capturados pelas regras na geração automatizada da rede.

3.2.4 *Event Processing Under Uncertainty [8]* (2012)

O trabalho de Artikis et al. [8] aponta a necessidade do tratamento de incertezas no processamento de eventos e propõe uma classificação das possíveis fontes de incerteza. Além disso, os autores reconhecem a necessidade de modelagem e propagação da incerteza, e propõem a teoria de probabilidades como uma possível base matemática para realizar tais tarefas. Para ilustrar a incerteza no processamento de eventos é apresentado um cenário de detecção de crimes.

O autor destaca as incertezas em sistemas de processamento de eventos com base em dois de seus trabalhos [9] [10] e algumas fontes de incertezas listadas

nos trabalhos são: fluxos de eventos incompletos, reconhecimento errôneo de eventos, anotações de eventos inconsistentes e padrões de evento imprecisos. Dadas tais fontes de incerteza, os tipos de incerteza são classificados da seguinte forma: (1) incerteza nos eventos de entrada (*input*), (2) caso não haja incerteza nos eventos de entrada, pode ocorrer incerteza somente no padrão de composição dos eventos, e (3) o último caso é a incerteza em ambos, na entrada e no padrão.

As propostas de Artikis et al. [8] para tratamento de incerteza são aplicáveis no cenário de detecção de crimes, mais especificamente sobre a modelagem de um sistema de vigilância visual (monitoramento) que deve ser capaz de detectar e rastrear pessoas sob uma ampla variedade de condições no ambiente. A origem do fluxo de eventos é baseada em imagens. Neste cenário, o sistema deve analisar ações e interações entre pessoas e/ou entre pessoas e objetos no ambiente para que alertas de crimes em tempo real sejam enviados para agentes de segurança. Na solução, a representação de um evento contém os atributos do domínio e adicionalmente possui o atributo ”certeza” relacionado à ocorrência do evento e que pode assumir um valor entre zero e um, *certainty* $[0, 1]$. Para expressar eventos que ocorrem durante um período de tempo e não em um ponto específico no tempo, é utilizada a noção de ”intervalo”, como ilustra a representação do evento na Tabela 3.1.

Attribute	**Type**
Event Name	String
Occurrence Time	Time-stamp/Interval
Detection Time	Time-stamp
Certainty	Double $(0, 1]$

Tabela 3.1: Representação do Evento [8]

Quando o tempo de ocorrência de um evento é incerto, como no caso do ”relato de um crime que ocorreu em algum momento entre 8h e 9h”, sua associação com um contexto temporal é inconclusiva. Para representar a incerteza de ocorrência do evento dentro de um intervalo de tempo, isto é, quando não se conhece o tempo exato de ocorrência do evento, entra em cena a noção dos atributos probabilísticos. Assume-se a probabilidade do evento ocorrer em cada ponto nesse intervalo de tempo e cada ponto do intervalo recebe a mesma probabilidade, tratando-se portanto da distribuição uniforme como meio natural para representar a incerteza neste caso.

O autor aborda a propagação da incerteza dos eventos de entrada para os eventos derivados, além de mecanismos de eliminação da incerteza nas expressões que definem as regras de derivação dos eventos. Por exemplo, filtrar "observações" que são suspeitas de serem um crime, usando a afirmativa *observation.crime_indication = true*, onde há incerteza no atributo Booleano *crime_indication* que é derivado de técnicas de visão computacional que não são totalmente precisas. Logo, a propagação da incerteza seria uma distribuição de probabilidade sobre os valores possíveis "true" e "false".

Os resultados do trabalho têm foco nos tipos de incertezas apresentados e que podem ser encontrados no processamento de eventos. São discutidas as possíveis maneiras de estender os sistemas tradicionais de processamento de eventos para lidar com as incertezas relatadas. O autor confirma a necessidade de desenvolver um *framework* para o processamento de eventos de forma eficiente, escalável e na presença de vários tipos de incerteza.

3.3 Abordagens fundamentadas da Teoria de Probabilidade

Esta seção apresenta um trabalho que possui fundamentação teórica puramente na Teoria de Probabilidade.

3.3.1 *Managing Measurement and Occurrence Uncertainty in Complex Event Processing Systems [56]* (2019)

O trabalho de Moreno et al. [56] apresenta uma proposta para incorporar e gerenciar diferentes tipos de incertezas que podem ocorrer em eventos e regras CEP. O conceito de confiança é introduzido como o grau de crença sobre a ocorrência de um evento, ou sobre a inferência de determinada regra. A teoria de probabilidade é utilizada para expressar a confiança e atribuir probabilidades para os eventos e as regras.

Em Moreno et al. [56], a atribuição de probabilidade para cada evento é feita através do atributo de confiança (*conf*). Dado um evento simples *e*, tal probabilidade

coincide com $(1-P_{fp}(e))$, onde $P_{fp}(e)$ é a probabilidade de um falso positivo[5] do evento em questão. Esta informação é normalmente obtida do fabricante do sensor ou fontes similares. O que também pode ser devido a redes de comunicação não confiáveis que duplicam pacotes ou outras causas. Em qualquer dos casos, o autor destaca a importância de levar em conta a informação de confiança explicitamente nos eventos.

A probabilidade do evento complexo é dada pelo produto dos eventos independentes, $P(e_1; ...; e_n) = P(e_1 \cdot ... \cdot e_n)$. Tais valores podem ser multiplicados pela probabilidade da regra P(R), quando estimada pelos usuários especialistas. P(R) é a confiança da regra representada por uma probabilidade que captura a possível imprecisão devido a suposições incompletas ou errôneas sobre o ambiente em que o sistema opera. Neste trabalho P(R) é uma constante. Outro multiplicador é a confiança do processo de comparação que decide se valores dos atributos dos eventos preenchem requisitos do padrão.

A avaliação do trabalho de Moreno et al. [56] tem maior foco na performance da solução, devido às operações necessárias para calcular as incertezas associadas aos eventos e às regras. Foram detectadas diferenças no tempo de execução de uma aplicação sem incerteza e com incerteza (a mesma aplicação recebendo eventos estendidos com valores de incerteza). A avaliação demonstrou *overheads* pequenos (entre 1,08s e 1,15s) em alguns testes. No pior dos casos, em um carga de processamento de 200K eventos com e sem incerteza, foi observada uma diferença de tempo pouco acima de 3 min.

Outro ponto da avaliação está relacionado ao uso de incerteza em eventos e regras com o objetivo de verificar se os resultados fornecidos para as aplicações são mais precisos. Para tal tarefa, são utilizados geradores de amostras de eventos imprecisos de entrada, assumindo distribuição normal para os valores de incerteza. Tais amostras tentam representar eventos reais que seriam obtidos em um ambiente com incerteza e dispositivos de medição imprecisos. Além das variações de incerteza, são simuladas faixas de valores dos atributos dos eventos. Em qualquer um dos casos, os resultados de acurácia não parecem dificultar a adição de incerteza na solução proposta.

[5]Eventos ausentes no fluxo e que realmente aconteceram são referidos como falsos negativos (FN); eventos no fluxo que foram inseridos erroneamente são falsos positivos (FP).

3.4 Abordagem fundamentada na Lógica Fuzzy

Esta seção apresenta um trabalho que possui fundamentação teórica na Lógica Fuzzy.

3.4.1 *FSCEP: A New Model for Context Perception in Smart Homes [35]* (2016)

O trabalho de Jarraya et al. [35] propõe um modelo de raciocínio e representação de eventos integrando conhecimento do domínio e lógica *fuzzy*, denominado FSCEP (*Fuzzy Semantic Complex Event Processing*). Este modelo tem por objetivo lidar com múltiplas dimensões de incerteza provenientes de dados de sensores no ambiente de IoT, mais especificamente em uma aplicação de *smart homes*.

A autora considera que a incerteza tratada no trabalho pode ser decomposta em múltiplas dimensões ou facetas. Por exemplo: (1) *Freshness*: se o dado é muito antigo, pode estar desatualizado; (2) *Precision*: o dado impreciso pode estar correto ainda que inexato, por exemplo, diferentes precisões para uma localização, sensor GPS (*outdoor*) com menor precisão ou sensores de presença em ambientes (*indoor*) com maior precisão; (3) *Contradiction*: dados que fornecem informações contraditórias, por exemplo, um sensor localiza um usuário no ambiente **A** enquanto outro sensor o localiza no ambiente **B**.

Para ilustrar a solução, é apresentado um cenário onde uma pessoa (Monika) vive em uma *smart home* equipada com vários sensores e atuadores. Em particular, a sala de estar e o escritório possuem *beacons* (que localizam o *smartphone* no ambiente), sensores de pressão em cadeiras, sensores de movimento e uma câmera. Monika está trabalhando em seu escritório onde é detectada pela câmera, no entanto ela deixou seu *smartphone* na sala de estar onde também é detectada pelo *beacon*. Neste caso, duas interpretações da localização de Monika são possíveis a partir dos eventos: ela pode estar na sala de estar ou no escritório. Notificações de outros sensores podem tornar a detecção mais contraditória, considerando a imprecisão dos sensores, má calibração, disparos inesperados de notificações de presença, deslocamento entre ambientes, etc.

Em uma abordagem CEP "pura", apenas uma interpretação seria mantida para o evento complexo de saída. Em FSCEP, várias interpretações podem ser mantidas. No exemplo considerado, as localizações em ambos ambientes são mantidas com um valor de confiança para cada uma. A solução propõe um índice de "confiança" atribuído para cada sensor (atribuído pelo especialista do domínio) cujos valores são usados para calcular a confiança dos eventos complexos resultantes. Para exemplificar a noção de **confiança** em Jarraya et al. [35], o mecanismo de detecção da câmera (através da análise de imagens) é mais confiável do que o mecanismo de detecção dos sensores de presença (captura de movimentação). Observa-se que a definição do valor de confiança dos sensores está fora do escopo de Jarraya et al. [35].

Dentro do modelo FSCEP vale destacar alguns pontos. Inicialmente, um evento simples é "enriquecido" com informações do sensor e do domínio, em um processo chamado de semantização do evento (*event semantization*). Como ilustra a Figura 3.11, o evento semântico é representado por triplas RDF[6] (*Resource Description Framework*), por exemplo, o evento recebido pelo *smartphone* indica que o mesmo está localizado na sala (*phone:MonikaNokia, islocatedIn, Livingroom*) e tem valor de confiança (*phone:MonikaNokia, hasTrust, 0.6*).

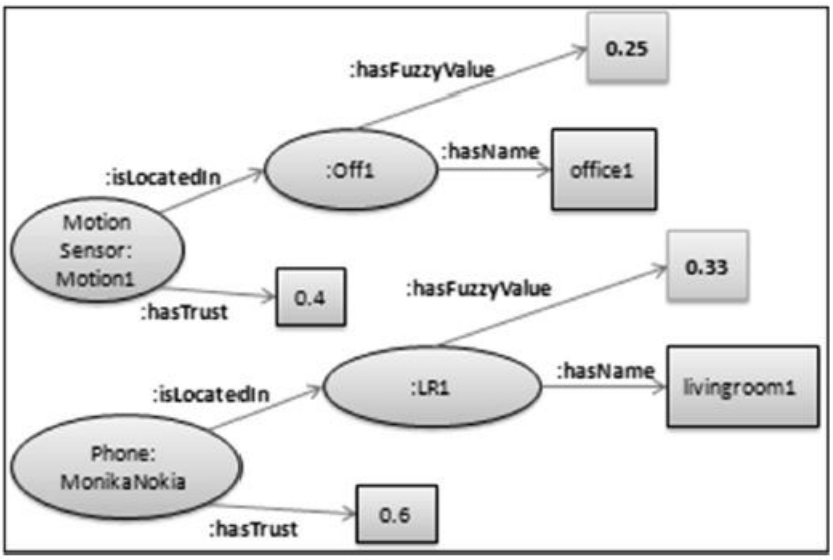


Figura 3.11: Representação do Evento em RDF [35]

Os valores fuzzy (cor laranja, Fig. 3.11) são calculados em um processo de fuzzificação, cuja base são valores de confiança dos sensores e dados de contexto que tipificam os sensores (presença, pressão, movimento). De modo geral, a

[6]https://www.w3.org/RDF/

base do cálculo relaciona os eventos de interesse sobre todos os eventos ocorridos nos ambientes, que no fundo usa o princípio da abordagem frenquentista de probabilidade. Exemplificando, dado que na **sala de estar** ocorreu um evento de presença a partir do *smartphone* (grau de confiança: 0.4) e no **escritório** ocorreu um evento de presença a partir da câmera (grau de confiança: 0.8), nesta situação, o peso de confiança (*Trust Weights*) dos sensores de presença calculado para a sala de estar $WT_{livingroom} = 0.4/(0.4 + 0.8) = 0.33$ e para o escritório $WT_{office} = 0.8/(0.4 + 0.8) = 0.67$. Isso significa que o usuário está realizando uma atividade na sala de estar com 0.33 de confiança e no escritório com 0.67 de confiança. O resultado é um evento fuzzy semântico do tipo *Fuzzy Semantic Complex Event*, ou $FSCE_{presence} = \{[livingroom; 0.33], [office; 0.67]\}$.

O trabalho de Jarraya et al. [35] apresenta como contribuição uma extensão CEP para lidar com múltiplas dimensões de incerteza que foram tratadas da seguinte forma: *Freshness*: para manter os dados de contexto atualizados, utilizou-se a própria tecnologia CEP. No exemplo, a localização do usuário é determinada imediatamente através da posição dos sensores assim que são disparados. *Precision*: no processo de semantização são adicionados ao evento a localização e o valor de confiança do sensor (traduzido em precisão de acordo com o tipo e mecanismo de detecção do sensor). *Contradiction*: neste caso o processo de fuzzificação proposto permite múltiplas interpretações do valor fuzzy resultante, resolvendo contradições e ambiguidades.

3.5 Abordagem fundamentada em Cadeias de Makov

Esta seção apresenta um trabalho que possui fundamentação teórica em Cadeias de Makov.

3.5.1 *Complex Event Processing under Uncertainty Using Markov Chains, Constraints, and Sampling - [62]* (2018)

O trabalho de Rincé et al. [62] propõe um método para estimar a probabilidade do reconhecimento de CEs (*Complex Event*) em um período razoável de tempo a partir de um fluxo de eventos incertos ou LLE (*Low Level Events*, como

definido pelo autor). O trabalho aborda o tipo de incerteza no evento, quando a detecção dos eventos não é garantida (os eventos podem ser perdidos ou detectados incorretamente).

A solução propõe um modelo baseado em um formalismo de cronicidade dos eventos (*chronicle formalism*), que descreve um evento complexo (CE) como a composição sequencial de eventos simples (LLE). Além disso, utiliza um conjunto de operadores baseados no tempo e intervalos de duração dos eventos. O cálculo da probabilidade estimada de um CE é feito utilizando técnicas baseadas na Cadeia de Markov. Como ilustra a Figura 3.12, o conjunto de operadores pode associar dois eventos (C1 e C2) baseados nos seus tempos de ocorrência e duração.

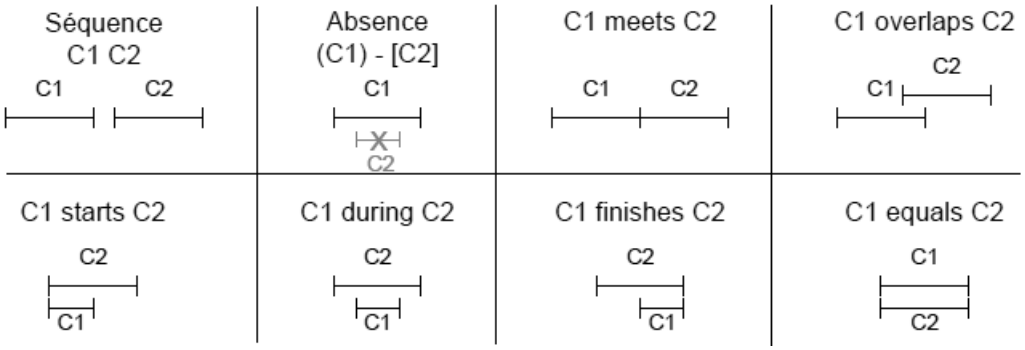


Figura 3.12: Operadores baseados em Intervalos de dois eventos

Consequentemente, é possível reconhecer um CE levando em consideração a duração dos eventos no fluxo transmitido. A Figura 3.13 ilustra o reconhecimento de um padrão (ou *chronicle*, como indicado pelo autor), onde um evento A precede dois eventos B, sem qualquer evento C entre os dois eventos B, ou seja, $A((BB) - [C])$.

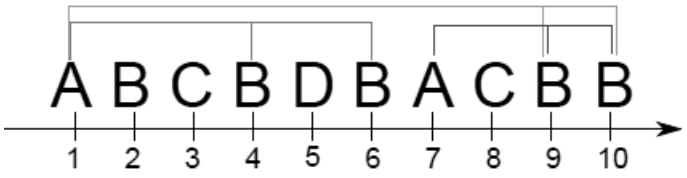


Figura 3.13: Todos os reconhecimentos de $A((BB) - [C])$

Um processo é proposto para estimar a probabilidade de reconhecimento de um padrão, que pode ser visto como uma consulta sobre um fluxo de eventos. O modelo de Markov precisa representar mudanças de estado do sistema observado-o sob condições de incerteza. Para ilustrar tal processo, um modelo é composto de dois subsistemas, o primeiro `Hurricane` (Furacão) que pode estar nos estados `Hurr` ou `Calm`,

significando a existência atual de um furacão ou não. O segundo é o `Alarm` que pode estar nos estados (`ON/OFF`) significando que o alarme de furação foi disparado ou não. Entretanto, o alarme não é completamente seguro e pode ser acionado sem furacão ou não ser disparado durante um furacão. Nessa situação de incerteza, as probabilidades de transição são fornecidas como ilustra a Figura 3.14.

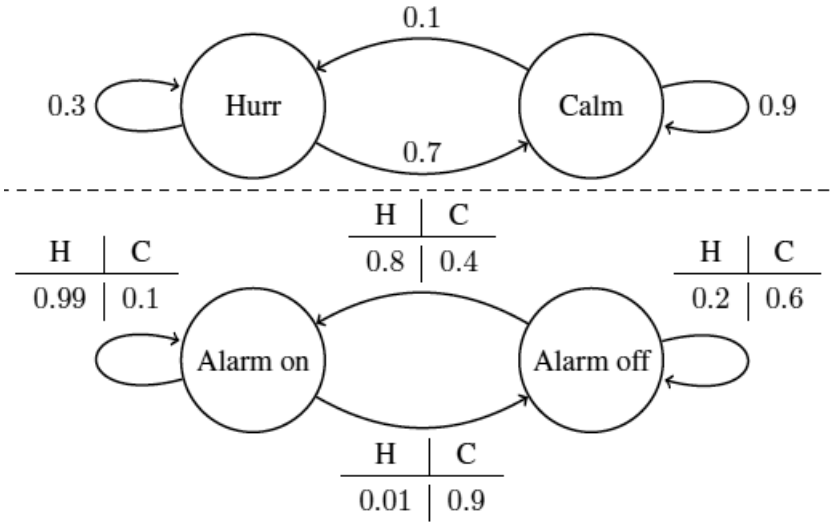


Figura 3.14: Modelo e probabilidades de transição

O modelo foi transformado em uma cadeia de Markov que combina os estados dos dois subsistemas, resultando na matriz dada na Figura 3.15.

t \ $t+1$	C & Off	C & On	H & Off	H & On
C & Off	0.54	0.36	0.02	0.08
C & On	0.81	0.09	0.001	0.099
H & Off	0.42	0.28	0.06	0.24
H & On	0.63	0.07	0.003	0.297

Figura 3.15: Cadeia de Markov (C:Calm, H:Hurr, Off:Alarm off, On:Alarm on)

Adicionalmente, uma técnica baseada em amostragens é sugerida para estimar a probabilidade de ocorrência de um padrão, contando o número de reconhecimentos em todas as amostras. Exemplos de padrões que podem ser verificados: *((H H) – [C]) equals ((off off) – [on])*, onde descreve duas ocorrências de furacão registradas, mas o alarme não é disparado. Ou ainda: *H during ((off off) – [on])*, que afirma que, durante um intervalo de tempo (*during*), o alarme não foi disparado enquanto ocorria um furacão.

Quanto aos resultados, o autor sugere técnicas para diminuir o tempo e o custo computacional da solução.

3.6 Abordagem fundamentada na Teoria de Dempster-Shafer

Esta seção apresenta um trabalho que possui fundamentação teórica na Teoria de Dempster-Shafer.

3.6.1 *Event Modelling and Reasoning with Uncertain Information for Distributed Sensor Networks [47]* (2010)

O trabalho de Ma et al. [47] propõe uma modelagem de eventos e raciocínio, a partir de múltiplas fontes de informação, integrando conhecimento do domínio e a Teoria de Dempster-Shafer (TDS) para lidar com a incerteza e informações incompletas no fluxo de eventos. Esta linha de trabalhos do autor surgiu em [49], a partir da demanda por sistemas CCTV (Closed-Circuit TeleVision) de vigilância em ônibus para detectar ameaças, evitar ataques terroristas e vandalismos crescentes nesse cenário. Em [48] houve alguns avanços da abordagem e até recentemente, em Ma et al. 2016 [50], a TDS é utilizada para lidar com o processo de análise de vídeos e resultados de algoritmos de classificação do perfil de gêneros (masc./fem.) de pessoas.

No modelo proposto em [47], o evento é definido como atômico (ocorrido ou não ocorrido) e instantâneo (ocorrido em um ponto específico no tempo, ou seja, um evento sem duração). O evento atômico não exclui a incerteza, por exemplo, uma pessoa que embarca em um ônibus pode ser homem ou mulher e este resultado pode apresentar incerteza. Entretanto, apesar do foco na detecção do gênero, observa-se que o ato de embarcar no ônibus é um evento atômico que acontece completamente ou não acontece. Assim, uma observação diz que algo aconteceu, mas a entidade que está sendo observada não é ainda completamente certa. Além disso, para usar o conhecimento do domínio, é introduzido o tipo de eventos do domínio, extraídos de opiniões de especialistas ou conhecimento prévio sobre o domínio da aplicação.

A representação do evento neste trabalho segue alguns pontos do modelo de representação da linha de trabalhos de Wasserkrug [81] [82] [84] [85], mas apresenta alguns conceitos diferentes. Formalmente, neste trabalho de Ma et al. [47], um evento e pode ser representado por: $e = (EType, occT, ID, rb, sig, v_1, ..., v_n)$, onde v_n são atributos do evento relacionados à aplicação. Por exemplo, o atributo *gênero* pode conter os valores {*masculino*}, {*feminino*}, {*masculino, feminino*} e {*obscurecido*}[7]. $EType$ descreve o tipo de um evento, ex. PBV (*Person Boarding Vehicle*). $occT$ é o momento de ocorrência do evento. ID é a identificação da fonte do evento, rb (do inglês *reliability*) é o grau de confiabilidade da fonte e sig é o grau de significância do evento, que nesta aplicação é um valor ou função sobre um evento. Por exemplo, neste cenário de vigilância de ônibus, um homem jovem embarca por volta das 22h em uma área com estatísticas de alta criminalidade. Este evento em si é mais significativo do que outro evento de embarque às 18h por uma senhora com idade avançada em uma área com baixa criminalidade. Em aplicações de vigilância, até 99% dos eventos são apenas eventos triviais, o que justifica um valor de significância embutido na representação dos eventos para facilitar o processamento subsequente.

A função de massa m atribui valores de massas para eventos agrupados pelo id da mesma pessoa (id. 3283), por exemplo, $m(male, 3283) = 0.4, m(female, 3283) = 0.3, m(\{male, female\}, 3283) = 0.2$ e $m(obscured, 3283) = 0.1$. O conceito de *Event Cluster* (EC) é introduzido para dar uma descrição completa de um fato observado com incerteza, a partir da perspectiva de uma fonte. Quando os ECs têm o mesmo tipo de evento (*EType*), o mesmo tempo de ocorrência (*occT*), porém com fontes diferentes (*source ID's*), significa que sensores de vários tipos ou modelos foram utilizados para monitorar a mesma situação. Portanto, a solução combina esses eventos que se referem ao mesmo fato observado de diferentes perspectivas aplicando a regra de combinação de Dempster (seção 2.2.3).

Na solução existe a etapa de inferência dos eventos a partir da definição de regras que o modelo sugerido preconiza da seguinte forma: a regra de inferência R é definida por $(LS, EType, Premise, Condition, m_{IEC})$, onde LS (*Life Span*) determina

[7]Neste trabalho, *obscurecido* não é o mesmo que {*masculino, feminino*}, pois pode indicar algum evento suspeito ou malicioso ocasionado pelo próprio passageiro tentando esconder o rosto. O atributo *gênero* pode ser visto como resultado de um programa de reconhecimento facial em que {*obscurecido*} significa que as informações sobre o reconhecimento facial não estão disponíveis.

o aspecto temporal da regra, um intervalo de tempo com início e fim. O $EType$ é o tipo do evento inferido e $Premise$ é um conjunto de $ETypes$ que são usados como pré-requisitos da regra. Por exemplo, um evento do tipo SAD (*Shout At Driver*) é derivado a partir dos eventos dos tipos PBV (*Person Boarding Vehicle*), PL (*Person Loiter*), PS (*Person Shout*), ou seja, para inferir um SAD é verificado se uma pessoa X entrou no ônibus, foi até a cabine do motorista e depois foi detectado um grito na cabine, $Premisse = \{PBV, PL, PS\}$. *Condition* especifica um conjunto de condições, relacionando atributos, para selecionar eventos específicos a partir do fluxo de eventos no processo de inferência. Finalmente, m_{IEC} é a função de massa para os eventos inferidos. O valor da massa é o grau conjunto de certeza de todos os eventos envolvidos na condição (*Condition*).

