

Um Framework Fuzzy para Suporte à Seleção em Regime de Incerteza de Recursos na IoT

Patrick Fernandes¹, Renato Dilli², Renata Reiser², Adenauer Yamin^{1,2}

¹Mestrado em Engenharia Eletrônica e Computação
Universidade Católica de Pelotas (UCPEL) – Pelotas, RS – Brasil

²Programa de Pós Graduação em Computação
Universidade Federal de Pelotas (UFPEL) – Pelotas, RS – Brasil

patrick.fernandes@sou.ucpel.edu.br, renato.dilli@inf.ufpel.edu.br,
reiser@inf.ufpel.edu.br, adenauer.yamin@ucpel.edu.br

Abstract. *The selection of resources in the scenario of high scalability of the IoT, considering the preferences of the clients, is significantly complex. In turn, the definition of the priority of the QoS attributes associated with the services provided by the resources, constitutes in itself a challenging step in the whole process. Considering this, this work has as general objective the conception of a Framework for management of fuzzy rules, to be integrated with the EXEHDA-RR proposal, for the treatment of uncertainties in the definition of the QoS attributes. The preferences of the clients are translated in rules using Interval Fuzzy Logic Type-2. The results obtained showed promising results indicating the continuity of the research.*

Resumo. *A seleção de recursos no cenário de elevada escalabilidade da IoT, considerando as preferências dos clientes, se reveste de significativa complexidade. Por sua vez, a definição da prioridade dos atributos de QoS associados aos serviços disponibilizados pelos recursos, constitui por si só uma etapa desafiadora em todo processo. Considerando isso, este trabalho tem como objetivo geral a concepção de um Framework para gerenciamento de regras fuzzy, a ser integrado à proposta EXEHDA-RR, para o tratamento de incertezas na definição dos atributos de QoS. As preferências dos clientes são traduzidas em regras com o emprego de Lógica Fuzzy Tipo-2 Intervalar. Os resultados obtidos se mostraram promissores indicando a continuidade da pesquisa.*

1. Introdução

A previsão é que se tenha até 2030 ao redor de 125 bilhões de recursos conectados a Internet das Coisas [IHS Markit 2017]. Cada um destes recursos poderá disponibilizar mais de um serviço. Considerando isso, a seleção dos recursos cujos serviços melhor satisfaçam os requisitos de um cliente, entre os vários recursos aptos, pode tornar-se uma tarefa complexa.

Uma proposta de seleção, tendo em vista o perfil operacional da infraestrutura da IoT, deve levar em conta os requisitos funcionais e os não funcionais dos recursos. Os requisitos funcionais definem o que os serviços deverão oferecer. Já os requisitos não

funcionais, em particular, os relacionados a Qualidade de Serviço (QoS), especificam expectativas operacionais dos serviços. Como por exemplo, o “tempo de resposta”, que é o tempo necessário para um serviço responder; e/ou “disponibilidade”, que é a porcentagem de tempo que um serviço continua funcionando; e/ou quanto a sua “precisão” [Khutade and Phalnikar 2014].

Neste sentido, os requisitos funcionais são empregados para definir se um determinado recurso é apto para atender as demandas de um cliente, por sua vez, a avaliação de parâmetros não funcionais de QoS dos serviços oferecidos pelos recursos, constituem a base para selecionar entre os recursos aptos, quais são mais adequados para atender a demanda de um cliente.

Deste cenário surge a motivação central deste trabalho, pois temos a necessidade de classificar recursos para então selecionar os mais apropriados, na circunstância de elevada escalabilidade inerente à IoT e, também considerando as incertezas do cliente na especificação de suas preferências, quanto aos parâmetros de QoS que lhe são mais oportunos.

Nesta perspectiva, o objetivo geral deste trabalho é a concepção de um Framework para Criação de Regras Fuzzy, a ser integrado à proposta EXEHDA-RR ([Dilli et al. 2018]), que ofereça suporte ao tratamento de incertezas empregando para isto regras dinâmicas Fuzzy Tipo-2, contribuindo para uma seleção de recursos na IoT, que melhor atenda às demandas dos diferentes clientes, apesar de suas incertezas.

Como base do Framework proposto é adotado o uso de Lógica Fuzzy (LF) pela sua capacidade de lidar com incertezas, o que não é possível de se fazer com as lógicas clássicas. A LF procura representar o raciocínio e o conhecimento humano da forma mais realista possível facilitando a utilização de sistemas complexos [Silva et al. 2019].