Relacionado aos resultados, vale destacar que a definição de evento neste trabalho é similar à definição Wasserkrug [81] onde eventos são considerados instantâneos e atômicos. Entretanto, uma regra em [81] só pode fornecer um único evento inferido com probabilidade $prob$, enquanto uma regra no modelo de Ma et al. [47] pode fornecer um conjunto de possibilidades com valores de massa, por exemplo, $m(\{male\}, ...) = 0.85$ e $m(\{male, female\}, ...) = 0.15$. As principais contribuições do trabalho de Ma et al. [47] se resumem a um modelo geral para representar incerteza, combinar eventos de múltiplas fontes e utilizar conhecimento do domínio como suporte ao processo de inferência.

3.7 Análise Comparativa dos Trabalhos Relacionados

Esta seção faz uma análise comparativa entre os trabalhos relacionados apresentados anteriormente com base no mapa geral de tópicos (Figura 3.6). Nesse viés, a Figura 3.16 destaca os principais conceitos e características consideradas dentro desse universo de trabalhos. Inicialmente foram apresentadas abordagens que tentam tratar a maior quantidade de tipos de incerteza em CEP (ex. nos atributos, ocorrência, regras, propagação e temporal). Além disso, foram apresentadas soluções aplicadas para domínios específicos com foco na diversidade das abordagens e métodos (ex: *Fuzzy, Markov, TDS*). Por fim, com exceção das primeiras abordagens (ou modelos para incerteza), buscou-se apresentar trabalhos mais recentes relacionados com o problema desta pesquisa e além daqueles trabalhos apresentados nos Surveys.

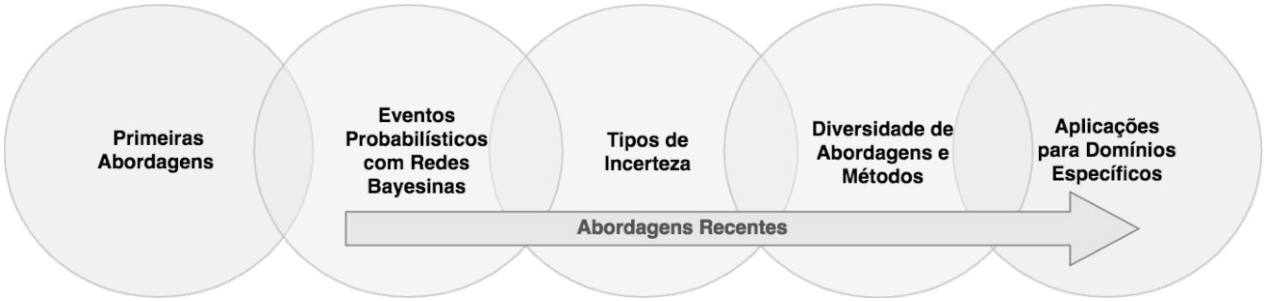


Figura 3.16: Conceitos e características dos trabalhos relacionados

Além disso, critérios são adotados com base no mapa geral de tópicos para realizar a comparação entre os trabalhos relacionados, como seguem:

1. **Fundamentação teórica**: especifica quais são as bases teóricas utilizadas em cada trabalho; Teoria de Probabilidade (TP), Redes Bayesianas (RB), Lógica Fuzzy (LF), Cadeia de Markov (CM), Teoria de Dempster-Shafer (TDS);

2. **Incerteza no evento**: especifica se o trabalho trata a incerteza no conteúdo (atributo) do evento ou ocorrência dos eventos;

3. **Incerteza nas regras**: especifica se o trabalho trata a incerteza nas regras (padrões) ou lida com a propagação da incerteza;

4. **Pressupostos de independência**: especifica se existem pressupostos de dependência ou independência entre os eventos processados.

5. **Filtragem de Eventos Relevantes**: especifica se existem mecanismos para o tratamento ou filtragem de eventos irrelevantes antes do processamento.

6. **Tecnologias CEP**: especifica se faz o uso de tecnologia CEP que facilita o armazenamento de consultas contínuas e permite uso de primitivas em tempo real para expressar conceitos de janelas de tempo, condições e correlações entre eventos.

Além dos critérios apresentados, são destacadas as principais vantagens e contribuições sinalizadas de cada trabalho. As limitações de cada trabalho também são identificadas, embora isso não signifique que a solução correspondente não possa superar tal limitação ou incorporar mecanismos que possam suprir as deficiências detectadas.

3.7.1 Discussão e Análise Comparativa

A seguir, os critérios de comparação são discutidos através da Tabela 3.2 sob a perspectiva de cada trabalho, com destaque para as limitações, para finalmente entender como uma abordagem baseada na TDS pode superar boa parte dos problemas levantados.

Trabalhos	Fundamentação Teórica	Incerteza no evento (Atrib./Ocorr.)	Incerteza nas regras ou propagação	Pressupostos de Independência	Filtragem de Eventos Relevantes	Tec. Proc. de Eventos Complexos
MRwURinECS (Wasserkrug2005)	TP e RB	Sim	Sim	Sim	Sim	Não
EPofUEinRBS (Wasserkrug2012)	TP e RB	Sim	Sim	Sim	Sim	Não
IUinCEP-MIV (Cugola2015)	TP e RB	Sim	Sim	Sim	Não	Sim
EPunderU (Artikis2012)	TP	Sim	Sim	Sim	Não	Sim
ManUnc (Moreno2019)	TP	Sim	Sim	Sim	Não	Sim
FuzzySCEP (Jarraya2016)	LF	Sim	Não	Sim	Não	Sim
CEPUwMarkov (Rince2018)	CM	Sim	Não	Não	Não	Sim
EMRwUIforDSN (Ma2010)	TDS	Sim	Sim	Sim	Sim	Não

Tabela 3.2: Comparativo entre os trabalhos relacionados.

Sobre o critério da **fundamentação teórica** dos trabalhos relacionados, em sua maioria, os trabalhos são baseados na teoria de probabilidades (TP) para representação das incertezas em CEP. Porém, vale destacar que a teoria de probabilidades apresenta limitações para representar e lidar com informações incompletas de eventos. Por exemplo (Ma et al. 2010 [47]), ao monitorar uma pessoa que embarca em um ônibus ou entra em um prédio, uma notificação de evento é gerada a partir de um algoritmo de classificação do sexo da pessoa, que pode detectá-la como homem com uma certeza de 85%. No entanto, o valor restante não implica que a pessoa é do sexo feminino com uma certeza de 15%, pelo contrário, é desconhecida. Na teoria de probabilidade, a observação desconhecida não pode ser modelada, o que dificulta um raciocínio subsequente. Observa-se também que a TDS fornece um mecanismo conveniente para a combinação de dois ou mais resultados divergentes ou conflitantes. Seguindo o exemplo anterior, para a mesma observação, se um segundo algoritmo fornece a informação de que embarcou um homem com 50% de certeza (resposta imprecisa e divergente em relação ao evento do primeiro algoritmo de 85% de certeza), tal divergência de informações poderia ser resolvida pela regra de combinação de

Dempster, que considera as duas fontes de informação dos dois algoritmos e permite calcular um terceiro resultado factível. Assim, vale reiterar que a TDS é adequada quando há pouca informação sobre os eventos para avaliar uma probabilidade ou quando a informação é imprecisa, incompleta, divergente ou conflitante.

Redes Bayesianas (RB) são utilizadas em Wasserkrug et al. 2005 [81] e 2012 [85] para atribuir o valor de probabilidade associada a uma regra para todo evento inferido a partir dela. Além disso, apenas os eventos inferidos compõem os nós da rede. Já em Cugola et al. 2015 [17] é construída uma Rede Bayesiana inicialmente simples, de forma automatizada a partir de uma regra. Esta RB é refinada e ”enriquecida” por um especialista do domínio. Ou seja, a RB é atualizada com informações ou detalhes relevantes que podem influenciar a ocorrência dos eventos na rede e evitar deduções erradas. As probabilidades dos eventos derivados são computadas considerando as informações inseridas na rede. No entanto, vale ressaltar que uma das principais desvantagens das RBs está na complexidade da sua definição. O uso da RB requer probabilidades condicionais numéricas completas sendo especificadas entre nós. Esse requisito é impraticável em muitas aplicações em que as probabilidades condicionais estão indisponíveis. Dependendo do nível de conhecimento do problema por parte do especialista e da disponibilidade de informações sobre o domínio em análise, poderia ser difícil ou impossível a obtenção das probabilidades de toda a rede completa.

Seguindo essa limitação, em Cugola et al. 2015 [17], a geração automática da RB se dá somente na parte qualitativa da rede que são as relações conceituais e de dependências entre as variáveis de um domínio. Porém, esta é a parte da rede em que os especialistas têm afinidade e são bons em julgar, diferentemente da parte quantitativa da rede, onde os especialistas não são tão bons pela dificuldade em avaliar as probabilidades associadas. Assim, o trabalho se propõe a construção automática da parte qualitativa da RB que, infelizmente, é a parte mais factível pelo especialista. Em contrapartida a parte mais desafiadora é avaliar as probabilidades das variáveis que formam a parte quantitativa e esta última é exatamente a parte designada ao especialista no passo de enriquecimento da rede. A TDS lida com este tipo de limitação como já foi discutido na seção 2.4.3.

Em Jarraya et al. 2016 [35] o evento FSCE utiliza uma função membro da lógica fuzzy (LF) e sugere isso como diferencial no trabalho. O resultado gerado

é compreendido como uma representação do evento *fuzzy* com dupla interpretação ($FSCE_{presence} = \{[livingroom; 0.33], [office; 0.67]\}$). Esta característica é interessante e também identificada na TDS que permite múltiplas interpretações de um evento. No entanto, neste trabalho, os cálculos dos valores *fuzzy* seguem princípios da teoria de probabilidade (ex.: axioma de aditividade), tornado-o menos flexível em relação à representação da incerteza na TDS (discussão seção 2.3.2). O trabalho não lida com incerteza nas regras, embora lide com a propagação da incerteza atribuindo a cada sensor um valor de confiança fornecido por um especialista. Em seguida, os eventos com o grau de confiança são processados gerando um evento complexo FSCEP com um valor de confiança resultante. Cabe avaliar a solução FSCEP frente a outras abordagens probabilísticas e não apenas comparar com resultados de abordagens CEP clássicas, para assim validar a precisão dos resultados.

Em Rince et al. 2018 [62], baseada em Markov (CM), existe a limitação quanto à definição de padrões de eventos complexos, onde especialistas do domínio podem não ter o conhecimento adequado (formalismos lógicos) para realizar essa tarefa. Nas *engines* CEP a filosofia declarativa das regras é concebida para capturar padrões complexos de eventos sem a exigência de um conhecimento de formalismo lógico específico. Por exemplo, padrões: *((H H) – [C]) equals ((off off) – [on])* que descreve duas ocorrências (*H*) de furacão registradas, mas o alarme não é disparado (*off*). Ou ainda: *H during ((off off) – [on])*, afirmando que durante um intervalo de tempo, o alarme não foi disparado enquanto ocorria um furacão. Outra limitação dessa abordagem é que seu foco está em um tipo de incerteza em CEP, diferente das outras abordagens que focam em dois ou mais tipos de incerteza. Além disso, é apontado pelo autor o alto custo computacional de raciocínio sob incerteza com MLN. Exige um esforço computacional incompatível com as restrições impostas pelos domínios de aplicações CEP que exploram a manipulação de milhares de eventos por segundo, com centenas de regras implantadas na *engine* de processamento.

Em relação ao critério dos **tipos de incertezas** tratadas nos trabalhos relacionados, observa-se que a incerteza no conteúdo do evento é a mais abordada seguida da propagação da incerteza. Esses são exatamente os dois tipos de incertezas em CEP que pretende-se abordar nesta tese. Destaca-se que o trabalho que aborda a maior quantidade de tipos de incerteza é o de Cugola et al. 2015 [17]. No entanto, este trabalho lida com a propagação da incerteza utilizando RB combinada com

conhecimento especialista. Neste caso, a TDS se mostra mais adequada para lidar com conhecimento especialista e supera certas limitações da RB (seções 2.4.2 e 2.4.3).

Sobre o critério de **pressupostos de independência entre os eventos**, a maior parte dos trabalhos relacionados pressupõem a independência dos eventos de entrada. Ou seja, a probabilidade do primeiro evento não interfere ou tem relação com a probabilidade do segundo. O trabalho de Wasserkrug et al. 2005 [81] assume a independência de eventos, pelo menos até o nível dos eventos primitivos. Wasserkrug et al. 2012 [85] pressupõe independência dos eventos até o momento do uso da tabela de probabilidade condicional, que necessita da especificação das dependências probabilísticas entre variáveis no cenário do surto de gripe ou ataque bioterrorista. Cugola et al. 2015 [17] pressupõe a independência dos eventos até o uso da RB, especificamente na etapa de enriquecimento da rede, em que devem ser estipulados os valores de probabilidades nas relações de dependência entre os eventos. Em Rince et al. 2018 [62], os eventos são dependentes (uso da cadeia de Markov) onde a probabilidade do evento que leva o sistema para o próximo estado depende do estado atual sistema. As situações de dependência entre eventos podem afetar a complexidade do processamento. Finalmente, em Ma et al. 2010 [47] os eventos são independentes, uma característica que a TDS pressupõe para múltiplas fontes de eventos.

O critério de **filtragem de eventos irrelevantes** especifica se existem mecanismos para o tratamento ou filtragem de eventos que podem ser descartados ou desconsiderados antes do processamento. Nos trabalhos de Wasserkrug et al. 2005 [81] e 2012 [85], os modelos propõem expressões (ex.: *Sel*, *Selectability*) para filtrar somente tipos específicos de eventos no fluxo, os quais devam ser relevantes para as regras. Este mecanismo é discutido como ponto de melhora na performance da solução. Em Jarraya et al. [35] coloca-se a possibilidade de filtrar eventos relevantes sobre limiares do grau de confiança ou valores *fuzzy* derivados dos eventos do fluxo. Em Ma et al. [47] define-se um valor de significância para os eventos como ponto de filtragem dos eventos relevantes. O autor destaca que este mecanismo é essencial para o domínio da aplicação de vigilância e monitoramento, no qual 99% dos eventos são considerados triviais (ou irrelevantes) para a detecção de perigos e ameaças em tempo real.

O último critério especifica se os trabalhos fazem uso de recursos de **tecnologias CEP** que facilitam o desenvolvimento de aplicações que possam reagir em

tempo real. As linguagens CEP atuais fornecem um conjunto variado de operadores que viabilizam e flexibilizam a execução dessas tarefas de seleção, correlação e filtragem de eventos de maneira satisfatória (ex.: Esper EPL - *Event Processing Language*). Em Wasserkrug et al. 2005 [81] não se utiliza linguagem de processamento e propõe-se expressões próprias para manipulação de eventos. Em Ma et al. [47] também são propostos conceitos próprios de manipulação de eventos, como *cluster* de eventos considerando os tipos de eventos, além de *life span* usado para determinar o intervalo temporal dos eventos. Entretanto, linguagens CEP atuais já fornecem recursos como consultas contínuas, primitivas de condição e correlações entre eventos e conceitos de janelas de tempo, que viabilizam o desenvolvimento de aplicações.

Finalmente, cabe destacar algumas lacunas nas abordagens apresentadas. Inicialmente, nas abordagens probabilísticas o espaço amostral representa o conjunto de resultados possíveis. Porém, ao analisar a TDS, o recurso do quadro de discernimento apresenta uma alternativa flexível para modelar problemas que consideram a formação de subconjuntos dos resultados possíveis também uma resposta factível. A maior parte das abordagens não utiliza dados de sensores reais e não consideram informações de incerteza específicas do fabricante dos sensores, por exemplo, imprecisão do sensor, no processamento da solução. Em alguns trabalhos relacionados, os parâmetros utilizados são estimados arbitrariamente. Por exemplo, probabilidade de regras, probabilidades condicionais, e a incerteza nas medições dos sensores são taxadas em +/− 10%, +/− 25% e até sensores +/− 50% imprecisos. Ou seja, as informações de incerteza nos evento precisam de uma representação formal para tratar a incerteza nos eventos associando um nível de incerteza relacionado à fonte produtora do evento (por ex.: leituras de sensores) de forma natural, sem generalizações de precisões artificiais para todos os sensores. Isso exige uma investigação da mecânica de funcionamento dos sensores e das leituras dos fenômenos capturados por cada um deles. Outro problema é conseguir modelar a "ignorância parcial" do conhecimento especialista ou informações incompletas nos eventos. Ou seja, as abordagens exigem a probabilidade atribuída para um evento e também a sua negação, o que força a conclusão de que o conhecimento de um evento implica necessariamente no conhecimento do seu complemento. Esta obrigatoriedade é exigida pela lei de aditividade da teoria de probabilidade, que é base para boa parte dos trabalhos relacionados. Porém, este requisito é impraticável em muitas aplicações

em que as probabilidades não estão disponíveis ou não existe conhecimento suficiente para obtê-las. Algumas abordagens (ex.: Wasserkrug et al. [81] [85], Cugola et. al [17]) exigem que se tenha as informações sobre a probabilidade de todos os eventos, ou seja, a solução não suporta eventual ignorância do especialista ou falta de informação sobre a probabilidade dos eventos ocorridos. É importante lembrar que os problemas da ignorância parcial e informações incompletas são comuns em domínios reais. Assim, tais abordagens não oferecem uma alternativa para representar e raciocinar sobre informações incertas. Outra lacuna observada refere-se ao fato de que a maioria das abordagens não avalia as soluções considerando dados de sensores reais. Além disso, não utilizam um conjunto de métricas que permitiria uma avaliação de performance razoável das abordagens (por exemplo, *Accuracy, Precision, Recall, F-Measure, curva ROC e AUC*).

3.8 Síntese

Este capítulo apresentou uma ampla visão das principais pesquisas, abordagens e técnicas para realizar o tratamento de incertezas presentes no processamento de eventos. Foi realizado um mapeamento dos trabalhos relacionados a partir do estudo de *surveys* que abordam o tema. Em seguida foi apresentado um mapa geral de tópicos para estruturar o conhecimento sobre o assunto. A partir do mapa, foram estabelecidos critérios de seleção dos trabalhos relacionados. Ao final, os trabalhos foram comparados segundo critérios de comparação apresentados em uma tabela comparativa e uma discussão sobre os principais problemas e limitações das abordagens foi apresentada.

4 Abordagem *DST-CEP*

Este capítulo apresenta a abordagem *DST-CEP* (*Demspter-Shafer Theory for Complex Event Processing*) para lidar com o problema desta pesquisa. Para isso, inicialmente a hipótese desta pesquisa é retomada. Em seguida, são apresentadas as propostas de representação formal de eventos e o modelo arquitetural da abordagem *DST-CEP*. São descritas as etapas de modelagem da incerteza nos eventos e a propagação para os eventos complexos sob a perspectiva dos elementos e funções da Teoria de Dempster-Shafer. Finalmente, um cenário de IoT é apresentado com os detalhes da aplicação e abordagem *DST-CEP*.

Buscou-se nessa pesquisa tratar os problemas de incerteza no processamento eventos em aplicações de IoT baseadas em CEP. Portanto, como apresentado na Seção 1.3, a **hipótese de pesquisa** neste trabalho é:

A Teoria de Dempster-Shafer possibilita desenvolver uma abordagem para adequadamente modelar e tratar os problemas de incerteza originados de dados de sensores não confiáveis e, especificamente para aplicações de IoT baseadas em processamento de eventos, lidar com a incerteza na origem dos eventos e sua propagação para os eventos derivados.

4.1 Representação de Eventos na abordagem *DST-CEP*

Uma das contribuições deste trabalho é um mecanismo de representação das informações modeladas de incerteza, levando-as em consideração explicitamente nos eventos. Portanto, a modelagem de incerteza nos eventos a partir dos elementos da TDS (por exemplo, função de massa e fator de desconto) necessita de uma representação formal dos eventos para a abordagem *DST-CEP*. Inicialmente as notificações dos sensores são eventos de evidência que possuem a seguinte representação:

$$e_i = (idSource, Ts, \{(epl_i, df_i)\}) \tag{4.1}$$

- $idSource$: identificação da fonte produtora do evento;

- Ts: *timestamp* do evento;
- epl_i: *evidence payload* ou o dado da evidência (ex.: localização, área, temperatura, nível de CO, etc.);
- df_i: fator de desconto relacionado à precisão do sensor;

As notificações dos eventos de evidência de entrada na abordagem *DST - CEP* são evidências que permitem extrair a conjectura de hipóteses de um determinado domínio, onde cada hipótese tem um valor de massa associado. Dessa forma, os eventos derivados nessa etapa são representados por eventos de conjecturas de hipóteses (*hc*), como segue:

$$hc_i = (idSource, Ts, (epl_i, df_i), \{(H_i, m_i), ..., (H_n, m_n)\}) \tag{4.2}$$

- H_i: hipótese de um domínio;
- m_i: valor de massa para a hipótese.

Portanto, os eventos *hc* na abordagem *DST-CEP* são carregados com informações de incerteza relacionadas à fonte produtora do evento (ex.: leituras de sensores), as hipóteses de um domínio e os respectivos valores de massa associados. Tais informações de incerteza precisam ser identificadas, modeladas e calculadas, como será apresentado nas seções seguintes.

4.2 Modelo Arquitetural

O modelo arquitetural *DST-CEP* foi projetado e implementado utilizando blocos de construção ou DSTBB (do inglês *DST-CEP Buiding Block*). O modelo inclui componentes da TDS em um contexto de processamento de eventos originados a partir de uma coleção de sensores. A figura 4.1 apresenta uma instância de um bloco de construção *DST-CEP*.

Inicialmente, um DST-CEP Building Block (DSTBB) é dividido em níveis de processamento. Como ilustra a figura 4.1, o Nível de Sensores (*Sensor Level*) possui uma coleção de sensores $S_1, S_2, ..., S_n$ que enviam eventos primitivos $e_1, e_2, ..., e_n$ (ou leituras

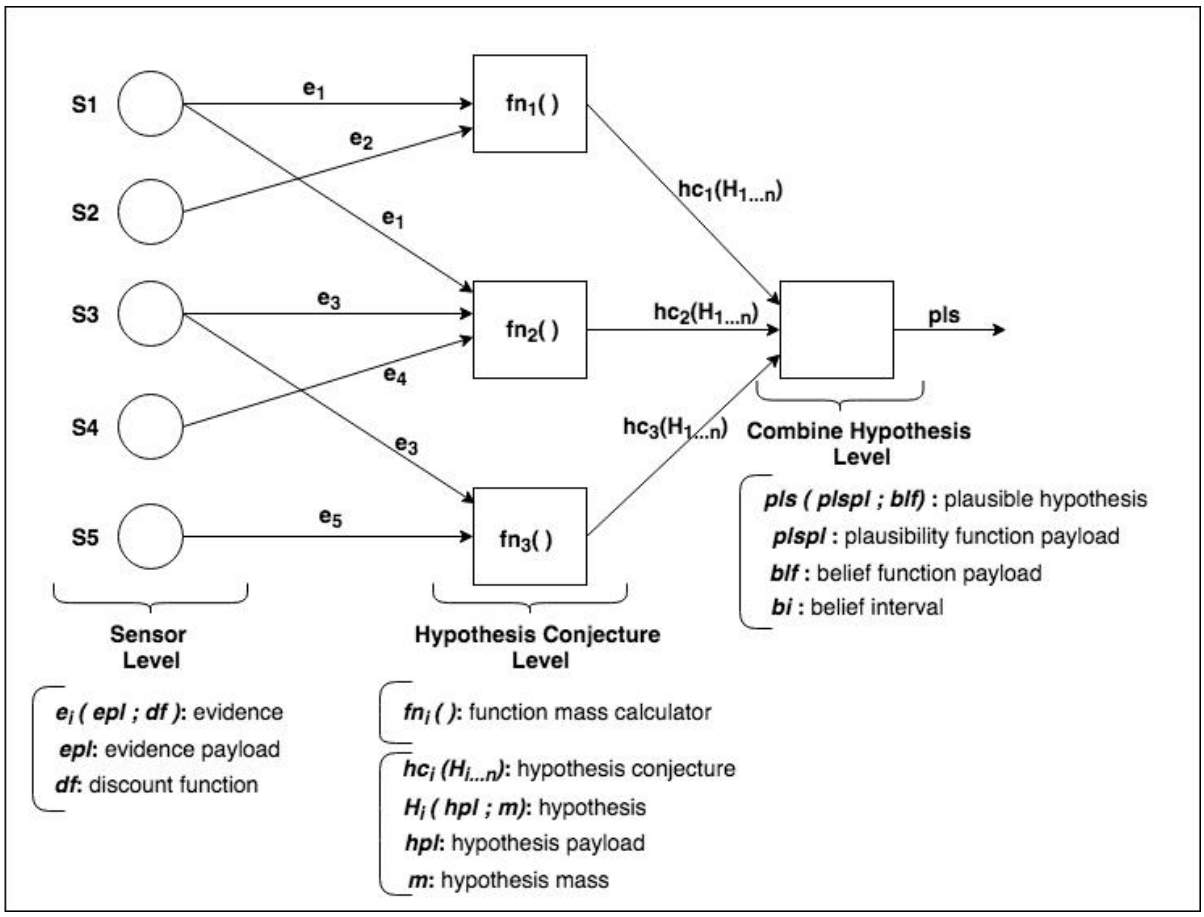


Figura 4.1: *DST-CEP* building block (**DSTBB**).

de sensores) que são evidências de entrada em *DST-CEP*. Cada evento $e_i(epl, df)$ é composto pelo dado da evidência denominado epl (*evidence payload*), e o fator de desconto df (*discount factor*) calculado a partir da precisão do sensor.

O Nível de Conjectura de Hipóteses (*Hypothesis Conjecture Level*) possui um conjunto de regras CEP que realizam o processamento dos eventos primitivos. Os resultados são eventos derivados que representam conjecturas de hipóteses (h_c) com o valor de massa associado para cada hipótese. A geração das hipóteses pode ocorrer a partir das evidências de um ou mais sensores. O valor de massa gerado para cada hipótese (H_1,...,H_n) do conjunto de hipóteses é calculado através da função de massa $fn_i()$ (*mass function*), cuja formulação depende do domínio da aplicação. Observa-se na Figura 4.1 a possibilidade de diferentes funções de massa ($fn_1(), fn_2(), fn_3()$), o que se traduz em um conjunto de regras específicas definidas por um ou mais especialistas.

No Nível de Combinação de Hipóteses (*Combine Hypothesis Level*) a partir de um conjunto de hipóteses derivadas é utilizada a Regra de Combinação de Dempster $dr()$ para calcular a hipótese mais plausível, denominada pls (*plausible*

hypothesis). Dados adicionais podem compor os resultados, como a função de crença (*belief function*), função de plausabilidade (*plausibility function*) e intervalo de crença (*belief interval*). A composição de um conjunto de DSTBBs representa uma rede de processamento de eventos EPN (*Event Processing Network*) ou uma EPN Bulding Blocks, como ilustra a Figura 4.2.

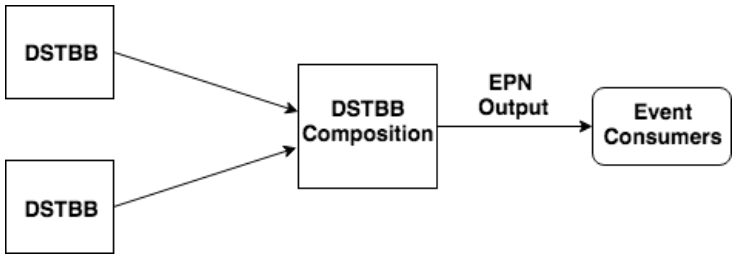


Figura 4.2: EPN Building Block

4.2.1 Modelagem de Incerteza nos Eventos

A Incerteza nos eventos decorre da observação incorreta dos fenômenos sob análise, causada pela precisão limitada dos sensores, que introduz incerteza nos dados observados. Dentre os motivos, pode-se citar:

- Pode depender da **aleatoriedade de fabricação** dos sensores, **diferentes níveis de precisão** definidos por diversos fabricantes, **imprecisões na técnica de medição** ou ruído nas fontes.

Funções de Massa

Na abordagem *DST-CEP*, cada hipótese no quadro de discernimento deve receber um valor de massa representado por uma função de massa. Para ilustrar exemplos de funções de massa, considere um **Sistema Multisensor de Identificação de Alvos**, ou seja, uma aplicação de visualização de vigilância aérea para dados de radares sensores, pronta para delimitar faixas de segurança e alertas. Os dados utilizados nesta aplicação são encontrados em [14]. O objetivo é evitar o problema de abatimento de aviões errados ou considerados amigáveis, além de evitar ataques inimigos de aviões bombardeiros, caças e/ou mísseis.

As funções de massa não são independentes de domínio, ou seja, cada função leva em consideração parâmetros específicos e mecânica de funcionamento do

sensor que precisa ser estudado. No exemplo, RWR, TVSU e o IRST são sensores alocados para a tarefa de identificação de alvos. O **RWR** (*Radar Warning Receiver*) detecta as emissões de rádio dos sistemas de radar e analisa as características da radiação emitida para determinar a identidade do emissor, com o objetivo de emitir um alerta quando um sinal de radar que pode ser uma ameaça for detectado. O **TVSU** (*Television Sighting Unit*) identifica os alvos com base na forma do alvo observado. Ele combina um conjunto de imagens de um alvo antes da identificação positiva ser feita. O número de imagens necessárias pode variar enormemente para alguns alvos e ângulos. O **IRST** (*InfraRed Search and Track*) pode diferenciar entre alvos com base no comportamento do alvo observado. Ex.: Ele está voando em um "corredor seguro"? Está voando em uma rota comercial conhecida? Seu comportamento de tática/vôo parece hostil?

Para evitar um erro drástico de operação (ex.: abatimento do alvo sem a rigorosa precisão de identificação), os dados coletados dos diversos sensores são combinados para obtenção mais precisa da natureza do alvo. Infelizmente, todos os sensores apresentados também são suscetíveis a falhas e possuem individualmente diferentes confiabilidades, como ilustra a Figura 4.3 [14]. Por exemplo, o alcance do sensor RWR permite detectar um alvo a partir de uma distância de aproximação de 70 nmi (milhas náuticas). Entretanto, estudos estatísticos e testes do sensor RWR ilustram que sua performance, a essa distância de aproximação, apresenta uma resposta de identificação com confiança extremamente baixa. À medida que o alvo se aproxima do sensor RWR, o valor de confiança das respostas do sensor RWR aumenta, como ilustra o gráfico de performance RWR (Fig. 4.3).

Vale destacar que os outros sensores IRST e TVSU possuem curvas de performance diferentes em função dos dados coletados por cada um, além dos mecanismos específicos de detecção empregados. Por exemplo, o sensor TVSU (vide Figura 4.3) baseado na detecção de alvos a partir de imagens possui um alcance inferior a 50 nmi. Entretanto, o valor de confiança dos resultados aumenta vertiginosamente à medida que o alvo se aproxima dentro da faixa de alcance do sensor TVSU. Observa-se também que todos os sensores **não** são 100% confiáveis, mesmo sob a distância mínima do alvo. Além disso, os níveis de confiança máxima de cada sensor possuem diferentes patamares. Dessa forma, as hipóteses (H_i) de identificação dos alvos resultantes de cada sensor devem ter associado à sua resposta um valor de massa (m) baseado

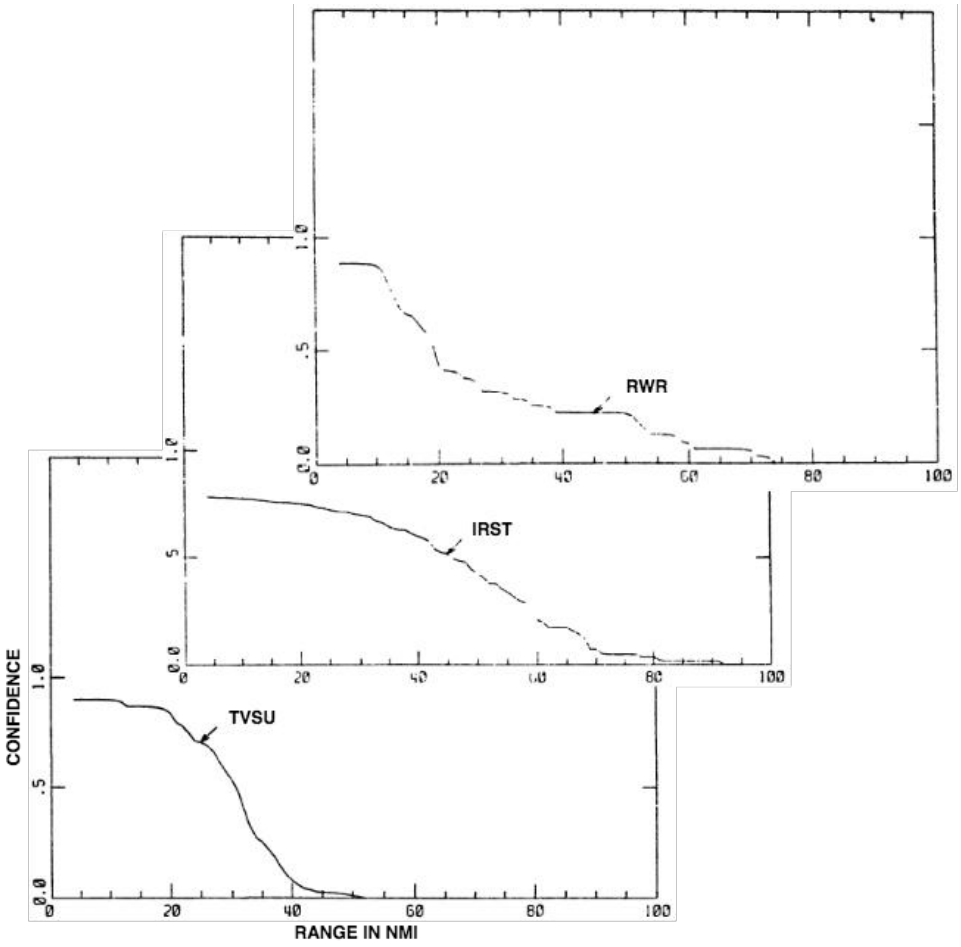


Figura 4.3: Confiança dos Sensores [14]

no nível de confiança (performance) de cada sensor (fator de desconto d_f). Dado o problema, as hipóteses de identificação que compõem o quadro de discernimento são $\Theta = \{Aircraft, Bomber, Fighter, Missile\}$.

Podemos citar dois exemplos de declarações de valores de massa para as hipóteses. À medida que o alvo se aproxima, o Sensor RWR (*Radar Warning Receiver*) coleta informações específicas sobre o alvo, computa os dados e declara possíveis hipóteses de identificação. Entretanto, a declaração do Sensor RWR pode indicar duas possibilidades de hipóteses (ex: Aircraft ou Bomber) devido à distância de aproximação, fatores do ambiente (clima tempo) e/ou mecanismo de identificação do sensor. Nesse sentido, DST-CEP modela esse tipo de problema permitindo a declaração de massa para a hipótese de identificação do alvo (Aircraft ou Bomber) declarada pelo Sensor RWR como $m_{RWR}(Aircraft, Bomber) = 0.6$, cujo valor de massa procede dos dados de performance do sensor. O valor de $m_{RWR}(\Theta) = 0.4$ é a incerteza associada à declaração do Sensor RWR neste instante da leitura dos dados. Da mesma

forma, o Sensor IRST (*InfraRed Search and Track*) coleta informações específicas sobre o alvo, computa os dados e declara a seguinte hipótese de identificação $m_{IRST}(Bomber, Fighter) = 0.7$ e $m_{IRST}(\Theta) = 0.3$. Diferente da declaração de hipótese do sensor anterior RWR, a identificação do Sensor IRST declarou as hipóteses *Bomber* ou *Fighter* considerando assim ser um alvo inimigo, com valor de massa 0.7 associado à hipótese. O valor de $m_{IRST}(\Theta) = 0.3$ representa a incerteza do Sensor IRST neste instante da leitura dos dados.

Fator de Desconto

Eventualmente, na abordagem *DST-CEP* os eventos poderiam ter associados um **fator de desconto** (df) para expressar outras incertezas que afetam a confiabilidade das fontes de informação. Ou seja, o fator de desconto poderia considerar fatores que vão além da precisão do sensor, por exemplo, a resolução do sensor, campo de visão do sensor, ângulo de funcionamento, sensibilidade do sensor, distância do fenômeno observado, medição em condições específicas do ambiente, ou outros fatores definidos pelo fabricante que podem afetar ou degradar de alguma forma a confiabilidade da informação fornecida. Alternativamente, um especialista do domínio poderia fornecer o fator de desconto e integrá-lo às notificações dos eventos antes do processamento.

Nesse viés, a confiabilidade da fonte do evento pode ser expressa de forma inversa ao fator de desconto. Quanto maior a confiabilidade r (do inglês, *realiability*), tem-se um menor fator de desconto (df). Desta forma, tem-se $r = 1 - df$. Assim, o valor de massa atribuído a determinada hipótese do ambiente é dado com o fator desconto, como segue:

$$m^{df}(H) = \begin{cases} (1 - df) \cdot m(H), H \subset \Theta \\ df + (1 - df) \cdot m(\Theta), H = \Theta \end{cases} \tag{4.3}$$

Onde $0 \leq df \leq 1$, sendo interpretado da seguinte forma:

- $(df = 0)$: a fonte é absolutamente confiável;
- $(0 < df < 1)$: a fonte é confiável com um fator de desconto df;

- $(df = 1)$: a fonte é completamente não confiável.

Logo, $m^{df}(H)$ é o valor de massa com o desconto (df), cujo resultado é atribuído a h, que representa uma hipótese em um dado domínio.

4.2.2 Modelagem da Propagação de Incerteza

Na seção anterior observou-se a modelagem de incerteza associada às notificações dos eventos primitivos. A propagação da incerteza dos eventos primitivos para a inferência de eventos derivados (eventos complexos) caracteriza o problema da **Propagação da Incerteza**. Tal problema causa a geração e propagação de resultados não confiáveis para uma rede de processamento de eventos (EPN), partindo do princípio de que em uma EPN são detectados eventos complexos a partir da composição de eventos primitivos ou até outros eventos complexos de entrada. Nesta seção, é apresentada a forma como a abordagem *DST-CEP* modela a propagação da incerteza usando elementos da Teoria de Dempster-Shafer, como o quadro de discernimento e a regra de combinação de Dempster. Além disso, são apresentados métodos para representar as relações de incerteza usando grafos e semi-grafos[1] [43,44, 88].

Na Figura 4.4, o quadro de discernimento $\Theta = \{h_1, h_2, h_3\}$ e suas relações com duas fontes distintas de evidências (e_1 e e_2) podem ser representados através de grafos, consistindo em nós conectados por arestas direcionadas. Usando grafos é possível acrescentar, de maneira simples, as declarações das relações de incerteza diretamente entre o espaço evidencial $E = \{e_j, ..., e_m\}$ e o espaço de hipóteses $H = \{h_i, ..., h_n\}$. Os números nas arestas representam os valores de massa para as hipóteses. Assume-se valores de massa apenas para exemplificação numérica e para fins de esclarecimento nesta seção.

Às vezes é difícil descrever claramente muitas relações de incerteza em situações complexas usando representação em grafo. Neste caso, o semi-grafo pode ser usado para representar regras a partir das relações na Figura 4.4, como ilustra a Figura 4.5.

[1]Semi-grafos significa que tanto grafos como declarações das relações de incerteza são utilizados em conjunto na representação.

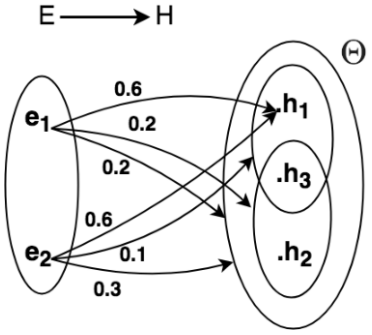


Figura 4.4: Grafo para representar declarações das relações de incerteza.

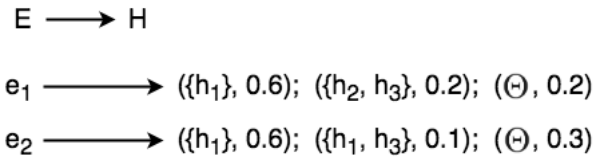


Figura 4.5: Semi-grafo para representar regras simples.

Dado o exemplo na Figura 4.4, podemos definir um conjunto de regras a partir do quadro de discernimento:

- IF e_1 THEN $\{h_1\}$ with 0.6, $\{h_2, h_3\}$ with 0.2
  Θ with 0.2;

- IF e_2 THEN $\{h_1\}$ with 0.6, $\{h_1, h_3\}$ with 0.1
  Θ with 0.3;

  where $E = \{e_1, e_2\}$ and $H = \{h_1, h_2, h_3\}$

Combinação de Hipóteses

A partir das conjecturas de hipóteses coletadas de várias fontes de informação, podemos propagar as incertezas inerentes às hipóteses de modo a combiná-las. Para realizar tal operação, como ilustra a Figura 4.6, a partir de um conjunto de conjecturas de hipóteses de diversas fontes e seus valores de massa relacionados, a regra de combinação de Dempster $dr()$ é utilizada através das Equações 4.4, 4.5 e 4.6, para obter a hipótese combinada que representa um consenso das evidências originais e possivelmente conflitantes. Além disso, a regra de combinação de Dempster permite obter e identificar a hipótese teoricamente mais plausível (*pls*).

DST-CEP deve permitir combinar diferentes fontes de evidências através da regra de combinação. No início do exemplo, a partir das evidências e_1 e e_2, tem-se as massas distribuídas para as hipóteses, como seguem:

$$m_1(h_1) = 0.6$$
$$m_1(h_2, h_3) = 0.2$$
$$m_1(\Theta) = 0.2$$

$$m_2(h_1, h_3) = 0.1$$
$$m_2(h_1) = 0.6$$
$$m_2(\Theta) = 0.3$$

Dadas as duas atribuições de massas distintas m_1 e m_2, a regra de combinação de Dempster realiza a **soma ortogonal** das atribuições para produzir uma nova massa, que representa um consenso das evidências originais e possivelmente conflitantes, neste caso indicada pela notação $m_1 \oplus m_2$:

$$m_1 \oplus m_2(H_i) = \sum_{X \cap Y = H_i} m_1(X) \cdot m_2(Y) \tag{4.4}$$

Formalmente, a soma ortogonal ($\oplus$) é definida pela soma dos produtos das massas de todos os elementos cuja interseção $X \cap Y = H_i$.

A Tabela 4.1 ilustra a soma ortogonal das massas m_1 e m_2. Inicialmente, é realizado o produto de todas as massas combinadas. Em cada quadro da tabela é mantido somente o elemento de interseção. Por exemplo, dados $m_1(h_1)$ e $m_2(h_1, h_3)$, matém-se o elemento de interseção (h_1) acompanhado do resultado do produto das massas correspondentes, $(h_1) = 0.06$.

A única dificuldade com este esquema é que ele pode atribuir massas para alguns quadros da tabela cujo conjunto vazio, por exemplo os elementos $m_1(h_2, h_3)$ e $m_2(h_1)$, têm interseção nula ($\emptyset$). Porém, a massa atribuída ao conjunto vazio deve ser igual a zero ($m(\emptyset) = 0$) por definição. Para lidar com esta situação, é realizada a normalização das massas atribuídas a todos os outros conjuntos da seguinte forma:

$$m_1 \oplus m_2(H_i) = \frac{1}{k} \cdot \sum_{X \cap Y = H_i} m_1(X) \cdot m_2(Y) \tag{4.5}$$

	$m_2(\Theta) = 0.3$	$m_2(h_1, h_3) = 0.1$	$m_2(h_1) = 0.6$
$m_1(\Theta) = 0.2$	$(\Theta) = 0.06$	$(h_1, h_3) = 0.02$	$(h_1) = 0.12$
$m_1(h_2, h_3) = 0.2$	$(h_2, h_3) = 0.06$	$(h_3) = 0.02$	Conj. Vazio 0.12
$m_1(h_1) = 0.6$	$(h_1) = 0.18$	$(h_1) = 0.06$	$(h_1) = 0.36$

Tabela 4.1: Exemplo de Soma Ortogonal das Massas

A nova massa normalizada atribuída a (H_i) é, assim, a soma ortogonal das massas, dividida pelo **fator de normalização *k***.

O fator de normalização ***k*** é o resultado da subtração de **um** menos a **soma das massas atribuídas aos conjuntos vazios** ($X \cap Y = \emptyset$) após a combinação, como segue:

$$k = 1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y) \tag{4.6}$$

Considerando no exemplo um único quadro com interseção nula (vide Tabela 4.1), o fator de normalização ***k*** resultante é:

$$k = 1 - 0.12 = \mathbf{0.88} \tag{4.7}$$

Considerando a equação (4.5) e o valor ***k*** acima, as **novas massas normalizadas** (m_3) são:

$$\begin{aligned}
m_3(\Theta) &= 0.06/0.88 = \mathbf{0.0682} \\
m_3(h_2, h_3) &= 0.06/0.88 = \mathbf{0.0682} \\
m_3(h_1, h_3) &= 0.02/0.88 = \mathbf{0.0227} \\
m_3(h_1, h_2) &= 0.0 \text{ (sem interseção na soma ortogonal)} \\
m_3(h_3) &= 0.02/0.88 = \mathbf{0.0227} \\
m_3(h_2) &= 0.0 \text{ (sem interseção na soma ortogonal)} \\
m_3(h_1) &= (0.18 + 0.06 + 0.36 + 0.12)/0.88 = \mathbf{0.8182}
\end{aligned}$$

A combinação das hipóteses geradas por múltiplas fontes (ex.: detectores) ocorre na saída. Como ilustra a Figura 4.6, dentre os resultados da hipótese combinada (h_c) é possível identificar a hipótese mais plausível (*pls*).

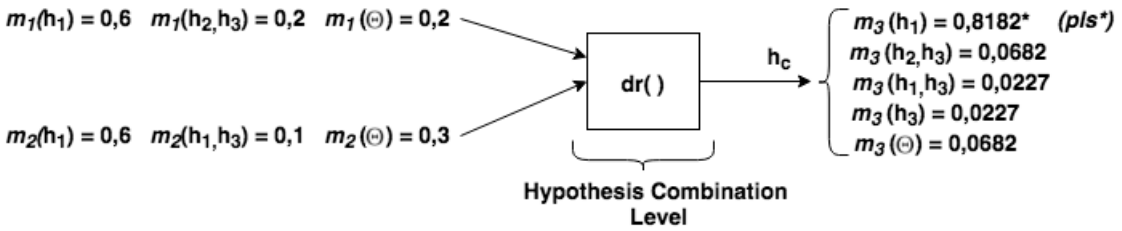


Figura 4.6: Combinação de Hipóteses (recorte da Figure 4.1).

Estes resultados mostram que o maior valor de massa apresentado é $m_3(h_1) = 0.8182$, ou seja, dentre as evidências e as hipóteses analisadas, a hipótese mais plausível é h_1, portanto a mais assertiva a ser considerada.

Retomando o exemplo do sistema de detecção de alvos e as massas atribuídas designadas pelos sensores RWR e IRST, temos respectivamente:

$$m_{RWR}(Aircraft, Bomber) = 0.6$$
$$m_{RWR}(\Theta) = 0.4$$

$$m_{IRST}(Bomber, Fighter) = 0.7$$
$$m_{IRST}(\Theta) = 0.3$$

O Sensor RWR declara ser um *Aircraft* ou um *Bomber* configurando implicitamente a incerteza do sensor sobre a natureza do alvo, se amigo ou inimigo. O Sensor IRST declara ser um *Bomber* ou um *Fighter*, ou seja, o sensor aponta a natureza inimiga do alvo, embora o tipo do inimigo não esteja claro neste momento. Nessa situação de hipóteses conflitantes dos Sensores RWR e IRST, é aplicada a regra de combinação e o resultado é apresentado pelas massas das hipóteses combinadas (m_{hc}), como segue:

$$\begin{aligned} m_{hc}(Bomber) &= 0.42 \\ m_{hc}(Bomber, Fighter) &= 0.28 \\ m_{hc}(Aircraft, Bomber) &= 0.18 \\ m_{hc}(\Theta) &= 0.12 \end{aligned}$$

Ao analisar os resultados, verifica-se que a hipótese mais plausível infere que o alvo que se aproxima é um ***Bomber*** considerando os dados lidos a partir de múltiplos sensores.

A abordagem *DST-CEP* permite o processo de análise de incertezas presentes no ambiente de IoT, considerando o processamento de dados capturados de sensores não confiáveis denominados de evidências. Com a contribuição do uso de elementos da Teoria de Dempster-Shafer para a modelagem e processamento de incertezas em fluxos de eventos, permite-se que a abordagem *DST-CEP* possa propagar a incerteza de eventos primitivos (espaço evidencial) para os eventos derivados (espaço de hipóteses). Além disso, permite-se a combinação de conjecturas de hipóteses baseadas em evidências de múltiplas fontes, para finalmente identificar a hipótese mais assertiva dentro de um domínio, através dos resultados DST-CEP.

4.3 Comparação com os Trabalhos Relacionados

No Capítulo 3 de trabalhos relacionados foi feita uma discussão e análise comparativa entre os trabalhos selecionados. Entretanto, nesta Seção 4.3, a abordagem *DST-CEP* é incluída no comparativo de trabalhos relacionados com foco nos critérios e contribuições atendidas pela abordagem. A Tabela 4.2 ilustra os critérios de comparação.

Trabalhos	Fundamentação Teórica	Incerteza no evento (Atrib./Ocorr.)	Incerteza nas regras ou propagação	Pressupostos de Independência	Filtragem de Eventos Relevantes	Tec. Proc. de Eventos Complexos
MRwURinECS (Wasserkrug2005)	TP e RB	Sim	Sim	Sim	Sim	Não
EPofUEinRBS (Wasserkrug2012)	TP e RB	Sim	Sim	Sim	Sim	Não
IUinCEP-MIV (Cugola2015)	TP e RB	Sim	Sim	Sim	Não	Sim
EPunderU (Artikis2012)	TP	Sim	Sim	Sim	Não	Sim
ManUnc (Moreno2019)	TP	Sim	Sim	Sim	Não	Sim
FuzzySCEP (Jarraya2016)	LF	Sim	Não	Sim	Não	Sim
CEPUwMarkov (Rince2018)	CM	Sim	Não	Não	Não	Sim
EMRwUIforDSN (Ma2010)	TDS	Sim	Sim	Sim	Sim	Não
DST-CEP (Bezerra2021)	TDS	Sim	Sim	Sim	Sim	Sim

Tabela 4.2: Comparativo entre a abordagem *DST-CEP* e os trabalhos relacionados.

A **fundamentação teórica** especifica a base teórica utilizada na abordagem *DST-CEP*, neste caso é a Teoria de Dempster-Shafer. A maioria dos trabalhos são baseados na teoria de probabilidades (TP) para representação das incertezas em CEP. Porém, vale lembrar que a teoria de probabilidades apresenta limitações para representar e lidar com informações incompletas de eventos. Já *DST-CEP* fornece mecanismos convenientes para a combinação de dois ou mais resultados divergentes ou conflitantes, ex.: regra de combinação. Em relação ao critério dos **tipos de incertezas** tratadas nos trabalhos relacionados, observa-se que a incerteza no conteúdo do evento é a mais abordada seguida da propagação da incerteza. Esses são exatamente os dois tipos de incertezas abordadas em *DST-CEP*. Sobre o critério de **pressupostos de independência entre os eventos**, a maior parte dos trabalhos relacionados pressupõem a independência dos eventos de entrada. Ou seja, a probabilidade do primeiro evento não interfere ou tem relação com a probabilidade do segundo. Em tal característica a abordagem *DST-CEP* já pressupõe para múltiplas fontes de eventos independentes. Os critérios de **filtragem de eventos irrelevantes** e o uso de recursos de **tecnologias CEP** são atendidos pela abordagem *DST-CEP* e são vistos como ponto de melhora na performance da solução. Isso por que a linguagem Esper EPL está incorporada a abordagem *DST-CEP* e tal linguagem CEP fornece um conjunto de operadores que viabiliza e flexibiliza a execução de tarefas de seleção, correlação e filtragem de eventos de maneira satisfatória. Por fim, a abordagem *DST-CEP* faz uso de recursos de tecnologias CEP que facilitam o desenvolvimento de aplicações que possam reagir em tempo real.

4.4 Síntese

Este capítulo apresentou a abordagem *DST-CEP*, inicialmente contribuindo com uma representação formal dos eventos carregados com informações de incerteza e posteriormente processados na solução. Em seguida, foi apresentado o modelo arquitetural da abordagem *DST-CEP* onde as informações de incerteza são modeladas nos eventos e são propagadas para eventos derivados sob a perspectiva dos elementos e funções da Teoria de Dempster-Shafer. Para ilustrar o uso da abordagem *DST-CEP*, um cenário do sistema de detecção de alvos em tempo real baseado em múltiplos sensores foi apresentado.

5 Implementação do Framework *DST-CEP*

Este capítulo apresenta a implementação do framework *DST-CEP* considerando o modelo arquitetural *DST-CEP* definido no Capítulo 4. Inicialmente, são apresentados a modelagem e os detalhes de implementação do framework. Em seguida, é apresentado um estudo de caso de uma aplicação de IoT de processamento de eventos na presença de incerteza. Finalmente, a aplicação do estudo de caso é desenvolvida com o uso do framework e os níveis de processamento são apresentados detalhando cada etapa de implementação do framework *DST-CEP*.

5.1 Framework DST-CEP

Baseado na Teoria de Dempster-Shafer, todo o formalismo apresentado no modelo arquitetural *DST-CEP* (Figura 4.1, Capítulo 4) foi implementado através do framework *DST-CEP* com algoritmos e funções da TDS usando tecnologias CEP (Esper[1]) e Java. O framework *DST-CEP* utiliza CEP para armazenar consultas contínuas executadas enquanto os dados fluem através das consultas. Cada consulta contínua implementa primitivas CEP em tempo real para reagir, processar e derivar outros eventos de mais alto nível. As declarações de consultas são escritas em EPL[2] [24], o que permite expressar conceitos de janelas de tempo, condições e correlações entre eventos, que facilitam o desenvolvimento de aplicações que possam reagir a situações complexas em tempo real.

O framework *DST-CEP* captura funcionalidades comuns em aplicações de IoT que lidam com o problema de incerteza em processamento de eventos. O framework *DST-CEP* provê uma solução, através de um conjunto de classes e interfaces, para permitir a construção de aplicações, especificando apenas as particularidades de cada aplicação de processamento de eventos sob condições de incerteza. Ou seja, ao utilizar o framework, o trabalho do desenvolvedor consiste em prover apenas as classes específicas do domínio da sua aplicação. Desse modo,

[1]Esper é um Event Stream Processing (ESP) e Engine CEP de correlação de eventos.

[2]A linguagem EPL é fornecida pelo Esper e implementada na solução *DST-CEP*.

o framework *DST-CEP* apresenta um conjunto de classes extensíveis e objetos que colaboram para cumprir funcionalidades específicas da aplicação e da Teoria de Dempster-Shafer, como por exemplo, a regra de combinação de Demspter.

O Diagrama de Classes do Framework *DST-CEP* é apresentado na Figura 5.1. O conjunto de classes do framework *DST-CEP* pode ser dividido em três categorias: as classes que configuram a rede EPN, as classes que gerenciam a produção dos eventos e as classes que processam os eventos (EPA's) efetuando cálculos das funções da Teoria de Dempster-Shafer.

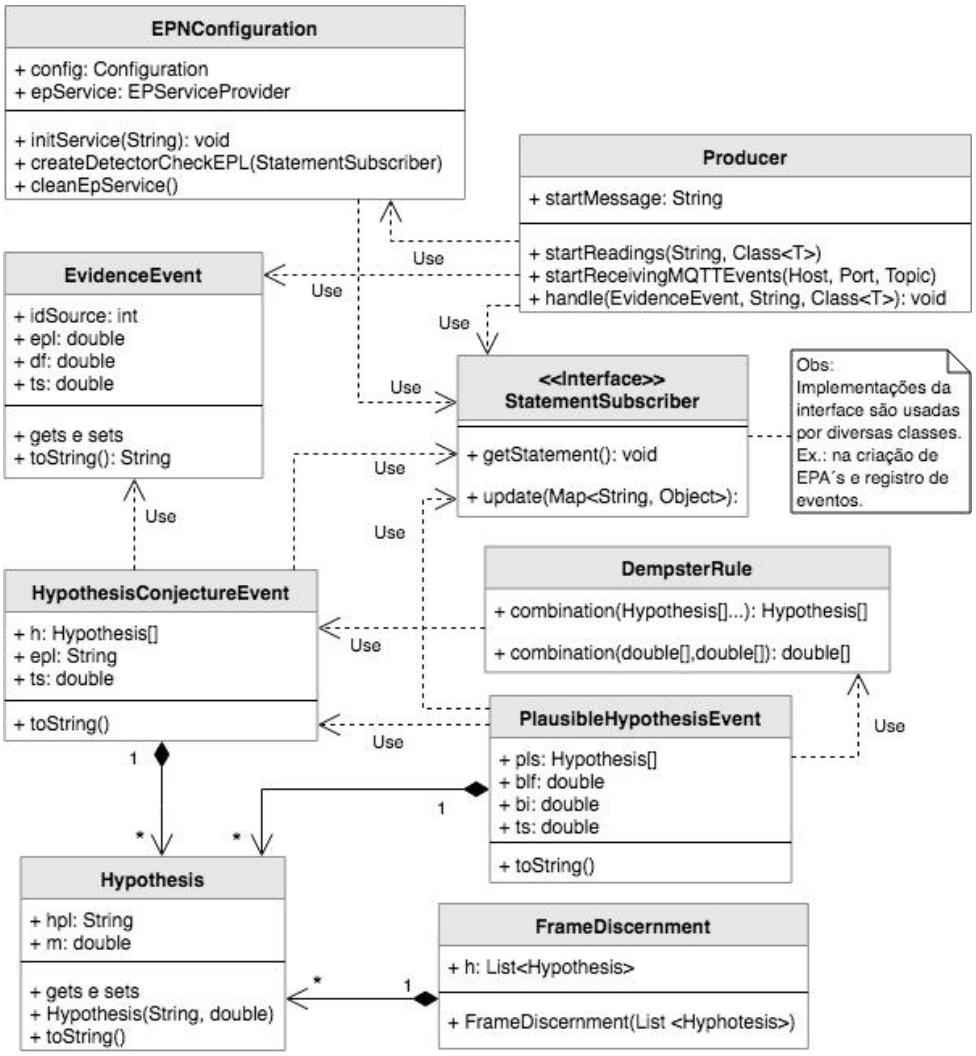


Figura 5.1: Diagrama de Classes do Framework DST-CEP.

Configuração da rede EPN

Como ilustra a Figura 5.1, a classe do framework *DST-CEP* responsável pela configuração da rede EPN é a *EPNConfiguration*. Nesta classe o método *initService()*

inicia as configurações e serviços da *engine Esper*. O método *createDetectorCheckEPL()* cria, checa (valida) e implanta as consultas contínuas (expressões EPL) da aplicação, tornando os EPA's ativos (através de subscritores) na rede EPN.

Gerenciamento e Produção dos Eventos

A classe do framework *DST-CEP* responsável pelo gerenciamento e produção dos eventos é a classe *Producer*. Nesta classe, o método genérico *handle()* permite receber eventos da aplicação que estendem a classe *EvidenceEvent* (ou eventos da aplicação que são os eventos de evidência no framework). Quando se deseja validar aplicações, uma possível fonte de dados podem ser arquivos de *dataset* contendo fluxos de eventos coletados a partir de sensores reais. Nessa situação, o método *startReadings()* é um método genérico que realiza a leitura de arquivos de fluxos de eventos registrados por sensores. Eventualmente, outra possibilidade de validação das aplicações é através da coleta de fluxo de eventos diretamente dos sensores, para isso utilizando padrões de mensagens de IoT, como por exemplo o protocolo MQTT[3](do inglês *Message Queue Telemetry Transport*). Neste caso, na classe *Producer* o método denominado *startReceivingMQTTEvents()* realiza o recebimento de eventos diretamente de um tópico do sensor de interesse (requisição através do *Broker* responsável por rotear mensagens até o destinatário). A classe *Producer* possivelmente poderia ser estendida para outros serviços de barramento de mensagens ou plataformas de fluxo de eventos distribuídos de código aberto, como por exemplo, a interface Apache Kafka[4], assim facilitando a comunicação com fluxos de eventos de sensores de variadas aplicações de IoT.

Processamento dos Eventos

O framework *DST-CEP* apresenta classes projetadas a partir do modelo arquitetural DST-CEP (Figura 4.1) e rede EPN. Inicialmente qualquer aplicação que vai utilizar o framework *DST-CEP* deverá definir um quadro de discernimento composto de um conjunto de hipóteses específicas do domínio da aplicação. Para isso, as classes *FrameDiscernment* e *Hypothesis* do framework (Figura 5.1) atendem essa premissa,

[3] `https://mqtt.org/`

[4] `https://kafka.apache.org/`

assim permitindo o registro do quadro de discernimento e a respectiva composição de hipóteses relacionadas com um domínio. Uma hipótese (classe *Hypothesis*) possui os seguintes atributos: nome da hipótese (hpl) e seu respectivo valor de massa (m).

Seguindo a organização em níveis do modelo arquitetural, no nível de sensores uma coleção de sensores envia eventos (leituras de sensores) que são eventos de evidência de entrada para a rede EPN. No framework *DST-CEP*, a classe *EvidenceEvent* representa este tipo de evento de entrada. Cada evento de evidência é composto pelo $idSource$ identificador da fonte produtora do evento (sensor), dado da evidência epl (*evidence payload*), o fator de desconto df a partir da precisão do sensor e o *timestamp* do evento ts.

Em seguida, no nível de conjectura de hipóteses do modelo arquitetural *DST-CEP* (Figura 4.1), os eventos de evidência são entradas para os agentes de processamento de eventos (EPA's) intermediários da EPN. Os EPA's recebem e processam as evidências. Na saída, tem-se as conjecturas de hipóteses, representadas pela classe *HypothesisConjectureEvent* do framework *DST-CEP*. Este evento de conjectura de hipótese é constituído de dados da leitura do sensor (epl,ts) e dados das hipóteses. Neste ponto, os dados das hipóteses, em específico o valor de massa m para cada hipótese, são calculados a partir de funções de massa, como ilustra o modelo arquitetural (Figura 4.1). As funções de massa são utilizadas pelos EPA's intermediários na rede EPN. Além disso, tais funções de massa são específicas do domínio da aplicação e são chamadas em consultas EPL implementadas nos respectivos EPA's. Para realizar tal tarefa, o framework *DST-CEP* (Figura 5.1) fornece a interface *StatementSubscriber* onde cada classe EPA implementa e declara uma consulta EPL através do método *getStatement()*. A interface *StatementSubscriber* fornece o método *update* que recebe eventos detectados nas consultas em tempo de execução e permite apresentar os resultados.

No nível de combinação de hipóteses do modelo arquitetural *DST-CEP* (Figura 4.1), os eventos de conjectura de hipóteses são combinados no processamento da hipótese plausível (pls) que ocorre na saída. Para alcançar tal resultado, no framework *DST-CEP* (Figura 5.1), os eventos de conjectura de hipóteses (classe *HypothesisConjectureEvent*) são entradas na regra de combinação de Dempster. A classe *DempsterRule* realiza a regra de combinação Dempster através da chamada do método *combination()* que utiliza as Equações 4.5 e 4.6. O resultado do

cálculo da combinação é o evento de conjectura da hipótese plausível, representado pela classe *PlausibleHypothesisEvent*. Da perspectiva da rede EPN, o último EPA da rede específico da regra de combinação de Dempster, consome os eventos do tipo *HypothesisConjectureEvent* que são as saídas do processamento dos EPA's intermediários na EPN. A classe *DempsterRule* pertence ao framework e é utilizada através do método *combination()*, chamado partir da consulta EPL implementada no EPA de combinação. O resultado do processamento deste EPA de combinação é o evento *PlausibleHypothesisEvent*.

A comunicação entre os elementos do modelo arquitetural *DST-CEP* apresentado no Capítulo 4 (Figura 4.1) deve ser refletida no framework *DST-CEP*. Para ilustrar essa perspectiva de comunicação do framework, o Diagrama de Sequência é apresentado na Figura 5.2. Cabe observar que algumas classes ilustradas neste diagrama são específicas da aplicação que será implementada na seção 5.4.

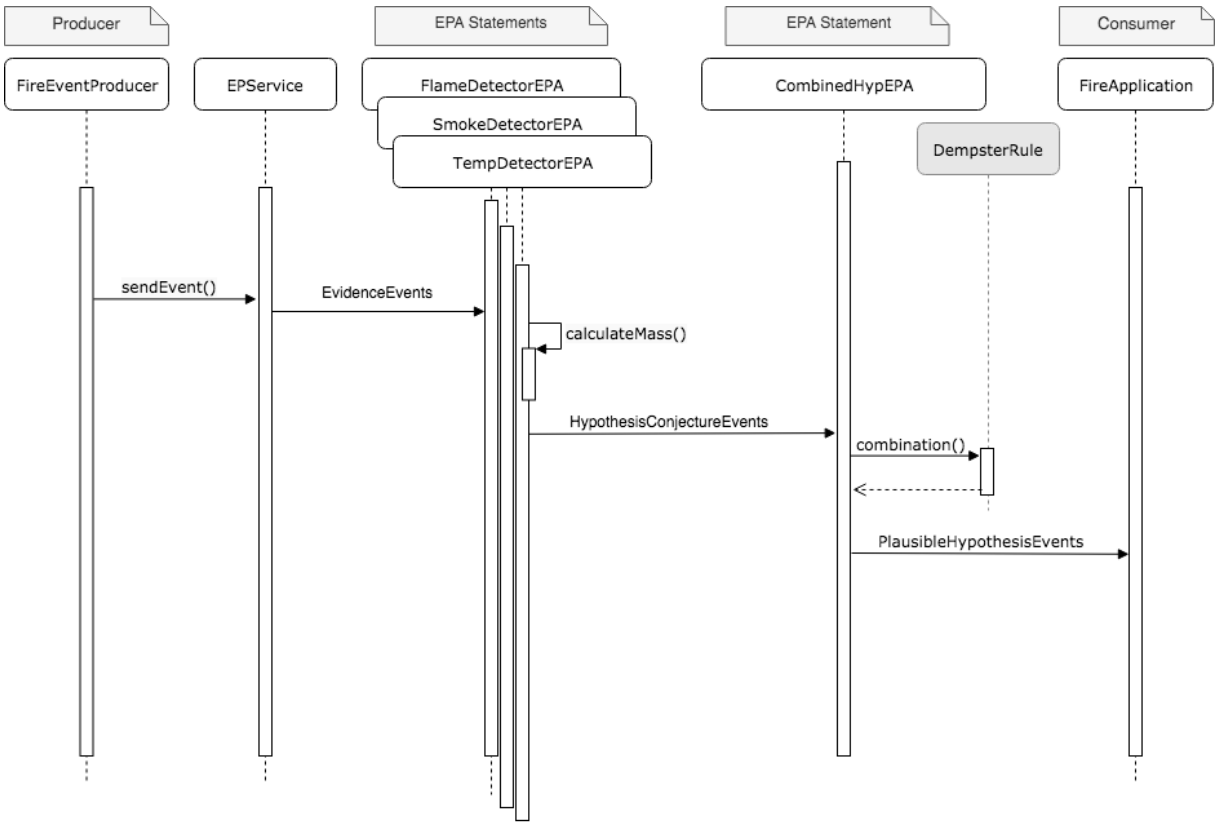


Figura 5.2: Diagrama de Sequência do Framework DST-CEP.

O diagrama de sequência (Figura 5.2) representa as interações entre objetos das classes do framework *DST-CEP*, realizadas através de operações com ênfase na ordenação temporal em que as mensagens são trocadas. Como ilustra a Figura

5.2, inicialmente os eventos primitivos (*EvidenceEvent*) são enviados pelo produtor de eventos. O motor Esper captura os eventos através das instâncias dos EPA's (classes da aplicação, Seção 5.4, Figura 5.6) que utilizam as funções de massa para calcular as massas dos eventos de conjecturas de hipóteses (*HypothesisConjectureEvent*). Em seguida, as conjecturas de hipóteses são combinadas pelo *CombinedHypEPA* utilizando a regra de combinação de Dempster para calcular a hipótese plausível (*PlausibleHypothesisEvent*), que finalmente é consumida pela aplicação.

5.2 Método de Desenvolvimento de uma Aplicação utilizando a abordagem *DST-CEP*

Esta seção apresenta um método para desenvolver aplicações de IoT que processam fluxo de dados utilizando a abordagem *DST-CEP* para tratar incerteza. O desenvolvimento nesse contexto inicia com a ideia da aplicação que irá processar e lidar com o problema de incerteza a partir de dados de sensores não confiáveis e termina com o produto pretendido (o software). Sendo assim, o método de desenvolvimento consiste em definir os passos que transformam a ideia inicial de solução em uma aplicação final em determinado domínio. Portanto, seguem os passos que devem orientar o desenvolvedor a fazer o uso da abordagem *DST-CEP* para construir aplicações:

1. O primeiro passo constitui realizar o **levantamento de requisitos**. Nesta etapa, o objetivo é compreender o problema que deverá ser tratado. Para isso, o levantamento de informações, estudos e requisitos, devem subsidiar os desenvolvedores a ter uma visão do que deve ser construído para solucionar o problema. Os desenvolvedores devem fazer uso dos requisitos levantados para construir as funções de massa da aplicação, as regras de processamento CEP e os modelos que representem a aplicação que será desenvolvida.

2. O segundo passo confere **projetar a arquitetura da aplicação** visando definir as classes, os objetos e suas relações, todos concebidos conceitualmente pelo modelo arquitetural *DST-CEP* já disponibilizado pela abordagem *DST-CEP*. O uso de diagramas UML são recursos recomendados nessa etapa.

3. O terceiro passo é de **implementação** e configura o desenvolvimento de classes específicas da aplicação, bem como o uso de interfaces e classes disponibilizadas pelo framework *DST-CEP* que devem dar suporte a esta etapa. Observa-se que o framework *DST-CEP* disponível e que será utilizado nessa etapa, está incluído na abordagem *DST-CEP*.

A abordagem *DST-CEP* propõe facilitar o desenvolvimento das aplicações. Isso deve ao fato da abordagem *DST-CEP* propor um método de desenvolvimento que descreve um conjunto de passos que orienta o desenvolvedor a construir aplicações. Além disso, a abordagem preconiza um modelo conceitual que apresenta o uso de funcionalidades da Teoria de Dempster-Shafer sob a perspectiva do processamento de eventos em uma rede EPN para lidar com dados de sensores não confiáveis. Para isso, um modelo arquitetural *DST-CEP* é concebido na abordagem *DST-CEP* para guiar a modelagem da solução da aplicação sendo desenvolvida. Por fim, o desenvolvedor possui o benefício de uso do framework *DST-CEP* para se preocupar apenas com as classes específicas do domínio da aplicação, ou seja, sem se preocupar em implementar classes que configuram a rede EPN, classes que gerenciam a produção dos eventos e classes que efetuam cálculos das funções da Teoria de Dempster-Shafer.

5.3 Estudo de Caso: Detecção de Incêndio

O estudo de caso descreve uma aplicação de IoT que detecta princípio de incêndio em tempo real baseada em dados de múltiplos sensores não confiáveis. Esta aplicação foi desenvolvida utilizando a abordagem *DST-CEP*. O primeiro passo a ser considerado no método de desenvolvimento da aplicação é o **levantamento de requisitos**. Para isso, o cenário da aplicação é descrito com o levantamento de informações suficientes para elaborar as funções de massa, algoritmos em alto nível e regras de processamento CEP na fase de implementação.

5.3.1 Cenário da Aplicação

Considere um sistema multisensor de detecção de incêndio que pode ser utilizado em aplicações de IoT, tais como *Smart Buildings*, *Smart Homes*, ou

Smart Factories. Este sistema consiste de sensores e agentes de processamento de eventos. Os sensores são componentes automáticos do sistema de detecção de incêndio, que incluem sensores de chama, fumaça e temperatura. Eles são capazes de detectar rapidamente informações físicas e químicas geradas por um incêndio, e transmiti-las aos agentes de processamento para detectar o incêndio. Estes agentes de processamento de eventos são chamados detectores, uma vez que utilizam informações baseadas em sensores para detectar a ocorrência ou não de incêndio.

Para este domínio, com base na NBR-17240[5], devem ser definidos: o tipo de sistema de detecção, lógica de funcionamento e ações a serem tomadas para cada evento do sistema. Dentre os tipos de sistemas de detecção, existem os sistemas de detecção convencional e endereçável (cada detector recebe um endereço), que não permitem o ajuste do nível de alarme dos dispositivos de detecção. Em especial, para o estudo de caso da abordagem *DST-CEP*, considera-se um sistema de detecção algorítmico que permite o ajuste do nível de alarme e monitora continuamente os valores dos dispositivos de detecção, comparando-os com aqueles previamente definidos para aquela instalação ou ambiente.

Quando esse sistema usa um único sensor para coletar informações, ele pode se tornar não confiável devido à interferência causada por poeira, eletromagnetismo, vapor de água, luz, vibração ou outras condições do ambiente. Eventualmente, o sistema não consegue efetivamente distinguir entre sinais de incêndio e sinais de interferência do ambiente. Consequentemente, ele não consegue enviar avisos de incêndio de maneira confiável e imediata, podendo ainda gerar falsos alarmes. Por exemplo, um detector de chamas não pode ser usado de forma generalizada, pois a chama não é a única fonte de radiação infravermelho. Qualquer superfície quente, tais como fornos, lâmpadas halógenas ou incandescentes, emitem radiações que coincidem com a radiação da chama, assim ocasionando falsos alarmes. Detectores de chamas podem responder mais rapidamente a um incêndio com chama do que um detector de temperatura ou fumaça, no entanto, são inadequados para incêndios de combustão lenta. Um sensor de gases e fumaça muito sensível pode disparar perante qualquer indício de fumaça. O mesmo sensor, quando mal regulado com uma alta sensibilidade, se torna um problema, podendo ocasionar falsos alarmes

[5]http://www.segmafire.com.br/wp-content/uploads/sites/179520/2017/06/NBR-17240-2010-Substituindo-NBR-9441-Alarme.pdf

com a fumaça produzida ao cozinhar, com um cigarro ou até mesmo com o vapor de um banho quente. Considerando este cenário de aplicação, o sistema de detecção de incêndio deve utilizar três sensores (sensor de temperatura, gases/fumaça e chama) para melhorar a confiabilidade das detecções de incêndio e não incêndio. Portanto, lidar com as incertezas inerentes e o processamento de informações coletadas de múltiplos sensores não confiáveis e possivelmente conflitantes torna-se um problema neste cenário.

5.3.2 Funções de Massa da Aplicação

Na abordagem *DST-CEP*, cada hipótese no quadro discernimento deve receber um valor de massa representado por uma função de massa. Os detectores são capazes de detectar as hipóteses de incêndio (*Fire*) e não incêndio (*Non-Fire*) através da captura de informações físicas e químicas geradas pelo princípio da combustão ou fogo. A fase inicial de incêndio é levada em consideração no estudo e, por isso, foram modelados detectores mais restritivos capazes de detectar o menor sinal ou evidência de incêndio. As funções de massa aqui implementadas são adequadas para valores quantitativos de sensores, onde tem-se um limiar utilizado como parâmetro para o acionamento de ações a partir dos valores lidos. Em *DST-CEP*, foram propostos algoritmos adaptados da técnica *Min – Max Normalization* [11, 34] considerando intervalos de incerteza dos detectores e a elaboração dos algoritmos deve garantir a identificação de um princípio de incêndio, no menor tempo possível.

Função de Massa do Sensor de Chama

Para ilustrar o funcionamento da primeira função de massa, inicialmente considera-se um sensor de chama utilizado para detectar qualquer tipo de ignição e princípio de chama, e realizar notificações (*Fire*, *Non-Fire*) com os respectivos valores de massa. O princípio de funcionamento do sensor de chama, neste exemplo o Flame detection Sensor[6] Module LM393, consiste em detectar radiação infravermelha (IR) gerada pela chama. Segundo as leis da termodinâmica, o calor à superfície de um corpo é transformado em radiação que se propaga no vazio. Esta radiação é análoga à da luz visível, mas com comprimento de onda superior, situando-se no domínio dos

[6]https://circuits-diy.com/ir-infrared-flame-sensor-module-arduino/

infravermelhos (IR). Desta forma, ao atingir a superfície de um outro meio, uma parte da radiação é refletida, outra transmitida e outra absorvida, degradando-se em calor. O fenômeno da chama é associado aos corpos emissores de radiações (raios IR), cuja transmissão se faz uniformemente em todas as direções. O sensor de chama (Flame detection Sensor Module LM393) pode detectar radiação IR com um comprimento de onda que varia de 750nm a 1100nm (nanômetros). Portanto, assumimos um *threshold* de $thr = 750nm$ como limiar para a detecção de chama, sendo sensível ao espectro da chama. A geração de falsos alarmes é um problema para qualquer detector de incêndio. A radiação IR é um exemplo de radiação invisível ao olho humano, que é caracterizada por comprimentos de onda entre 700 e 1.000.000 nm. Especificamente, para o detector de chama pode-se considerar outras fontes de emissão de radiação abaixo de 750nm, que não seja o fogo, na área de cobertura a ser protegida e que poderá originar falsos alarmes. Por exemplo, flashes de luz, cigarros acesos, solda, luz solar (direta ou refletida), radiação de superfícies quentes, etc. A precisão do detector é de aproximadamente +/− 45nm. Como ilustra o Algoritmo 1 da função de massa, o mesmo consiste em capturar diretamente o dado do sensor (epl) e comparar se o valor lido excede o limiar (thr) específico do sensor. Os valores $Max(epl)$ e $Min(epl)$ são extraídos a partir do erro (+/−) do detector. O Algoritmo 1 transforma um valor de leitura do sensor (epl) em um valor de massa (m) dentro do intervalo $[x - y]$. Um exemplo numérico é apresentado na Figura 5.3.

Neste exemplo (Figura 5.3), observa-se que um valor de leitura epl= 760nm ultrapassa o limiar thr = 750nm por uma pequena diferença, assim indicando a hipótese de incêndio. Nesta situação, são setados os valores $Max(epl)$ e o intervalo x e y como ilustra a figura. Todos os valores então são utilizados como base para o cálculo do valor de massa (m) e, para esta primeira leitura, com epl= 760nm, a hipótese de incêndio tem o valor de massa $m(Fire)$ = 0.61 e, por definição, na Equação 2.2, tem-se $m(NoFire)$ = 0.39. A próxima leitura no exemplo tem o valor epl = 790nm. Assim, os valores de massa calculados são $m(Fire)$ = 0.94 e $m(NoFire)$ = 0.06. A última leitura no exemplo é epl = 715nm, resultando em $m(Fire)$ = 0.11 e $m(NoFire)$ = 0.89.

Algoritmo 1: Flame Mass Function.

Data: Value of sensor readings (epl);

Threshold thr; $Max(epl)$ and $Min(epl)$ values from sensor data;

Interval x and y;

Result: Mass value (m)

Initialization;

if *new data sensor (epl)* **then**

  Read current epl data sensor;

  if *($epl > thr$)* **then**

    Set $Max(epl)$

    Set interval: upper x, lower y

    $m = \frac{(epl-thr)}{Max(epl)-thr} * (x - y) + y$

  else

    Set $Min(epl)$

    Set interval: upper x, lower y

    $m = \frac{(epl-Min(epl))}{thr-Min(epl)} * (x - y) + y$

  end

  return m;

end

Função de Massa do Sensor de Temperatura

Detectores de temperatura são utilizados para monitorar ambientes cuja característica de combustão pode gerar muito calor no início e pouca fumaça. São apropriados para espaços pequenos confinados, onde se esperam fogos rápidos e de alta temperatura. Também são indicados para ambientes com vapor, gases ou muitas partículas em suspensão, onde os detectores de fumaça estão sujeitos a alarmes indesejáveis. Existem duas abordagens de detecção e alerta de incêndio. A primeira é a detecção de máxima temperatura, que atua quando o sensor reconhece uma temperatura limite prefixada. A segunda abordagem, que será utilizada na função de massa, é a detecção termodiferencial, que reage à taxa de variação de temperatura. O sensor de temperatura utilizado é o DHT11, que é um sensor composto por um termistor, ou seja, um resistor sensível à variações de temperatura. Assumimos a taxa de variação de temperatura de 10ºC/minuto, que é um valor coerente com produtos comerciais e norma NBR-17240. A precisão do sensor DHT11 é de +/– 2ºC.

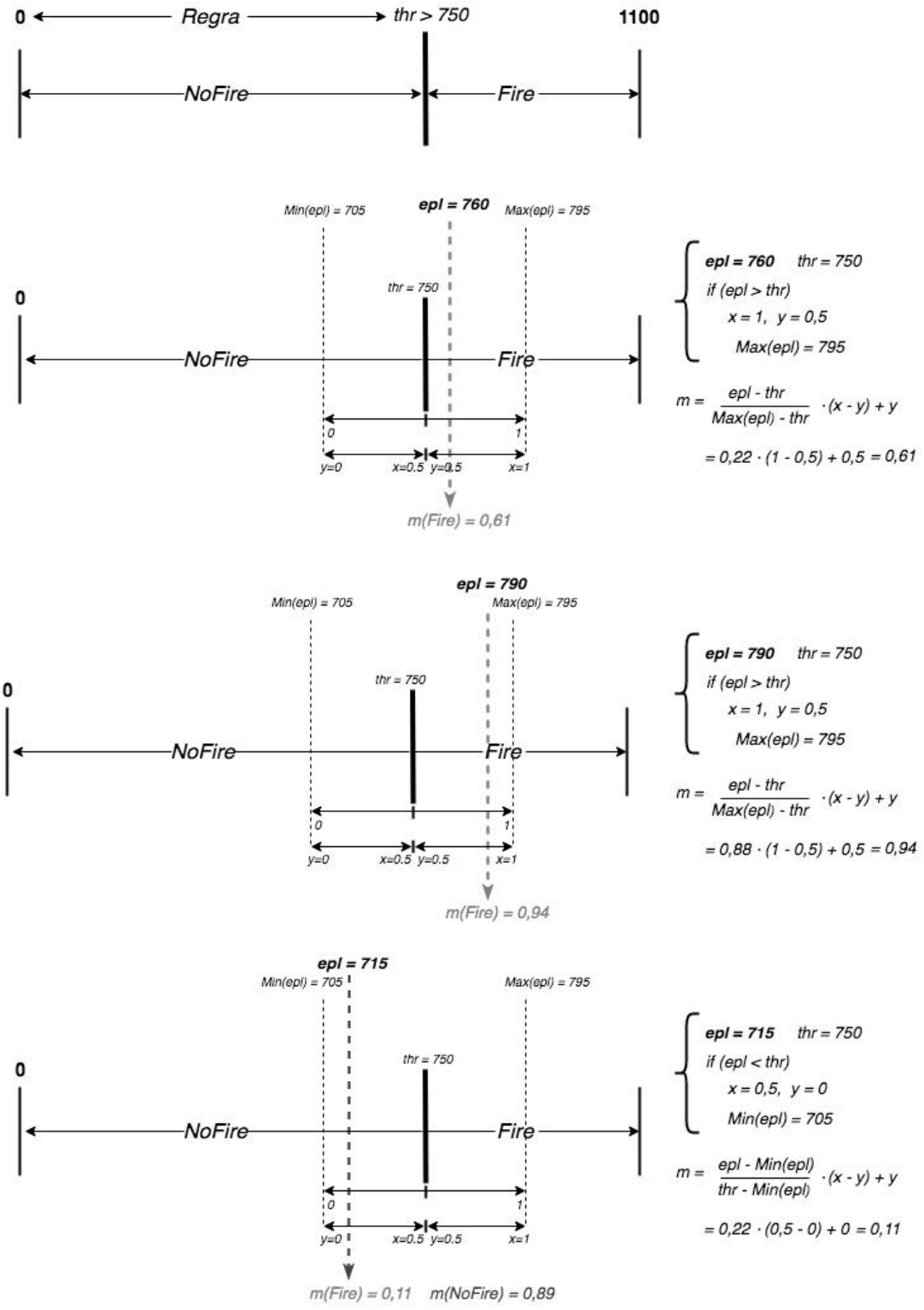


Figura 5.3: Exemplo numérico da função de massa p/ diferentes leituras do sensor de chama.

Como ilustra o exemplo numérico, na parte superior da Figura 5.4 a partir da primeira leitura epl_1= 38ºC é possível calcular o limiar de thr = 48ºC considerando uma variação de temperatura de 10ºC de diferença. Ou seja, se dentro de 1 minuto a próxima leitura do sensor for acima de 48ºC temos uma hipótese de incêndio. Na parte inferior da figura 5.4, tem-se uma segunda leitura, epl_2 = 49.3ºC, que está acima

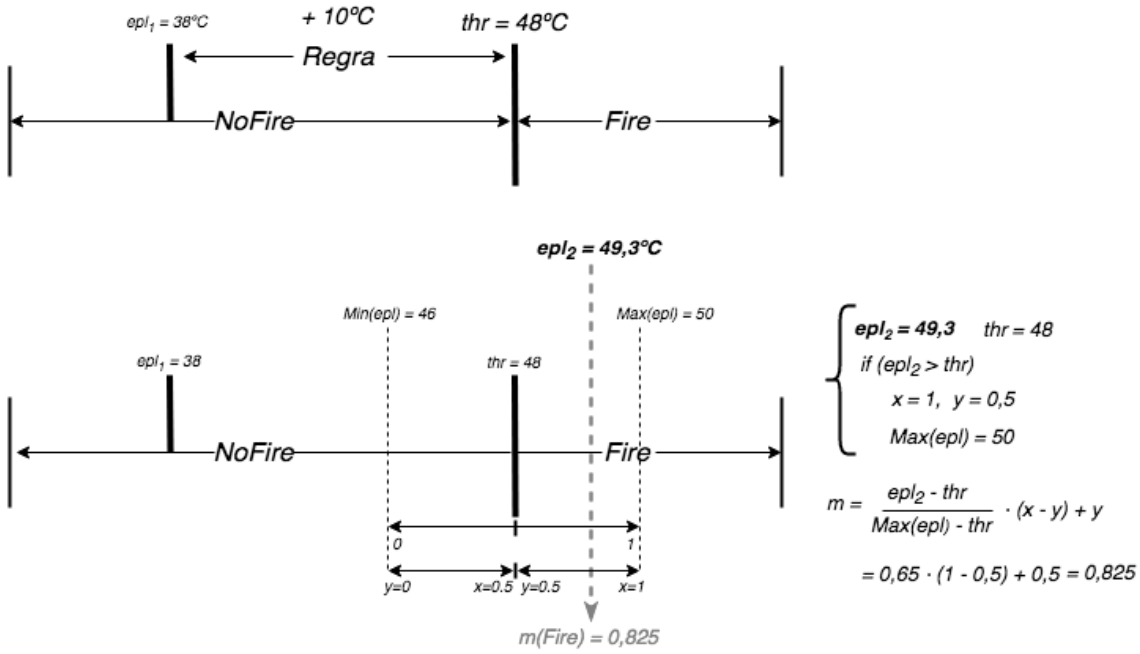


Figura 5.4: Exemplo numérico da função de massa p/ o detector de temperatura.

do limiar, entretanto dentro da faixa de imprecisão (+/− 2ºC). Nesta situação, são setados os valores $Max(epl)$ e o intervalo x e y como ilustra a figura. Todos os valores então são utilizados como base para o cálculo do valor de massa (m), que para esta leitura epl= 49.3ºC a hipótese de incêndio tem o valor de massa $m(Fire)$ = 0.825 e $m(NoFire)$ = 0.175.

Como ilustra o Algoritmo 2 da função de massa do detector de temperatura, o mesmo consiste em capturar diretamente o dado do sensor, por exemplo epl_1 e em seguida calcular o limiar thr somando-se 10ºC ao valor de epl_1. Observa-se que um limiar é gerado e verificado a cada nova leitura dentro de 1 min. Quando a próxima leitura epl_2 ocorrer dentro do intervalo de 1 min (janela de tempo deslizante), então é verificado se epl_2 é maior que o limiar thr atual. Se sim, o valor $Max(epl)$ é setado e o restante do Algoritmo 2 converte o valor de leitura do sensor (epl_2) em um valor de massa (m) dentro do intervalo $[x - y]$.

Função de Massa do Sensor de Fumaça

Seguindo o domínio das hipóteses de incêndio, é apresentada a função de massa considerando neste exemplo o sensor[7] MQ-2, que é um detector que monitora os níveis dos gases de combustão presentes no ambiente, por exemplo GLP (Gás

[7]https://sandboxelectronics.com/?p=165

Algoritmo 2: Temperature Mass Function.

Data: Values of current sensor readings epl_1, epl_2;

Threshold thr; $Max(epl)$ and $Min(epl)$ values from thr;

Interval x and y;

Result: Mass value (m)

Initialization;

while *new data sensor from sliding window (1 min)* **do**

  $thr = epl_1 + 10^oC$;

  if $(epl_2 > thr)$ **then**

    Set $Max(epl)$

    Set interval: upper x, lower y

    $m = \frac{(epl_2 - thr)}{Max(epl) - thr} * (x - y) + y$

  else

    Set $Min(epl)$

    Set interval: upper x, lower y

    $m = \frac{(epl_2 - Min(epl))}{thr - Min(epl)} * (x - y) + y$

  end

  return m;

end

Liquefeito de Petróleo), Propano, Metano, Monóxido de Carbono e outros tipos de gases. Dentre suas aplicações, ele pode ser utilizado em um sistema de detecção precoce de incêndio. A detecção dos gases de combustão dá-se numa fase anterior a uma detecção de aumento de temperatura. O funcionamento do sensor é baseado na variação da sua resistência interna, ou seja, ao entrar em contato com determinado gás, sua resistência interna é modificada conforme a concentração daquela substância. A detecção ocorre através da proporção de um gás em relação a outro, portanto em ppm (partes por milhão). Ao monitorar as leituras do sensor é possível configurar limiares para realizar determinadas ações. Especificamente, este detector também é projetado para detectar monóxido de carbono (CO) de qualquer fonte de combustão. O monóxido de carbono é um gás venenoso que pode ser fatal quando inalado, pois inibe a capacidade do sangue de transportar oxigênio, deixando uma vítima desorientada, não conseguindo se salvar ou sair de um ambiente (ex.: prédio) para pedir assistência. Concentrações de CO entre 1 e 30 ppm geralmente podem ocorrer em condições

normais do dia a dia, podendo ser uma indicação de uma condição transitória que pode aparecer hoje e nunca reaparecer. Tal condição poderia ser um pré-alarme de CO com baixo nível, podendo se transformar em uma concentração de CO prejudicial. A partir de uma concentração de 50 ppm de CO, o sangue humano dá preferência ao monóxido de carbono, ao invés de reagir com o oxigênio disponível no ar, desativando 7% da capacidade da hemoglobina de transportar oxigênio. De acordo com o *Consumer Product Safety Commission*[8], do governo dos Estados Unidos, conforme os níveis de CO aumentam acima de 70ppm, os sintomas se tornam mais perceptíveis e podem incluir dor de cabeça, fadiga, náuseas e pessoas cardíacas podem sentir um aumento da dor no peito. Segundo a literatura, com 100ppm de concentração de CO há uma forte tendência a asfixia do ser humano. Portanto, assumimos um *threshold* de thr = 70ppm como limiar para a geração de alerta. Os sensores da família MQ-x são sensores que passam por um procedimento de calibração onde o coeficiente de variação é de 0.03, utilizado por pesquisadores na avaliação da precisão ou desvio padrão relativo.

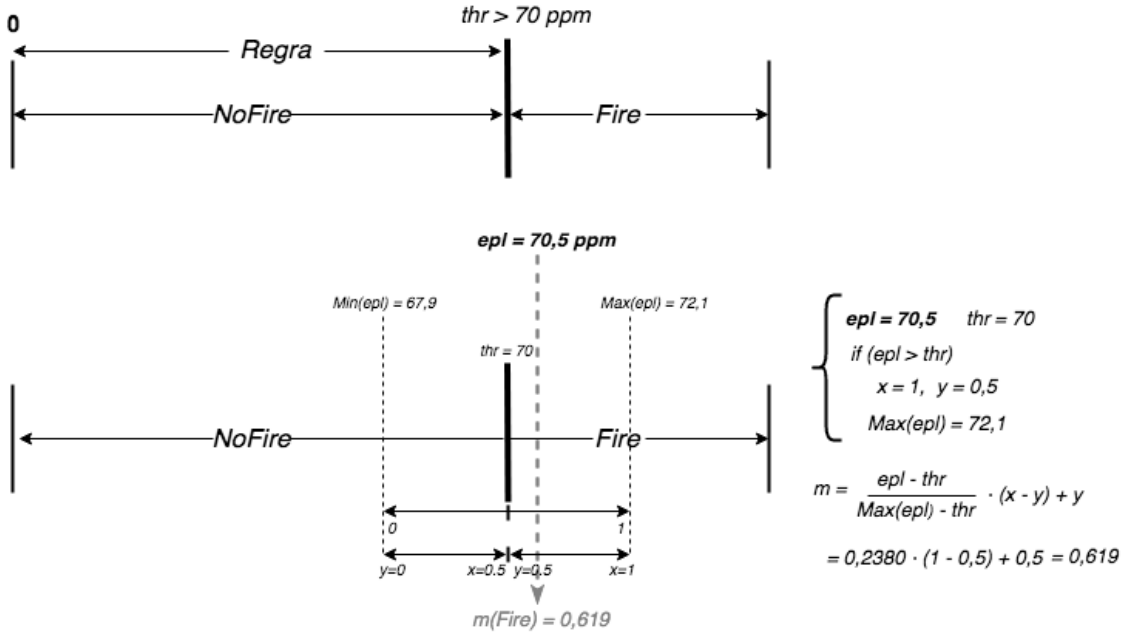


Figura 5.5: Exemplo numérico da função de massa do detector de fumaça.

Como ilustra o Algoritmo 3 da função de massa, o mesmo consiste em capturar diretamente o dado do sensor (epl) e comparar se o valor lido excede o limiar (thr) específico do sensor. Os valores $Max(epl)$ e $Min(epl)$ são extraídos e o restante

[8]https://www.cpsc.gov/Safety-Education/Safety-Education-Centers/Carbon-Monoxide-Information-Center/Carbon-Monoxide-Questions-and-Answers

Algoritmo 3: Smoke Mass Function.

Data: Value of sensor readings (epl);
Threshold thr; $Max(epl)$ and $Min(epl)$ values from sensor data;
Interval x and y;
Result: Mass value (m)
Initialization;
if *new data sensor (epl)* **then**
  Read current epl data sensor;
  if *($epl > thr$)* **then**
    Set $Max(epl)$
    Set interval: upper x, lower y
    $m = \frac{(epl-thr)}{Max(epl)-thr} * (x - y) + y$
  else
    Set $Min(epl)$
    Set interval: upper x, lower y
    $m = \frac{(epl-Min(epl))}{thr-Min(epl)} * (x - y) + y$
  end
  return m;
end

do Algoritmo 3 transforma um valor de leitura do sensor (epl) em um valor de massa (m) dentro do intervalo $[x - y]$. Um exemplo numérico é apresentado na Figura 5.5.

Neste exemplo (Figura 5.5), observa-se que um valor de leitura epl = 70.5ppm ultrapassa o limiar thr = 70ppm. Nesta situação, são setados os valores $Max(epl)$ e o intervalo x e y, como ilustra a figura. Todos os valores então são utilizados como base para o cálculo do valor de massa (m), que para esta leitura de epl= 71.5ppm, o valor de massa é $m(Fire)$ = 0.619.

5.4 Implementação do Estudo de Caso

O segundo passo a ser considerado no método de desenvolvimento é **projetar a arquitetura da aplicação**. Para isso, são definidas classes, objetos e suas relações, todos concebidos conceitualmente pelo modelo arquitetural *DST-CEP* disponível na abordagem *DST-CEP*. Diagramas UML são recursos recomendados

nessa etapa. A implementação da aplicação de detecção de incêndio faz o uso de classes do framework *DST-CEP*. A criação da arquitetura da aplicação através de classes do framework é apresentada no diagrama de classes da Figura 5.6. As classes com rótulos dos nomes (cabeçalhos) em azul são implementações de classes do framework *DST-CEP*, enquanto as classes com cabeçalho branco são específicas da aplicação de detecção de incêndio (ou *FireApplication*). Como apresentado na seção 5.1, o framework *DST-CEP* possui um conjunto de classes e interfaces que facilitam a construção de aplicações. Assim, observa-se na Figura 5.6 que ao utilizar o framework, o trabalho do desenvolvedor consiste em prover as classes que são específicas do domínio da aplicação e que em sua maioria implementam ou estendem as classes do framework *DST-CEP*.

Seguindo o uso da estrutura do framework *DST-CEP*, as classes da aplicação *FireApplication* podem ser divididas em três categorias: classes que realizam a configuração da rede EPN, classes que realizam a produção dos eventos e as classes que processam os eventos (EPA's). Como ilustra a Figura 5.6:

1. A classe da aplicação que inicia a configuração da EPN é a *FireEPNConfiguration*. Nesta classe o método *initService()* inicia as configurações do framework. Além disso, esta mesma classe apenas instancia e cria os EPA´s da rede EPN através dos EPStatements e StatementsSubscribers. Para isso, o método *createDetectorCheckEPL()* é usado através da classe da aplicação que estende a classe do framework *EPNConfiguration* tornando os EPA's ativos na rede EPN.

2. A classe responsável por iniciar a produção dos eventos específicos da aplicação é a classe *FireEventProducer* através do método *startReadings()*. O método *handle()* usa os eventos da aplicação *FlameEvidenceEvent*, *TemperatureEvidenceEvent* e *SmokeEvidenceEvent*, onde todos estendem a classe *EvidenceEvent* do framework.

3. As classes da aplicação responsáveis pelo processamento dos eventos são *TempDetectorEPA, FlameDetectorEPA, SmokeDetectorEPA* e *CombinedHypEPA*. Cada classe EPA implementa a interface *StatementSubscriber* possuindo a declaração da consulta EPL (tempDetectorEPL, flameDetectorEPL, smokeDetectorEPL e combinedHypEPL). Vale observar na Figura 5.6 que os EPA's dos três detectores realizam chamadas aos métodos da classe *MassSpecialist* para realizar o cálculo das funções de massa, apresentadas na seção 5.3.2, de cada

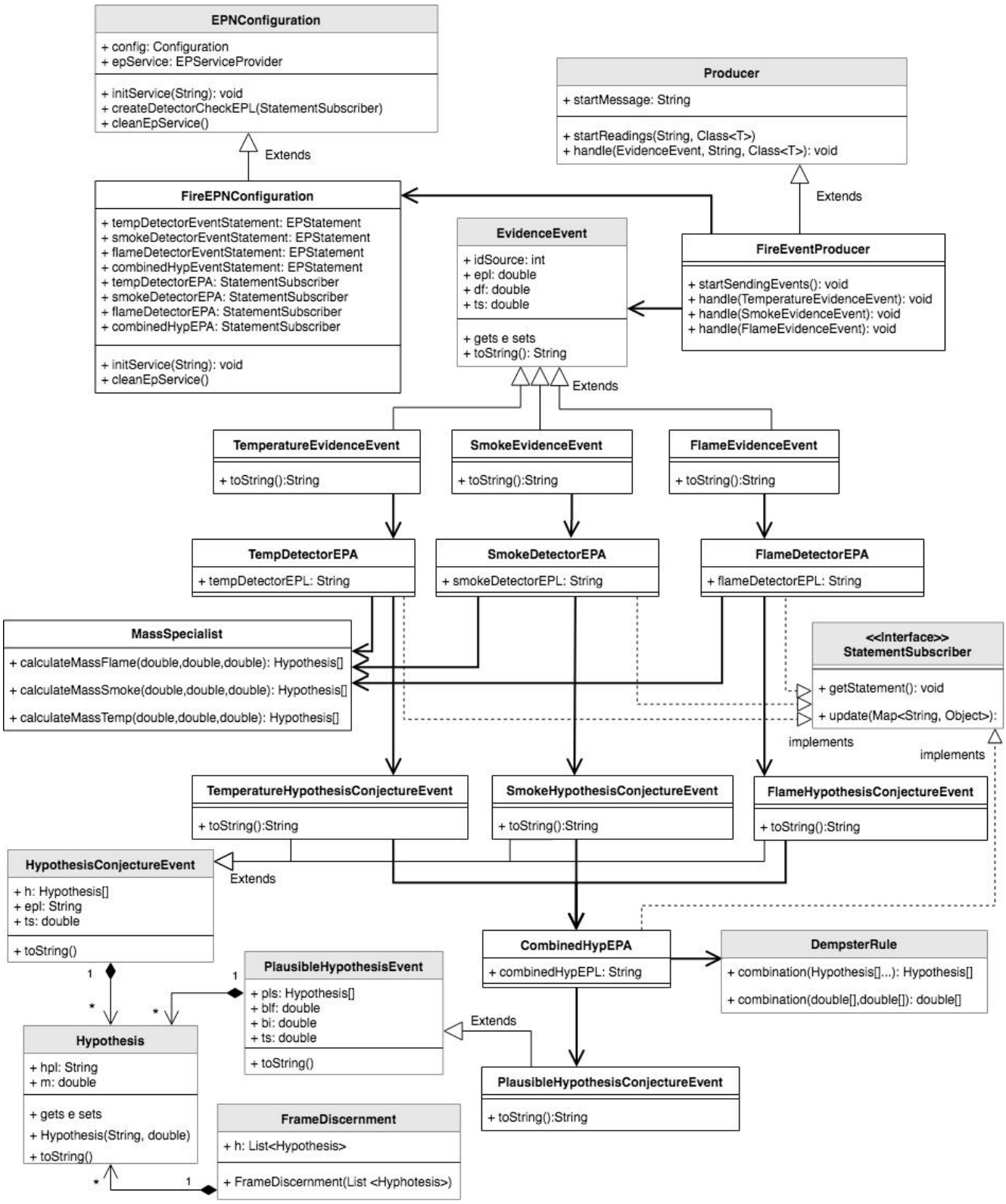


Figura 5.6: Diagrama de Classes da Aplicação e Framework DST-CEP.

detector. Os resultados desse processamento são os eventos de conjectura de hipóteses representados pelas classes *TemperatureHypothesisConjectureEvent*, *SmokeHypothesisConjectureEvent* e *FlameHypothesisConjectureEvent*. Cada uma

dessas classes estende *HypothesisConjectureEvent* do framework. Tais eventos são entradas para a regra de combinação de Dempster. A classe da aplicação *CombinedHypEPA* realiza a regra de combinação de Dempster através da chamada do método externo *combination()* da classe *DempsterRule* do framework. O resultado da combinação (saída da rede EPN) é o evento de hipótese plausível representado pela classe da aplicação *PlausibleHypothesisConjectureEvent*, que estende a classe *PlausibleHypothesisEvent* do framework.

5.4.1 Implementação da Aplicação nos Níveis de Processamento do Framework *DST-CEP*

O terceiro passo a ser considerado no método de desenvolvimento é a **implementação** da aplicação. Nesta etapa ocorre o desenvolvimento de classes específicas da aplicação, bem como o uso de interfaces e classes disponibilizadas pelo framework *DST-CEP*. A estrutura do framework *DST-CEP* é dividida em níveis de processamento, seguindo o modelo arquitetural *DST-CEP* (Figura 4.1). A implementação da aplicação de detecção de incêndio segue este modelo, como ilustra a Figura 5.7, onde se destacam três níveis de processamento: sensores, conjectura de hipóteses e combinação de hipóteses. Esta seção descreve como os eventos da aplicação de IoT de detecção de incêndio são processados e calculados nos níveis de processamento do framework *DST-CEP*.

Como ilustra a Figura 5.7, inicialmente o quadro de discernimento é composto de hipóteses primitivas definidas para esta aplicação de detecção de incêndio que são $\Theta = \{Fire, Non\text{-}fire\}$. Todos os subconjuntos formados pelas hipóteses primitivas dão origem a todas as hipóteses possíveis formadas por $2^{\Theta} = \{\emptyset, \{Fire\}, \{Non\text{-}fire\}, \Theta\}$, onde $Fire$ indica que um incêndio está ocorrendo, $Non\text{-}Fire$ indica que nenhum incêndio está ocorrendo, Θ denota incerteza sobre se há um incêndio ou não incêndio. O valor da massa do conjunto vazio é zero por definição (Equação 2.2).

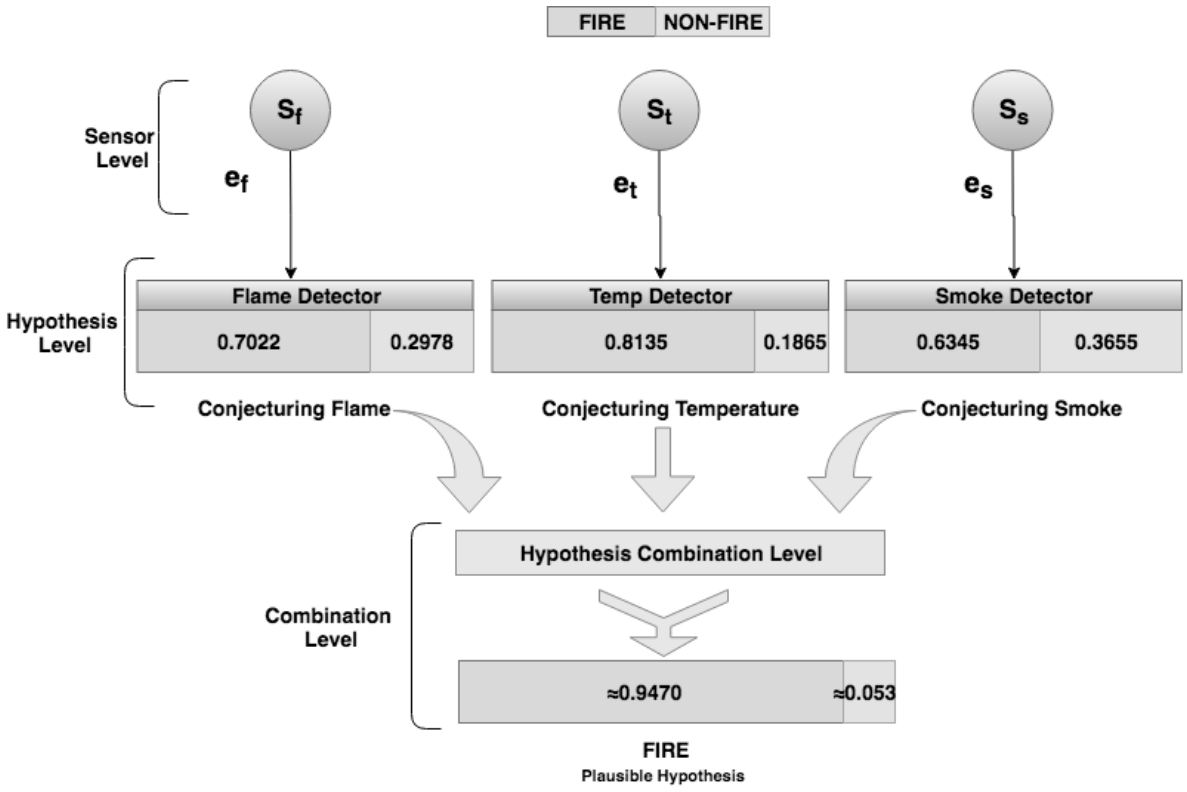


Figura 5.7: Níveis de processamento computacional *DST-CEP* do sistema de detecção de incêndio.

Nível de Sensores

No nível de sensores (Figura 5.7), três detectores enviam leituras (eventos de evidência) para o nível de hipóteses. O valores das leituras dos eventos de evidências apresentados ao longo desta seção foram capturados do *dataset* de sensores. Dada a definição formal do evento na Seção 4.1, uma instância do evento de evidência de chama (do inglês, *FlameEvidenceEvent*) é apresentada, como segue:

```
FlameEvidenceEvent(idSource=1, ts=2020-06-07T19:06:10, 768.2, 45)
```

O *FlameEvidenceEvent* é composto por uma identificação *idSource* do sensor, o *ts* é o *timestamp* do evento, o valor *epl* (*evidence payload*) lido pelo sensor de chama é 768.2 nm, o fator de desconto *df* relacionado a precisão é +/− 45nm. Este evento de evidência é então enviado para o nível de conjectura de hipóteses. Da mesma forma que o detector de chama, todos os outros dois detectores enviam leituras de eventos de evidência.

```
SmokeEvidenceEvent(idSource=2,ts=2020-06-07T19:6:10, 70.56, 0.03)
```

```
TemperatureEvidenceEvent(idSource=3, ts=2020-06-07T19:6:10, 49.44, 2)
```

O *SmokeEvidenceEvent* é composto pela identificação do sensor, o *timestamp* do evento, o valor *epl* (*evidence payload*) lido pelo sensor de fumaça que é 70.56 ppm, o fator de desconto *df* relacionado a precisão é +/− 0.03. Em seguida, o *TemperatureEvidenceEvent* é composto pela identificação do sensor, o *timestamp* do evento e o valor *epl* (*evidence payload*) lido pelo sensor de temperatura que é 49.44ºC; o fator de desconto *df* relacionado à precisão é +/− 2ºC.

Nível de Hipóteses

O nível de conjecturas de hipóteses recebe os eventos de evidência e realiza o cálculo das massas para hipóteses "yes fire" (*yf*) e "non-fire" (*nf*) da aplicação através dos algoritmos das funções de massa apresentados na seção 5.3.2. Para ilustrar o cálculo das massas para as hipóteses, tomemos como exemplo o *FlameEvidenceEvent*. Como foi apresentado no Algoritmo 1 (Seção 5.3.2), o sensor de chama pode detectar chamas com comprimentos de onda a partir do limiar (*thr*) de 750 nm. O Algoritmo 1 da função de massa do detector de chama recebe os dados do sensor de chama epl = 768.2 nm e verifica se o valor excede o limiar thr = 750.0. Se sim, então a hipótese "yes fire" é confirmada (*yf*) e o valor de massa para esta hipótese deve ser calculado. Os valores $epl = 768.2nm$, $thr = 750$ e $Max(epl) = 795$ são utilizados na computação da massa, segundo a equação do Algoritmo 1. Os resultados dos valores de massa são então obtidos: yf= 0.7022 e nf= 0.2978. Finalmente, tais valores de massa são carregados no evento de conjectura de hipótese do detector de chama *FlameHypothesisConjectureEvent*, como segue:

```
FlameHypothesisConjectureEvent(idSource=1, ts=2020-06-07T19:6:10, 768.2, 45, (
    yf, 0.7022), (nf, 0.2978))
```

O evento acima representa uma instância da definição formal do evento de conjectura de hipótese (*hc*) apresentada na Seção 4.1. Ou seja, são eventos de evidência enriquecidos com informações das hipóteses e suas respectivas massas (yf= 0.7022 e nf= 0.2978.). Como ilustra a Figura 5.7, da mesma forma que o detector de chama, os outros dois detectores geram conjecturas de hipóteses. Logo, as funções de massa

dos detectores de fumaça e temperatura geram as seguintes instâncias de eventos de conjecturas de hipóteses:

```
SmokeHypothesisConjectureEvent(idSource=2, ts=2020-06-07T19:6:10, 70.56, 0.03,
    (yf, 0.6345), (nf, 0.365))
```

```
TemperatureHypothesisConjectureEvent(idSource=3, ts=2020-06-07T19:6:10, 49.44,
    2, (yf, 0.8135), (nf, 0.1865))
```

As funções de massa são aplicadas com base nos algoritmos já apresentados. Entretanto, a execução dos algoritmos é iniciada por um agente de processamento de eventos (EPA) contendo uma declaração EPL. Tal EPL é uma consulta responsável pela seleção e filtragem dos eventos, chamadas às funções de classes externas à EPL, além de incluir todo o resultado do processamento em um fluxo de eventos derivados de saída. Por exemplo, o Código 5.1 representa o EPA do detector de chamas denominado *FlameDetectorEPA*. Este EPA possui uma declaração *flameDetectorEPL*, onde inicialmente os dados *payload* e precisão (epl, df) do sensor são extraídos do fluxo de eventos *FlameEvidenceEvent()* (linha 4) e processados pela função de massa (*calculateMassFlame*) deste detector (linha 3). Observa-se nesta EPL a opção de implementar as funções de massa escritas em uma classe Java (*MassSpecialist*) que calcula os valores de massa. Esta mesma classe é importada e seus métodos acessados de dentro da declaração EPL. Os resultados das funções de massa (linha 3) são valores de massa associados às hipóteses e tais resultados são atribuídos a *h* (*array* de *Hypothesis[] h*) que são inseridas no fluxo de conjectura de hipóteses *FlameHypothesisConjectureEvent* (linha 1), que posteriormente será consumido pela regra de combinação de Dempster utilizada para combinar hipóteses.

Code 5.1: FlameDetectorEPA.

```
1 insert into FlameHypothesisConjectureEvent
2 select Flame.epl as epl, Flame.ts as ts,
3 lsdi.ufma.br.cep.dstheory.MassSpecialist.calculateMassFlame(Flame.epl,
      Flame.df) as h
4 from pattern [every Flame=FlameEvidenceEvent()
5 where timer:within(30 seconds)];
```

O Código 5.2 representa o EPA detector de temperatura. Inicialmente os dados *payloads* de temperaturas e precisão do sensor são extraídos do fluxo de eventos *TemperatureEvidenceEvent()* (linha 4), em seguida processados pela função de massa (*calculateMassTemp*) deste detector (linha 3). Dada a regra de negócio deste detector, que é capturar a variação de temperatura acima de 10ºC dentro de 1 min, a linha 3 detecta o padrão de duas leituras consecutivas em 1 min, e tais leituras são enviadas para função de massa verificar as variações de temperaturas e calcular os respectivos valores de massa. Vale observar que o critério de comparação de 10ºC de variação poderia ficar no nível da regra CEP. Entretanto, seria necessário implementar uma segunda regra para calcular o valor de massa da hipótese de não incêndio, ou seja, quando não houvesse uma variação de 10ºC/min entre as leituras de temperatura. Portanto, optou-se por uma regra que captura as últimas duas leituras consecutivas dentro de 1min, enquanto a função de massa verifica as hipóteses e realiza o cálculo dos valores de massas relacionados. Enquanto consome novos eventos, o detector captura variações do sensor de temperatura entre os eventos dentro da janela de tempo de 1 min.

Code 5.2: TempDetectorEPA.

```
1 insert into TemperatureHypothesisConjectureEvent
2 select t1.epl as temp1, t2.epl as temp2, t2.ts as ts,
3 lsdi.ufma.br.cep.dstheory.MassSpecialist.calculateMassTemp(t1.epl, t2.epl,
4 t2.df) as h from pattern [every t1=TemperatureEvidenceEvent() ->
5 t2=TemperatureEvidenceEvent()
6 where timer:within(1 minutes)];
```

O Código 5.3 representa o EPA do detector de fumaça. Inicialmente os dados *payload* e precisão (epl, df) do sensor são extraídos do fluxo de eventos *SmokeEvidenceEvent()* (linha 4) e processados pela função de massa (calculateMassSmoke) deste detector (linha 3). Os resultados das funções de massa (linha 3) são valores de massa associados às hipóteses e tais resultados são atribuídos a *h* (*array* de *double[] h*) que são inseridos no fluxo de conjectura de hipóteses *SmokeHypothesisConjectureEvent* (linha 1) que posteriormente será consumido pela regra de combinação de Dempster utilizada para combinar hipóteses.

Code 5.3: SmokeDetectorEPA.

```
1 insert into SmokeHypothesisConjectureEvent
2 select Smoke.epl as epl, Smoke.ts as ts,
3 lsdi.ufma.br.cep.dstheory.MassSpecialist.calculateMassSmoke(Smoke.epl,
      Smoke.df) as h
4 from pattern [every Smoke=SmokeEvidenceEvent()
5 where timer:within(30 seconds)];
```

Nível de Combinação de Hipóteses

Finalmente, no Nível de Combinação de Hipóteses o processamento dos dados coletados de várias fontes de informação são combinados para obter a identificação de incêndio mais precisa (Figura 5.7). Nesta etapa, a combinação das hipóteses geradas por múltiplos detectores ocorre na saída. Como ilustra a Figura 5.8, o resultado das hipóteses combinadas permite detectar a hipótese mais plausível (*pls*).

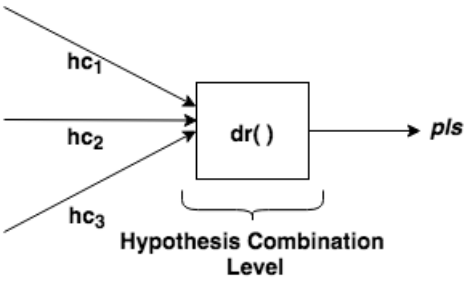


Figura 5.8: Nível de Combinação de Hipóteses *DST-CEP* (recorte da Figure 4.1).

A partir de um conjunto de conjecturas de hipóteses derivadas, a regra de combinação de Dempster $dr()$ é utilizada com as Equações 4.5 e 4.6 para calcular a combinação de hipóteses originadas dos detectores de temperatura, fumaça e chama. Vale observar que quando três ou mais detectores contribuem com informações, a aplicação da regra de Dempster é repetida usando os elementos calculados a partir da primeira aplicação da regra (por exemplo, a combinação de hc_1 com hc_2) com a hipótese (hc_3) do próximo detector. Isto significa que a regra de combinação de Dempster pode ser generalizada da Equação 4.5 para mais de duas hipóteses, como apresentado na Equação 5.1, e a ordem das várias combinações não afeta o resultado.

$$m_1 \oplus m_2 \oplus m_3 \oplus ... = (((m_1 \oplus m_2) \oplus m_3) \oplus ...) \tag{5.1}$$

O CombinedHypEPA utilizado para combinar e processar os três eventos (*TemperatureHypothesisConjectureEvent*, *SmokeHypothesisConjectureEvent*, *FlameHypothesisConjectureEvent*) ocorridos é apresentado no Código 5.4.

Code 5.4: CombinedHypEPA.

```
1 insert into PlausibleHypothesisConjectureEvent
2 select Temp.epl as tpl, Smoke.epl as spl, Flame.epl as fpl, Temp.ts as ts,
3         lsdi.ufma.br.dstcep.dstheory.DempsterRule.combination(
4         {Temp.h, Smoke.h, Flame.h}) as pls
5 from pattern [(every (Temp=TemperatureHypothesisConjectureEvent()
6         and Smoke=SmokeHypothesisConjectureEvent()
7         and Flame=FlameHypothesisConjectureEvent()))
8 where timer:within(1 minutes)]
```

Na linha 2, os valores das evidências são capturados a partir dos fluxos de conjecturas de hipóteses de temperatura, fumaça e chama. A regra de combinação de Dempster é chamada nas linhas 3-4 usando as hipóteses relacionadas. O resultado da combinação é atribuído a *pls* (linha 4). As linhas 5-8 verificam se as conjecturas das hipóteses ocorreram dentro de 1 minuto.

Retomando os valores de massa a partir das três conjecturas de hipóteses:

- $m_{Flame}(yf) = 0.7072$, $m_{Flame}(nf) = 0.2978$
- $m_{Smoke}(yf) = 0.6345$, $m_{Smoke}(nf) = 0.3655$
- $m_{Temp}(yf) = 0.8135$, $m_{Temp}(nf) = 0.1865$

A regra de combinação de Dempster é calculada, a conjectura de hipóteses plausíveis pls (ver Figura 5.8) gerada pelos múltiplos detectores é dada pelas Equações 4.5 e 4.6, e os resultados da combinação são $m_{pls}(yf)$ = 0.9470 e $m_{pls}(nf)$ = 0.0530. A partir da hipótese combinada pls, é possível detectar a hipótese mais plausível, que é a de incêndio (*Fire*). O resultado final apresentado é baseado em evidências em favor da hipótese de incêndio a partir de todos os detectores, ou seja, o maior valor de massa de cada detector conjecturando a hipótese de incêndio.

Entretanto, é importante observar contradições entre hipóteses neste estudo de caso. Ou seja, o maior valor de massa a partir de um detector a favor de incêndio

e, no mesmo instante, outro detector a favor de não incêndio. A tabela 5.1 ilustra exemplos de condições reais de incêndio e não incêndio (*Fire*, *Non-fire)*, além de hipóteses conflitantes que são descritas logo abaixo.

Sources	**Conflict 1**		**Conflict 2**		**Conflict 3**		**Conflict 4**	
	Fire	**Non-fire**	**Fire**	**Non-fire**	**Fire**	**Non-fire**	**Fire**	**Non-fire**
Temperature Detector	0.7311		0.8242		0.9667		0.5776	
Smoke Detector		0.7571	0.99		0.7027		0.99	
Flame Detector		0.9031		0.8785	0.99			0.5567
Real Condition	**NoFire**		**Fire**		**Non-fire**		**Fire**	
DST-CEP		**0.9152**	**0.9848**			**0.5869**	**0.9909**	

Tabela 5.1: Results of conflicting hypotheses and *DST-CEP*.

- Conflito 1: sensores de temperatura podem estar próximos ou em contato com materiais aquecidos (por exemplo, fios elétricos, placas aquecidas), neste caso, *Temperature Detector* conjectura de modo anormal e por engano, a hipótese de incêndio ($m_{Temp}(yf)$ = 0.7311), enquanto os outros dois detectores indicam a ausência de incêndio $m_{Flame}(nf)$ = 0.9031 e $m_{Smoke}(nf)$ = 0.7571. *DST-CEP* conclui corretamente o não incêndio com $m_{DST}(nf)$ = 0.9152;

- Conflito 2: ilustra uma condição real de incêndio com aumento de temperatura ($m_{Temp}(yf)$ = 0.8242) e fumaça, pois o detector notifica incêndio $m_{Smoke}(yf)$ = 0.99. Entretanto, devido à pouca visibilidade de chama, o detector *Flame Detector* notifica erroneamente o não incêndio $m_{Flame}(nf)$ = 0.8785. Usando *DST-CEP*, o resultado final da combinação concluiu a condição real de incêndio com $m_{DST}(yf)$ = 0.9848;

- Conflito 3: ilustra uma condição totalmente errônea de detecção de chama $m_{Flame}(yf)$ = 0.99, eventualmente ocasionada por alguma interferência como flashes de luz, cigarros acesos, solda, luz solar (direta ou refletida), radiação de superfícies quentes, etc. No entanto, o detector de temperatura não detecta evidências de variação brusca de temperatura $m_{Temp}(nf)$ = 0.9667 e o detector de fumaça também não detecta qualquer anormalidade $m_{Smoke}(nf)$ = 0.7027. Assim, as evidências consideradas em conjunto pelo *DST-CEP* concluem a situação de não incêndio $m_{DST}(nf)$ = 0.5869, coincidindo com a real condição do ambiente.

- Conflito 4: ilustra uma condição real de incêndio de combustão lenta, ou seja, uma alta quantidade de fumaça no início (*Smoke Detector* notifica incêndio, $m_{Smoke}(yf)$ = 0.99); mas com o aquecimento lento e a ausência de chamas, ou seja, os outros dois detectores notificam respectivamente incêndio ($m_{Temp}(yf)$ = 0.5776 e não incêndio $m_{Flame}(nf)$ = 0.5567). Nesta situação, o resultado da combinação *DST-CEP* com $m_{DST}(yf)$ = 0.9909 concluiu a real condição;

Como ilustrado, os resultados finais representam uma "síntese" de todas as evidências e conjecturas de hipóteses. Com o uso de elementos da Teoria de Dempster-Shafer para a modelagem e processamento de incertezas em fluxos de eventos, a abordagem *DST-CEP* alcança resultados positivos mesmo quando os sensores estão em situações anormais e geram hipóteses conflitantes. Ou seja, através dos resultados DST-CEP é possível identificar a hipótese mais assertiva dentro de um domínio, mesmo ao combinar hipóteses conflitantes baseadas em evidências de múltiplas fontes.

5.5 Síntese

Este capítulo apresentou o framework *DST-CEP*. Inicialmente um estudo de caso foi desenvolvido para ilustrar o uso do framework *DST-CEP* em um cenário de aplicação de detecção incêndio em tempo real baseado em múltiplos sensores. Algoritmos específicos das funções de massa foram descritos e implementados com base na investigação dos sensores utilizados. Neste capítulo a modelagem do framework foi apresentada e no estudo de caso foram apresentados a concepção do quadro de discernimento, os níveis de processamento do framework DST-CEP detalhados, a computação, os algoritmos, os cálculos realizados, e finalmente uma discussão com os resultados gerados em situações conflitantes. Além disso, o mesmo estudo de caso também é usado no capítulo seguinte de avaliação experimental com dados de sensores reais processados na abordagem *DST-CEP*.

6 Experimentos e Avaliação

Esse capítulo tem como objetivo investigar os benefícios do tratamento de incerteza com a abordagem *DST-CEP* através de experimentos e avaliações realizadas em uma aplicação de IoT. O capítulo anterior apresentou o estudo de caso do **Sistema Multisensor de Detecção de Incêndios** no domínio de uma aplicação de *Smart Home* equipada com múltiplos sensores. A partir desse estudo de caso, neste capítulo objetivou-se avaliar a solução *DST-CEP* com métricas de performance bem conhecidas, curvas ROC (*Receiver Operation Characteristic*) e AUC (*Area Under Curve*). Para tal avaliação, a solução foi submetida para o processamento de um conjunto de dados (*dataset*) coletados de sensores reais. Uma *baseline* foi apresentada para explorar a análise do processamento das regras CEP sem considerar incerteza para fins de comparação com os resultados DST-CEP que considera incerteza. Além disso, foram exploradas na *baseline* abordagens probabilísticas frente a abordagem DST-CEP.

6.1 Dataset de princípio de incêndio e modelos de sensores

Neste experimento, utilizou-se um conjunto de dados coletados por Umoh at al. [77]. Os autores forneceram seu conjunto de dados (*dataset*) com 2.100 registros a partir de três sensores: DHT11 usado como sensor de temperatura, sensor de fumaça MQ-2, e sensor de chama LM393. O experimento conduzido em [77] reproduziu condições do ambiente e de fogo. O *dataset* compreende as seguintes características: *temperature*, *smoke* e *flame*. Há também o registro de tempo (*timestamp*) TS, sendo que os detectores têm os *timestamps* sincronizados, o que significa que foram programados para realizar as leituras no mesmo instante. Além disso, o *dataset* apresenta a real situação por meio do rótulo (*Label*) chamado "fire outbreak detection" ou a base de verdade (*the ground truth*). O rótulo com valor 0 (zero) é uma detecção onde não há incêndio (*non-fire*), enquanto o valor 1 (um) é uma detecção onde existe incêndio (*fire*). Uma amostragem desse *dataset* é apresentada na Tabela 6.1.

TS	Temp	Smoke	Flame	Label
1519785480	33.566	65.797	483.05	0
1519785510	37.682	66.021	516.66	0
1519785540	45.647	79.257	920.62	1

Tabela 6.1: A sample of the fire outbreak dataset.

6.2 Métricas de Avaliação

As seguintes métricas de desempenho bem conhecidas foram analisadas [72]: *Accuracy, Precision, Recall e F-Measure*. Para este fim, os resultados foram representados usando a matriz de confusão da seguinte forma:

- *True Positive* (TP): quando ocorre uma conjectura correta de valor positivo. O valor da detecção a partir do *dataset* é sim (incêndio) e o valor da hipótese plausível da solução é também sim (incêndio);
- *True Negative* (TN): quando ocorre uma conjectura correta de valor negativo. O valor da detecção a partir do *dataset* é não-incêndio e o valor da hipótese plausível a partir da solução também é não-incêndio;
- *False Positive* (FP): quando ocorre uma conjectura errada de valor positivo. O valor da detecção a partir do *dataset* é não-incêndio e o valor da hipótese plausível a partir da solução é sim (incêndio);
- *False Negative* (FN): quando há uma conjectura errada de valor negativo. O valor da detecção a partir do *dataset* é sim (incêndio) e o valor da hipótese plausível da solução é não-incêndio.

A partir dos resultados acima, pode-se calcular *Accuracy* (Acc), *Precision* ($Prec$), *Recall* (Rec) e *F-Measure* ($F1$) [72]. As curvas ROC (*Receiver Operation Characteristic*) e AUC (*Area Under Curve*) também foram analisadas. De forma simplificada a abordagem ROC pode exibir um *trade-off* entre os resultados de sensibilidade e especificidade. As expressões matemáticas são apresentadas abaixo.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

$$F1 = \frac{2 * (Prec * Rec)}{Prec + Rec} \tag{6.2}$$

$$Prec = \frac{TP}{TP + FP} \tag{6.3}$$

$$Sensitivity = \frac{TP}{TP + FN} = Rec \tag{6.4}$$

$$Specificity = \frac{TN}{FP + TN} = 1 - \frac{FP}{FP + TN} \tag{6.5}$$

Em termos de detectores de incêndio, a Sensibilidade (Equação 6.4 - *Sensitivity*) pode ser definida como a capacidade que um detector tem de reconhecer o incêndio (*true positive rate*). Em comparação, a Especificidade (Equação 6.5 - *Specificity*) é definida como a capacidade que um detector tem de reconhecer o não-incêndio (*true negative rate*), ou a habilidade de excluir falsas notificações de incêndio ($1 - Specificity$).

6.3 Baseline

Esta seção descreve uma *baseline* para avaliar a solução *DST-CEP* sob a perspectiva das métricas de performance, inicialmente em comparação a sistemas com apenas um detector de incêndio. Além disso, esta seção descreve o processamento CEP sem considerar o tratamento de incerteza, para fins de comparação com os resultados *DST-CEP* que consideram incerteza no processamento. Finalmente, esta seção descreve abordagens probabilísticas frente à abordagem *DST-CEP*.

6.3.1 Sistema com um Detector de Incêndio

O objetivo desse experimento é avaliar o desempenho de um único detector implantado na EPN, com a finalidade de avaliar a performance de um sistema com apenas um detector de incêndio para conjecturar *fire* e *non-fire*. Por exemplo, o

detector de temperatura é considerado individualmente e sua taxa de performance é verificada. Em seguida, os valores gerados pelo detector de temperatura são registrados e avaliados em relação à solução DST-CEP. Além disso, o experimento deve mostrar resultados onde todos os detectores devem apresentar individualmente diferentes desempenhos (talvez ruins em algumas métricas), e como DST-CEP se comporta ao combinar informações conflitantes de detectores que possuem algumas métricas de performance baixas. Adicionalmente, o desempenho das curvas ROC e AUC de todos os detectores, e a abordagem *DST-CEP* foram avaliados.

6.3.2 Processamento CEP sem tratamento de incerteza

O objetivo desse experimento é avaliar os ganhos da abordagem *DST-CEP* em comparação ao processamento CEP "puro", ou seja, sem utilizar o tratamento de incerteza. Para isso, a mesma aplicação de detecção de incêndio do estudo de caso foi desenvolvida sem considerar o dado de incerteza dos sensores e sem usar a regra de combinação de Dempster. Por fim, medimos as taxas de performance das duas aplicações, primeiro com a abordagem *DST-CEP*, quando a incerteza é considerada no processamento, e segundo com o CEP puro, sem considerar o tratamento de incerteza no processamento.

O desenvolvimento da aplicação de detecção de incêndio sem considerar o tratamento de incerteza é dividido em duas etapas:

1. Os mesmos algoritmos das funções de massa (Seção 5.3.2) foram implementados para cada detector TD (TemperatureDetector), SD (SmokeDetector) e FD (FlameDetector), porém sem considerar a imprecisão dos sensores. Os eventos são disparados utilizando os mesmos limiares nas funções de massa, entretanto sem considerar as incertezas inerentes aos detectores.

2. Os resultados sem incerteza dos detectores TD, SD e FD, não foram combinados com a regra de Dempster. Para tal tarefa, duas perspectivas de combinação da Lógica Proposicional (LP) foram usadas. Os operadores de conjunção ($\wedge$) e disjunção ($\vee$) como justifica-se:

   (a) O operador de conjunção foi escolhido para criar a regra de conjunção (RC), por que neste cenário de detecção de incêndio, entende-se que a regra RC é

considerada bastante precisa. RC só deve gerar notificações de incêndio se a confirmação de incêndio ocorrer em todos os três detectores ($TD \wedge SD \wedge FD$). Ou seja, RC é verdadeira (notifica *Fire*) somente se TD, SD e FD forem verdadeiros (notificarem *Fire*). RC notifica *Non-Fire* se qualquer um dos sensores TD, SD ou FD notificarem falso, além de dois ou todos os detectores notificarem falso, portando uma regra mais "rígida".

(b) Por outro lado, a regra de disjunção (RD) pode ser considerada mais "flexível", por que as notificações de incêndio de RD não requer confirmação de incêndio em todos os três detectores, mas apenas a confirmação de incêndio em um ou dois detectores ($TD \vee SD \vee FD$). Entende-se também a regra RD como bastante "sensível", pois a mesma dispara incêndio com a notificação *Fire* de apenas um dos detectores, logo uma regra mais sensível merece ser avaliada.

Portanto, pretende-se considerar na detecção de incêndio regras mais rígidas (RC) e mais flexíveis ou sensíveis (RD), para comparar com a regra de combinação de Dempster da abordagem *DST-CEP*.

6.3.3 Modelos Probabilísticos

O objetivo desse experimento é explorar modelos probabilísticos baseados em [17], que admitem eventos caracterizados por um grau de incerteza usando a teoria da probabilidade para comparar com a abordagem DST-CEP. Para isso, duas aplicações de detecção de incêndio semelhantes ao estudo de caso foram desenvolvidas assumindo os seguintes modelos probabilísticos:

1. O modelo NPM (*Normal Probabilistic Model*) que, é dividido em duas etapas:

   (a) A primeira etapa de implementação do modelo NPM assume que os detectores possuem a função de distribuição de probabilidade (*pdf*) do erro conhecida e representada por uma distribuição Normal. Nesta aplicação, os valores de distribuições do erro dos detectores são os mesmos utilizados nos detectores na abordagem *DST-CEP*. Ou seja, os seguintes valores para os detectores apresentados na Seção 5.3.2: detector de fumaça (0.03), detector de chama (45.0) e detector de temperatura (2.0).

(b) Na segunda etapa da implementação do modelo NPM, os resultados dos detectores com os erros considerados devem ser processados. Para isso, a aplicação segue a definição em [17], indicando que as leituras dos detectores que vêm de fontes diferentes e independentes possuem a probabilidade geral do evento derivado como o produto das probabilidades dos eventos primitivos. Nesta situação, tem-se o resultado da probabilidade do evento derivado (ou composto). Ao final, os resultados foram observados e comparados com a abordagem *DST-CEP*.

2. O modelo IPM (*Improved Probabilistic Model*) que, é dividido em duas etapas:

(a) A primeira etapa de implementação do modelo IPM segue uma versão melhorada no NPM, visto que os erros de medição dos detectores são conhecidos e têm uma distribuição normal N(0, 1) para todos os detectores. Entende-se este modelo como melhorado, por que um menor desvio padrão corresponde a um detector mais preciso, e neste caso, tem-se a maioria dos detectores com os desvios reduzidos: detector de fumaça N(0, 1), detector de chama N(0, 1) e detector de temperatura N(0, 1).

(b) Na segunda etapa da implementação do modelo IPM, observou-se que o desempenho do IPM é melhorado ao substituir o produto dos resultados dos detectores individuais pela média ponderada desses resultados, alcançando melhor performance.

Portanto, o desempenho da solução *DST-CEP* foi comparado com os resultados de diferentes fontes de informação, que são apresentados na Tabela 6.2.

6.3.4 Resultados e Análises

Quanto ao experimento com a proposta de **Sistemas com um Detector de Incêndio**, a Tabela 6.3 apresenta os resultados das métricas de performance dos experimentos realizados. Ao analisar os detectores individualmente, o detector de fumaça apresenta o melhor desempenho em *Accuracy, Recall, e F-Measure* mas baixo valor de *Precision* (75,82%). O detector de temperatura também apresenta baixa *Precision* (50,17%) e as outras métricas de performance abaixo do detector de fumaça, destacando apenas o *Recall* (77.84%) como melhor resultado. A razão da baixa precisão

Sources	Description
Temperature Detector (TD)	Um único detector implantado na EPN, com a finalidade de avaliar a performance de um sistema com apenas um detector de temperatura para conjecturar *fire* e *non-fire*.
Smoke Detector (SD)	Um único detector implantado na EPN, com a finalidade de avaliar a performance de um sistema com apenas um detector de fumaça para conjecturar *fire* e *non-fire*.
Flame Detector (FD)	Um único detector implantado na EPN, com a finalidade de avaliar a performance de um sistema com apenas um detector de chama para conjecturar *fire* e *non-fire*.
Rule of Conjunction (RC)	Performance da regra de conjunção sem considerar o tratamento de incerteza nas fontes do processamento CEP para conjecturar *fire* e *non-fire*.
Rule of Disjunction (RD)	Performance da regra de disjunção sem considerar o tratamento de incerteza nas fontes do processamento CEP para conjecturar *fire* e *non-fire*.
Normal Probabilistic Model (NPM)	Performance do modelo probabilístico onde cada notificação de evento na fonte é acompanhada por um valor de probabilidade. A função de distribuição de probabilidade do erro é conhecida.
Improved Probabilistic Model (IPM)	Performance do modelo probabilístico onde cada notificação de evento na fonte é acompanhada por um valor de probabilidade. A função de distribuição de probabilidade do erro é conhecida e melhorada $N(0,1)$.

Tabela 6.2: Baseline de comparação.

é por que o detector de temperatura tem o pior (ou alto) número de falsas notificações de incêndio (FP). O detector de chamas apresenta o pior número de ocorrências de incêndio não notificadas (FN), portanto a pior performance de *Recall* (59,46%). Por outro lado, o detector de fumaça não apresentou qualquer falsa notificação de incêndio (FN), portanto justificando o melhor *Recall* entre todos os detectores (100%). Embora o detector de fumaça individualmente apresente bons resultados em algumas métricas, não é sugerido considerar a utilização de maneira única deste sensor para a detecção de incêndio, por que neste caso, se o mesmo falhar, tem-se um problema diretamente relacionado com a confiabilidade do sistema. Entretanto, *DST-CEP* permite um grau de confiabilidade maior, já que gera um resultado mesmo em caso de falha de um dos detectores.

Sources	**Performance Metrics**			
	Accuracy	**Precision**	**Recall**	**F-Measure**
Temperature Detector	73.71%	50.17%	77.84%	61.02%
Smoke Detector	91.57%	75.82%	100.0%	86.25%
Flame Detector	86.71%	85.94%	59.46%	70.29%
DST-CEP	91.70%	80.58%	90.22%	85.13%

Tabela 6.3: Resultados das métricas de performance dos detectores.

Neste cenário de detecção de incêndio foi considerada a quantidade de falsos negativos (FN) um fator importante porque é fundamentalmente crítico negligenciar ocorrências reais de incêndio. Sob essa perspectiva, *DST-CEP* tem o número de FNs efetivamente reduzido (não perfeito). Este fato é justificado pelo *Recall* de 90.22% alcançado pela abordagem *DST-CEP*, o que demonstra sua boa capacidade em detectar corretamente os eventos de incêndio ocorridos. Vale observar que o *Recall* representa a sensibilidade da solução *DST-CEP*, onde a taxa de verdadeiros positivos não é ignorada. Além disso, os críticos falsos negativos são reduzidos em relação às ocorrências reais de incêndio. Para avaliar a performance geral da abordagem *DST-CEP* no sistema de detecção de incêndio, foram analisadas as métricas gerais de *Accuracy*, ou seja, dentre todas as classificações quantas foram corretas, e *F-Measure*, que apresenta a média harmônica entre *Prec.* e *Rec.* *DST-CEP* atinge *Accuracy* de 91.70% e *F-Measure* de 85.13%, o que demonstra que detecta corretamente grande parte dos

eventos de incêndio e não-incêndio com menos dificuldade. O valor de 80,58% em *Precision* significa uma boa detecção de verdadeiros positivos (VP).

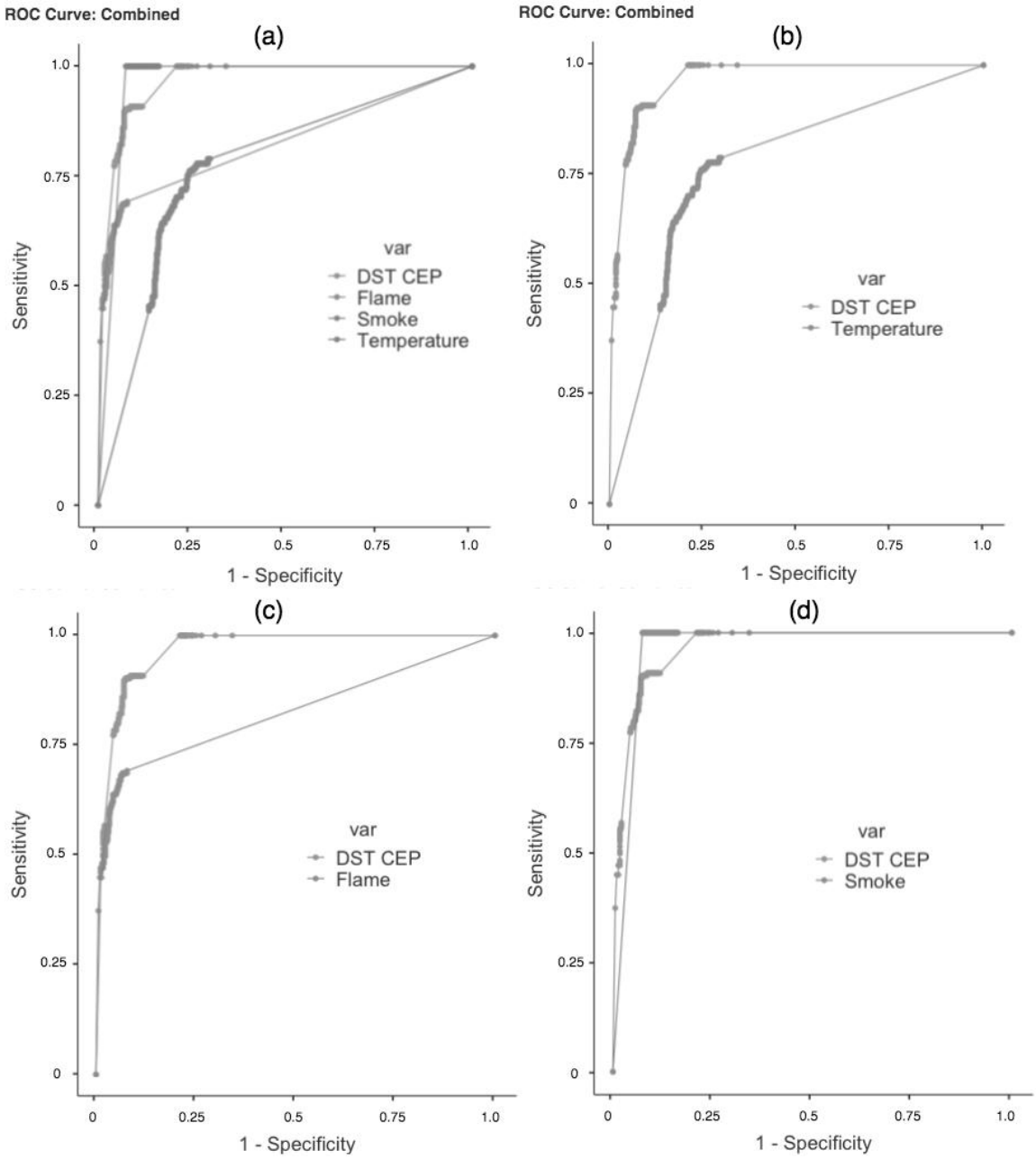


Figura 6.1: Curvas ROC de detectores e *DST-CEP*.

Na prática, é desejável ter um resultado que seja tanto sensível quanto específico. No Gráfico (a) (Figura 6.1), as curvas ROC apresentadas se referem aos detectores de incêndio e *DST-CEP*. As curvas mais sensíveis e específicas estão mais próximas do canto superior esquerdo, e neste caso a pior curva apresentada na Figura 6.1 (a) é a do detector de temperatura (linha rosa). Os Gráficos (b) e (c) (Figura 6.1) demonstram um maior poder de discriminação da relação entre sensibilidade e especificidade sendo que, o melhor resultado é apresentado na curva

DST-CEP (linha verde). Na Figura 6.1 (d), quando a curva *DST-CEP* (linha verde) cruza a curva (linha laranja) do detector de fumaça (*smoke*), mais atenção é requerida porque não existe uma relação dominante entre as curvas sobrepostas. A medida de AUC é apresentada como um resumo das curvas ROC e pode lidar com a situação de sobreposição de curvas [90]. Neste caso, como é observado na Tabela 6.4, a medida de AUC do *DST-CEP* mostra um melhor desempenho superando todos os resultados dos detectores individuais. Uma medida de AUC alta significa que a detecção de incêndio tem baixas taxas de falsos positivos (alta especificidade) quando a sensibilidade também é alta, o que é um comportamento desejado para a solução *DST-CEP*.

Area Under Curve			
DST-CEP	SD	TD	FD
0.9661	0.9631	0.7690	0.8229

Tabela 6.4: Resultado das medidas de AUC

Quanto ao experimento de **Processamento CEP sem tratamento de incerteza** em relação à abordagem *DST-CEP*, a Tabela 6.5 mostra os resultados dos experimentos. Ao analisar as métricas de performance dos detectores sem considerar incerteza no processamento, a regra RC é a fonte mais precisa por que gera a notificação de incêndio requerendo a confirmação de incêndio em todos os três detectores. Por este motivo, a RC é capaz de reconhecer as ocorrências de incêndio com poucos falsos positivo, conseguindo assim uma precisão de 93.55%, superando *DST-CEP*. Entretanto, RC tem a métrica de *Recall* baixa (47,28%) e os valores de *Accuracy* e *F-Measure* abaixo da solução *DST-CEP*.

Sources	**Performance Metrics**			
	Accuracy	**Precision**	**Recall**	**F-Measure**
Rule of Conjunction	85.26%	93.55%	47.28%	62.82%
Rule of Disjunction	75.97%	52.27%	100.0%	68.66%
DST-CEP	91.70%	80.58%	90.22%	85.13%

Tabela 6.5: Resultados das métricas de performance da regras RC e RD sem incerteza.

Como ilustra a Tabela 6.5, já a regra RD pode ser considerada mais sensível, por que as notificações de incêndio não requerem confirmação de incêndio em todos os

três detectores, mas apenas a confirmação de incêndio de um ou dois detectores. Por este motivo, RD detecta notificações de incêndio sem nenhum falso negativo (*Recall* de 100%). Entretanto, o RD gera muitas notificações de falso positivo, assim ocasionando baixa precisão (52.27%). A solução *DST-CEP* supera RD nas métricas de *Precision*, *Accuracy*, e *F-Measure*.

Quanto aos experimentos dos **Modelos Probabilísticos** em comparação à abordagem *DST-CEP*, a Tabela 6.6 mostra os resultados do experimento. Ao analisar as métricas de performance dos modelos probabilísticos, *DST-CEP* supera os resultados desses modelos. O melhor resultado do NPM é *Accuracy* com 61,14%. Observou-se que o desempenho do IPM poderia ser melhorado ao substituir o produto dos resultados dos detectores individuais pela média ponderada dos resultados. Dessa forma, foi observada uma melhora de desempenho do IPM em comparação ao NMP, como ilustra a Tabela 6.6. Mesmo com esta melhora, *DST-CEP* supera o IPM na maior parte das métricas de performance.

Sources	**Performance Metrics**			
	Accuracy	**Precision**	**Recall**	**F-Measure**
NPM	61.14%	32.94%	45.41%	38.18%
IPM	91.29%	79.52%	90.27%	84.56%
DST-CEP	91.70%	80.58%	90.22%	85.13%

Tabela 6.6: Resultados das métricas de performance dos modelos probabilísticos.

6.4 Discussão

A avaliação experimental realizada permitiu analisar a solução *DST-CEP* considerando métricas de performance em uma aplicação de detecção de incêndio em tempo real. Os resultados confirmaram os ganhos da abordagem *DST-CEP*. Ao analisar os experimentos com propostas de sistemas com um único detector, em geral, todos eles têm baixa performance individualmente. Observou-se que a maioria das métricas dos detectores estão na faixa de 70%, algumas até próximas de 50%. Entretanto, a solução *DST-CEP*, ao realizar a regra de combinação de Dempster, alcança bons resultados em todas as métricas de performance. Além disso, *DST-CEP* alcançou

a melhor curva ROC mesmo quando combina informações de detectores com baixa performance, causando notificações de incêndio conflitantes. Nesta situação, a curva ROC da solução *DST-CEP* obteve melhor sensibilidade e especificidade. Os resultados da área sob a curva (AUC) confirmaram a maior medida para a abordagem *DST-CEP*.

Adicionalmente, *DST-CEP* demonstrou significativas melhorias nos resultados de performance em relação ao processamento CEP sem tratamento de incerteza, ou seja, as regras RC e RD atenderam exaustivas 16 possibilidades de combinação dos resultados dos detectores que não consideram a imprecisão. A regra RC teve melhor *Precision*, mas teve resultados de *Recall*, *Accuracy*, e *F-Measure* abaixo da solução *DST-CEP*. Já RD é uma regra mais sensível às notificações de incêndio e teve o melhor *Recall*. No entanto, RD gerou muitas notificações falso positivas, portanto *DST-CEP* obteve os melhores resultados em *Accuracy*, *Precision* e *F-Measure*. Ambos os modelos probabilísticos NPM e IPM também foram superados pela solução DST-CEP em relação à maior parte das métricas de performance. Em geral, todos os experimentos realizados na *baseline* proposta confirmaram o resultado promissor da abordagem *DST-CEP*.

6.4.1 Limitações Identificadas

A partir do desenvolvimento da solução *DST-CEP* e da avaliação experimental, pudemos identificar algumas limitações. Uma dificuldade com a abordagem *DST-CEP* é a modelagem das funções de massa. Vale observar que a Teoria Dempster-Shafer não se concentra no julgamento, cálculo ou mecanismo pelo qual o valor da massa é determinado. Esta teoria se concentra na combinação de valores de massa distintos baseados em evidências de múltiplas fontes. Portanto, a modelagem de funções de massa pode ser complexa, dependendo do domínio do problema ou do cenário da aplicação.

Outra dificuldade é a falta de consenso na seleção de um limiar ótimo para que um detector gere uma notificação. Existe pouca orientação sobre a seleção do limiar ideal para que detectores sejam obrigados a gerar um alarme. Especificamente no estudo de caso apresentado, em sistemas de detecção de incêndio, a natureza do fogo é uma ciência à parte. Por exemplo, muitas variáveis, tais como tipo de combustível, projeto do detector, modo de combustão e estágios do incêndio ou

fumaça, podem afetar a resposta de um detector de incêndio [29]. Buscou-se limiares a partir das informações dos fabricantes dos sensores e os fenômenos observados.

7 Conclusões

Este documento apresentou uma solução para o tratamento do problema de incerteza no processamento de eventos em aplicações de IoT baseadas em CEP. A incerteza nessas aplicações usualmente é observada no processamento dos eventos primitivos (ex., leituras de sensores) e na sua propagação para os eventos complexos derivados (ex., situações de alto nível). Nesse sentido, foi apresentada a proposta *DST-CEP*, que é uma abordagem que utiliza a Teoria Dempster-Shafer (TDS) para tratar incertezas. Por meio do uso da TDS, a abordagem *DST-CEP* realiza combinação de dados de sensores não confiáveis em situações conflitantes e detecta os resultados corretamente em sua maioria. Para alcançar tal objetivo, a abordagem propõe um modelo arquitetural *DST-CEP* dividido em níveis de processamento para lidar com a incerteza nos eventos e sua propagação para os eventos complexos. Em suma, no primeiro nível de sensores foi elaborado um mecanismo de representação formal das leituras dos sensores e informações de incerteza, levando-as em consideração explicitamente nos eventos primitivos (ou eventos de evidências no modelo). No nível de hipóteses, as notificações dos eventos de evidência permitiram derivar eventos de conjecturas de hipóteses através das funções de massa. Em *DST-CEP*, para atender as funções de massa foram elaborados algoritmos considerando intervalos de incerteza dos detectores. No último nível de combinação de hipóteses, a partir de um conjunto de hipóteses derivadas é, utilizada a regra de combinação de Dempster para calcular a hipótese mais plausível.

A concretização do modelo arquitetural se deu através da implementação do framework *DST-CEP*. Este framework forneceu uma solução, através de um conjunto de classes e interfaces que facilita a construção de aplicações de IoT que lidam com o problema de incerteza em processamento de eventos. Para ilustrar o uso do framework, um estudo de caso foi elaborado descrevendo uma aplicação de IoT que detecta princípio de incêndio em tempo real baseada em dados de múltiplos sensores não confiáveis. Os algoritmos de funções de massa desta aplicação foram elaborados baseados em requisitos de normas reguladoras de sistemas de detecção de incêndio, além de investigação da mecânica de funcionamento dos sensores e

estudo dos fenômenos capturados por cada um deles. Esta aplicação de detecção de incêndio foi desenvolvida utilizando o framework *DST-CEP* observando as etapas de modelagem, implementação e níveis de processamento da aplicação e do framework *DST-CEP*, com ênfase em uma apresentação detalhada de todas as etapas para orientar o leitor. A partir da implementação do estudo de caso, a solução DST-CEP foi submetida para o processamento de um *dataset* coletado de sensores reais e posterior avaliação. Para comparar a solução *DST-CEP* com outras soluções e abordagens, uma *baseline* de experimentos foi elaborada e incluiu: sistemas de detecção de incêndio com apenas um detector implantado na EPN, sistema de processamento CEP que não considera o tratamento de incerteza e finalmente abordagens probabilísticas, baseadas em trabalhos relacionados. A avaliação se deu observando métricas de performance bem conhecidas (*Accuracy, Precision, Recall e F-Measure*), além de curvas ROC (*Receiver Operation Characteristic*) e AUC (*Area Under Curve*).

Ao retomar a **hipótese de pesquisa** deste trabalho: *A Teoria de Dempster-Shafer possibilita desenvolver uma abordagem para adequadamente modelar e tratar os problemas de incerteza originados de dados de sensores não confiáveis e, especificamente para aplicações de IoT baseadas em processamento de eventos, lidar com a incerteza na origem dos eventos e sua propagação para os eventos derivados.*, conclui-se que essa hipótese de pesquisa foi materializada através da arquitetura e framework propostos, e validada através do estudo de caso e seus experimentos de avaliação. Portanto, considera-se que o resultado deste trabalho foi satisfatório e devidamente validado neste processo de investigação científica.

7.1 Contribuições

Os problemas de incertezas levantados nesta pesquisa foram tratados pela abordagem *DST-CEP*. Logo, verificou-se as principais contribuições deste trabalho, como se segue:

- Um mapa geral de tópicos representando uma ampla visão das principais pesquisas, abordagens e técnicas para realizar o tratamento de incertezas em CEP. De maneira particular, este mapa de tópicos permitiu organizar o conhecimento sobre o tema desta pesquisa. Além disso, possibilitou estruturar informações

complexas sobre o problema desta pesquisa com o objetivo de relacionar e mesclar tópicos relevantes e adjacentes a partir de várias fontes como *surveys* e trabalhos relacionados;

- Um modelo arquitetural *DST-CEP* incluindo componentes da Teoria de Dempster-Shafer em um contexto de processamento de eventos originados de uma coleção de sensores. Tal modelo arquitetural é dividido em níveis de processamento para lidar com a incerteza nos eventos e sua propagação para os eventos complexos. Neste modelo, foi apresentado um mecanismo de representação formal das leituras dos sensores e informações de incerteza, levando-as em consideração explicitamente nos eventos primitivos denominados de eventos de evidências. Além disso, foi apresentado um mecanismo de representação formal dos eventos de conjecturas de hipóteses. Esta representação permite, após o processamento dos eventos de evidências, que sejam registradas as conjecturas de hipóteses de um determinado domínio, onde cada hipótese tem um valor de massa associado;

- Um framework *DST-CEP* como resultado da implementação do modelo arquitetural. Este framework fornece uma solução que permite a construção de aplicações de IoT que lidam com o problema de incerteza em processamento de eventos. Para isso, o framework *DST-CEP* fornece um conjunto de interfaces, classes extensíveis e objetos que colaboram para cumprir funcionalidades específicas de aplicações de IoT e da Teoria de Dempster-Shafer.

- Um estudo de caso detalhado que descreve um cenário de sistema de detecção de princípio de incêndio em tempo real, baseada em dados de múltiplos sensores não confiáveis. Além disso, foram elaborados algoritmos de funções de massa dos detectores de incêndio baseados em requisitos de normas reguladoras, investigação da mecânica de funcionamento dos sensores e estudos dos fenômenos capturados por cada um deles. Uma aplicação de IoT, específica do domínio de detecção de incêndio, modelada e desenvolvida utilizando o framework *DST-CEP*, que por sua vez é baseado no modelo arquitetural proposto. Uma avaliação da solução DST-CEP observando métricas de performance, curvas ROC e AUC, visando identificar corretamente as situações de incêndio na aplicação desenvolvida. Foi incluído também um comparativo

da solução *DST-CEP* com as outras abordagens e aplicações desenvolvidas no domínio da detecção de incêndio.

7.2 Trabalhos Futuros

Os planos futuros incluem explorar outros cenários de aplicações do modelo arquitetural *DST-CEP* proposto. Além de modelar a incerteza de diversos tipos de sensores, dos mais simples até os mais complexos, e inseridos no contexto de IoT nos quais a abordagem *DST-CEP* pode ser aplicada. Pretende-se também explorar outros elementos da TDS, tais como função de crença, função de plausabilidade, intervalo de crença, em domínios com muitas hipóteses.

Adicionalmente, um outro trabalho futuro inclui integrar as funções da solução *DST-CEP* diretamente à *engine* CEP. Estas funções podem aparecer em qualquer expressão ou regras EPL, e pode-se passar parâmetros específicos para efetivar seu cálculo. Tais funções podem ter seus nomes registrados diretamente na *engine*, através de arquivos ou API de configuração da *engine* CEP.

Uma outra possibilidade de trabalho futuro é a generalização da TDS com sistemas baseados em regras Fuzzy tipo 2 intervalares, ou seja, investir na pesquisa de uma abordagem TDS-Fuzzy2. Sabe-se que conjuntos Fuzzy do tipo 2 são conjuntos Fuzzy cujos graus de pertinência são conjuntos Fuzzy do tipo 1 e não um único valor. Fuzzy tipo 2 pode ser usado em situações onde existe incerteza a respeito dos graus de pertinência e incertezas nos parâmetros ou formatos das funções de pertinência. Assim, Fuzzy do tipo 2 consegue superar problemas da lógica Fuzzy tipo 1. Portanto, cabe investigar a generalização e a relação de compatibilidade entre a Teoria de Dempster-Shafer com sistemas baseados em regras Fuzzy tipo 2 intervalares, e como isso poderia contribuir para lidar com diferentes tipos de informações incertas em aplicações de IoT. Ou seja, a abordagem TDS-Fuzzy2 evidencia ser um caminho promissor de pesquisa.

7.3 Publicação Relacionada com a Pesquisa

O artigo publicado possui classificação A1 em Engenharias IV conforme estrato Qualis CAPES (2013-2016).

1. Bezerra, Eduardo D. C.; Teles, Ariel S.; Coutinho, Luciano R.; da Silva e Silva, Francisco José. 2021. “Dempster–Shafer Theory for Modeling and Treating Uncertainty in IoT Applications Based on Complex Event Processing”, Sensors 21, no. 5: 1863. https://doi.org/10.3390/s21051863

Referências Bibliográficas

[1] F. A. T. Abad, M. Caccamo, and B. Robbins. A fault resilient architecture for distributed cyber-physical systems. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on*, pages 222–231. IEEE, 2012.

[2] J. Agrawal, Y. Diao, D. Gyllstrom, and N. Immerman. Efficient pattern matching over event streams. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 147–160. ACM, 2008.

[3] V. Akila, V. Govindasamy, and S. Sandosh. Complex event processing over uncertain events: Techniques, challenges, and future directions. In *Computation of Power, Energy Information and Commuincation (ICCPEIC), 2016 International Conference on*, pages 204–221. IEEE, 2016.

[4] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V. Subrahmanian, P. Turaga, and O. Udrea. A constrained probabilistic petri net framework for human activity detection in video. *IEEE Transactions on Multimedia*, 10(8):1429–1443, 2008.

[5] E. Alevizos, A. Skarlatidis, A. Artikis, and G. Paliouras. Complex event recognition under uncertainty: A short survey. *Event Processing, Forecasting and Decision-Making in the Big Data Era (EPForDM)*, pages 97–103, 2015.

[6] E. Alevizos, A. Skarlatidis, A. Artikis, and G. Paliouras. Probabilistic complex event recognition: A survey. *ACM Computing Surveys (CSUR)*, 50(5):71, 2017.

[7] D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic. Stream Reasoning and Complex Event Processing in ETALIS. *Semantic Web*, 3(4):397–407, Oct. 2012.

[8] A. Artikis, O. Etzion, Z. Feldman, and F. Fournier. Event processing under uncertainty. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, pages 32–43. ACM, 2012.

[9] A. Artikis, M. Sergot, and G. Paliouras. A logic programming approach to activity recognition. In *Proceedings of the 2nd ACM international workshop on Events in multimedia*, pages 3–8. ACM, 2010.

[10] A. Artikis, A. Skarlatidis, F. Portet, and G. Paliouras. Logic-based event recognition. *The Knowledge Engineering Review*, 27(4):469–506, 2012.

[11] M. Babar and F. Arif. Smart urban planning using big data analytics to contend with the interoperability in internet of things. *Future Generation Computer Systems*, 77:65–76, 2017.

[12] G. Baptista. Processamento Distribuído de Eventos Complexos (CEP). Technical report, PUC-Rio, Departamento de Informática, 2011.

[13] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to probability*, volume 1. Athena Scientific Belmont, MA, 2002.

[14] P. L. Bogler. Shafer-dempster reasoning with applications to multisensor target identification systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(6):968–977, 1987.

[15] M. K. Chandy, O. Etzion, and R. von Ammon. 10201 executive summary and manifesto - event processing. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

[16] X. Chuanfei, L. Shukuan, W. Lei, and Q. Jianzhong. Complex event detection in probabilistic stream. In *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pages 361–363. IEEE, 2010.

[17] G. Cugola, A. Margara, M. Matteucci, and G. Tamburrelli. Introducing uncertainty in complex event processing: model, implementation, and validation. *Computing*, 97(2):103–144, 2015.

[18] B. Das. Representing uncertainties using bayesian networks. Technical report, ELECTRONICS RESEARCH LAB SALISBURY (AUSTRALIA), 1999.

[19] A. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, pages 325–339, 1967.

[20] A. P. Dempster et al. Upper and lower probability inferences for families of hypotheses with monotone density ratios. *The Annals of Mathematical Statistics*, 40(3):953–969, 1969.

[21] J. L. Devore. *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 2015.

[22] N. Díaz-Rodríguez, O. Cadahía, M. Cuéllar, J. Lilius, and M. Calvo-Flores. Handling real-world context awareness, uncertainty and vagueness in real-time human activity tracking and recognition with a fuzzy ontology-based hybrid method. *Sensors*, 14(10):18131–18171, 2014.

[23] M. Endler, E. H. Haeusler, V. P. de Almeira, and F. J. da Silva e Silva. Towards Real-time Semantic Reasoning for the Internet of Things. In *Proceedings of the First International Workshop on Semantic Multimedia Computing (SMC 2017)*, San Diego, California, USA, 2017.

[24] EsperTech. Esper - Complex Event Processing, 2017.

[25] O. Etzion, P. Niblett, and D. C. Luckham. *Event processing in action*. Manning Greenwich, 2011.

[26] I. Flouris, N. Giatrakos, A. Deligiannakis, M. Garofalakis, M. Kamp, and M. Mock. Issues in complex event processing: Status and prospects in the Big Data era. *Journal of Systems and Software*, 127, 2016.

[27] I. Flouris, N. Giatrakos, A. Deligiannakis, M. Garofalakis, M. Kamp, and M. Mock. Issues in complex event processing: Status and prospects in the big data era. *Journal of Systems and Software*, 127:217–236, 2017.

[28] L. M. Garshol. Metadata? thesauri? taxonomies? topic maps! making sense of it all. *Journal of information science*, 30(4):378–391, 2004.

[29] J. Geiman and D. T. Gottuk. Alarm thresholds for smoke detector modeling. *Fire Safety Science*, 7:197–208, 2003.

[30] A. M. Gianpaolo Cugola. Processing Flows of Information: From Data Stream to Complex Event Processing. *ACM Computing Surveys*, 44(3), June 2012.

[31] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Gener. Comput. Syst.*, 29(7):1645–1660, Sept. 2013.

[32] J. C. Helton. Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. *Journal of Statistical Computation and Simulation*, 57(1-4):3–76, 1997.

[33] S. Højsgaard, D. Edwards, and S. Lauritzen. *Graphical models with R*. Springer Science & Business Media, 2012.

[34] A. Iorshase and S. F. Caleb. A neural based experimental fire-outbreak detection system for urban centres. *Journal of Software Engineering and Applications*, 9(3):71–79, 2016.

[35] A. Jarraya, N. Ramoly, A. Bouzeghoub, K. Arour, A. Borgi, and B. Finance. A fuzzy semantic cep model for situation identification in smart homes. In *ECAI 2016: 22nd European Conference on Artificial Intelligence*, volume 285, pages 1678–1679. IOS Press, 2016.

[36] H. Kawashima, H. Kitagawa, and X. Li. Complex event processing over uncertain data streams. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on*, pages 521–526. IEEE, 2010.

[37] N. Khoussainova, M. Balazinska, and D. Suciu. Peex: Extracting probabilistic events from rfid data. In *In ICDE*, 2008.

[38] G. Lavee, M. Rudzsky, and E. Rivlin. Propagating certainty in petri nets for activity recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):326–337, 2013.

[39] J. Letchner and M. Balazinska. Lineage for markovian stream event queries. In *Proceedings of the 10th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 26–33. ACM, 2011.

[40] D. M. Levine, M. L. Berenson, and D. Stephan. *Estatística: teoria e aplicações-usando Microsoft Excel português*. Ltc, 2005.

[41] Z. Li, T. Ge, and C. X. Chen. ε-matching: Event processing over noisy sequences in real time. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 601–612. ACM, 2013.

[42] J. Liu, J. Yang, J. Wang, and H. Sii. Review of uncertainty reasoning approaches as guidance for maritime and offshore safety-based assessment. In *Safety and Reliability*, volume 23, pages 63–80. Taylor & Francis, 2002.

[43] W. Liu, J. Hong, M. F. McTear, and J. G. Hughes. An extended framework for evidential reasoning systems. *International journal of pattern recognition and artificial intelligence*, 7(03):441–457, 1993.

[44] J. D. Lowrance, T. D. Garvey, and T. M. Strat. A framework for evidential-reasoning systems. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 419–434. Springer, 2008.

[45] D. Luckham and R. Schulte. Event Processing Glossary - Version 2.0. Technical Report July, 2011.

[46] D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[47] J. Ma, W. Liu, and P. Miller. Event modelling and reasoning with uncertain information for distributed sensor networks. In *International Conference on Scalable Uncertainty Management*, pages 236–249. Springer, 2010.

[48] J. Ma, W. Liu, and P. Miller. An evidential improvement for gender profiling. In *Belief functions: theory and applications*, pages 29–36. Springer, 2012.

[49] J. Ma, W. Liu, P. Miller, and W. Yan. Event composition with imperfect information for bus surveillance. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 382–387. IEEE, 2009.

[50] J. Ma, W. Liu, P. Miller, and H. Zhou. An evidential fusion approach for gender profiling. *Information Sciences*, 333:10–20, 2016.

[51] P. R. Maciel, R. D. Lins, and P. R. Cunha. *Introdução às redes de Petri e aplicações*. UNICAMP-Instituto de Computacao, 1996.

[52] P. S. Mann. *Introductory statistics*. John Wiley & Sons, 2007.

[53] V. Mihajlovic and M. Petkovic. Dynamic bayesian networks: A state of the art. *University of Twente Document Repository*, 2001.

[54] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[55] A. d. P. M. Moreira et al. Aplicações da teoria da decisão e probabilidade subjetiva em sala de aula do ensino médio. 2015.

[56] N. Moreno, M. F. Bertoa, L. Burgueño, and A. Vallecillo. Managing measurement and occurrence uncertainty in complex event processing systems. *IEEE Access*, 7:88026–88048, 2019.

[57] R. Neut. Uncertainty analysis in bayesian networks. Master's thesis, 2014.

[58] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[59] S. Pepper et al. The tao of topic maps. In *Proceedings of XML Europe*, volume 3, page 77, 2000.

[60] J.-P. Poli and L. Boudet. A fuzzy expert system architecture for data and event stream processing. *Fuzzy Sets and Systems*, 343:20–34, 2018.

[61] C. Ré, J. Letchner, M. Balazinksa, and D. Suciu. Event queries on correlated probabilistic streams. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 715–728. ACM, 2008.

[62] R. Rincé, R. Kervarc, and P. Leray. Complex event processing under uncertainty using markov chains, constraints, and sampling. In *International Joint Conference on Rules and Reasoning*, pages 147–163. Springer, 2018.

[63] M. Roriz Junior. *DG2CEP : An On-line Algorithm for Real-time Detection of Spatial Clusters from Large Data Streams through Complex Event Processing Marcos Paulino Roriz Junior DG2CEP : An On-line Algorithm for Real-time Detection of Spatial Clusters from Large Data Stream*. PhD thesis, 2017.

[64] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[65] L. J. Savage. *The foundations of statistics*. Courier Corporation, 1972.

[66] N. P. Schultz-Møller, M. Migliavacca, and P. Pietzuch. Distributed complex event processing with query rewriting. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, page 4. ACM, 2009.

[67] K. Sentz, S. Ferson, et al. *Combination of evidence in Dempster-Shafer theory*, volume 4015. Citeseer, 2002.

[68] G. Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

[69] G. Shafer. What is probability. *Perspectives in Contemporary Statistics*, pages 19–39, 1992.

[70] Z. Shen, H. Kawashima, and H. Kitagawa. Probabilistic event stream processing with lineage. In *Proc. of Data Engineering Workshop*, 2008.

[71] P. Smets. Non-standard logics for automated reasoning. 1988.

[72] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[73] W. N. Stephens. Hypotheses and evidence. 1968.

[74] D. Stowell and M. D. Plumbley. Segregating event streams and noise with a markov renewal process model. *The Journal of Machine Learning Research*, 14(1):2213–2238, 2013.

[75] K. Teymourian. *A Framework for Knowledge-Based Complex Event Processing*. PhD thesis, Freie Universität Berlin, 2014.

[76] J. Q. Uchoa, S. M. Panotim, and M. d. C. Nicoletti. Elementos da teoria da evidência de dempster-shafer. *Tutorial do Departamento de Computação da Universidade Federal de São Carlos*, 1995.

[77] U. Umoh, E. Udo, and N. Emmanuel. Support vector machine-based fire outbreak detection system. *International Journal on Soft Computing, Artificial Intelligence and Applications*, 8(2), 2019.

[78] L. C. Van Der Gaag. Bayesian belief networks: odds and ends. *The Computer Journal*, 39(2):97–113, 1996.

[79] Y. Wang, K. Cao, and X. Zhang. Complex event processing over distributed probabilistic event streams. *Computers & Mathematics with Applications*, 66(10):1808–1821, 2013.

[80] Y. Wang, X. Li, X. Li, and Y. Wang. A survey of queries over uncertain data. *Knowledge and information systems*, 37(3):485–530, 2013.

[81] S. Wasserkrug, A. Gal, and O. Etzion. A model for reasoning with uncertain rules in event composition systems. *In: Proceedings of the 21st annual conference on uncertainty in artificial intelligence, pp 599–606*, 2005.

[82] S. Wasserkrug, A. Gal, and O. Etzion. A taxonomy and representation of sources of uncertainty in active systems. In *International Workshop on Next Generation Information Technologies and Systems*, pages 174–185. Springer, 2006.

[83] S. Wasserkrug, A. Gal, and O. Etzion. A model for reasoning with uncertain rules in event composition systems. *arXiv preprint arXiv:1207.1427*, 2012.

[84] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Complex event processing over uncertain data. In *Proceedings of the second international conference on Distributed event-based systems*, pages 253–264. ACM, 2008.

[85] S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin. Efficient processing of uncertain events in rule-based systems. *IEEE Transactions on Knowledge and Data Engineering*, 24(1):45–58, 2012.

[86] R. S. Wazlawick. *Metodologia de Pesquisa para Ciência da Computação*. Elsevier Brasil, 2 edition, 2014.

[87] F. Xiao. A novel evidence theory and fuzzy preference approach-based multi-sensor data fusion technique for fault diagnosis. *Sensors*, 17(11):2504, 2017.

[88] R. R. Yager and L. Liu, editors. *Classic Works of the Dempster-Shafer Theory of Belief Functions*. Springer Berlin Heidelberg, 2008.

[89] H. Zhang, Y. Diao, and N. Immerman. Recognizing patterns in streams with imprecise timestamps. *Information Systems*, 38(8):1187–1211, 2013.

[90] X. Zhang, X. Li, Y. Feng, and Z. Liu. The use of roc and auc in the validation of objective image fusion evaluation metrics. *Signal processing*, 115:38–48, 2015.