Os resultados atingidos com o Processador Fuzzy empregando regras que podem ser alteradas de forma dinâmica, conforme as preferências informadas pelo cliente, se mostraram promissores quando comparados com aqueles obtidos por um classificador reconhecido pela literatura da área.

Este artigo está organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados, na Seção 3, o Framework Fuzzy proposto, na Seção 4, a avaliação do Framework proposto e por fim na Seção 5, as considerações finais.

2. Trabalhos Relacionados

Foi realizada uma revisão bibliográfica, sendo discutidos nesta Seção somente os trabalhos que se relacionam de forma mais direta ao Framework Fuzzy proposto. Esta revisão foi central para as discussões referentes às funcionalidades a serem propostas para o Framework concebido.

No trabalho [Alhadithy and Al-Shargabi 2018] é proposto um modelo para descoberta e seleção de serviços em nuvem utilizando regras Fuzzy, onde a seleção de serviços é baseada em QoS, com o intuito de atender aos requisitos dos usuários, bem como suas restrições. O modelo possui um módulo responsável por monitorar a execução de um serviço em particular, ou de uma composição de serviços. Caso um serviço selecionado não esteja disponível, o modelo substituirá o serviço indisponível por um outro, que também corresponda aos requisitos do usuário especificados nas regras Fuzzy já geradas.

Segundo [Belouaar et al. 2018], aponta um modelo baseado em Lógica Fuzzy com o intuito de ajudar os consumidores de serviços da Web a classificar o serviço mais adequado, tendo em conta suas preferências. É aplicada a Lógica Fuzzy para representar dados imprecisos ou vagos. Aponta o uso de funções de pertinência triangulares e o método do Centro de Gravidade para efetuar a defuzzificação. A classificação do modelo proposto em um conjunto contendo seis serviços com sete atributos de QoS é comparado com uma classificação convencional. Exibiram cinco regras que foram executadas no motor de inferência.

Conforme [Kumar et al. 2017], aborda um novo modelo para seleção de serviço de nuvem sob o ambiente Fuzzy, utilizando o processo de hierarquia analítica (AHP) e técnica fuzzy para preferência de ordem por similaridade à solução ideal (TOPSIS). O método AHP é empregado para configurar a estrutura do problema da seleção de serviço da nuvem e para estimular o peso dos critérios, com o uso de comparações e o método TOPSIS para classificação final. A proposta leva em consideração os atributos de QoS para selecionar o serviço apropriado. Explora um conjunto de variáveis linguísticas pré-definidas, para avaliar cada peso dos critérios. Embora o método Fuzzy TOPSIS seja utilizado para resolver problemas de incerteza, com dados imprecisos, esse método não pode manipular dados discretos e informações incompletas [Jatoth et al. 2019].

A Tabela 1, apresenta a comparação dos trabalhos relacionados, onde o ponto preto significa a inclusão da funcionalidade, enquanto que o ponto branco representa a exclusão e vazio, em branco, indica que não foi identificado tal função. A primeira coluna da tabela exhibe os trabalhos relacionados, o uso da Lógica Fuzzy Tipo-1, Lógica Fuzzy Tipo-2 e Preferências de QoS, respectivamente.

Tabela 1. Comparação dos Trabalhos Relacionados

Trabalhos	LF Tipo-1	LF Tipo-2	Preferências de QoS
[Alhadithy and Al-Shargabi 2018]	●	○	●
[Belouaar et al. 2018]	●	○	●
[Kumar et al. 2017]	●	○	●
Framework Proposto	○	●	●

Na revisão bibliográfica realizada, foi possível identificar que atualmente a maioria dos trabalhos em classificação e seleção de recursos, consideram as preferências de QoS do cliente, bem como implementam o tratamento da incerteza empregando Lógica Fuzzy Tipo-1.

Com o intuito de caracterizar as contribuições desta pesquisa com relação aos Trabalhos Relacionados, destacaríamos dois pontos: (i) a disponibilização de um *framework* para gerenciamento de regras Fuzzy e preferências do cliente; e, (ii) o tratamento de incertezas com o emprego da Lógica Fuzzy Tipo-2.

3. Framework Fuzzy Proposto

Esta seção apresenta o funcionamento do Framework Fuzzy proposto, bem como suas características.

O Framework permite que o cliente defina suas preferências para cada um dos atributos de QoS dos recursos. A relevância para cada um destes atributos pode ser Low

(L), Medium (M) ou High (H). E dependendo da preferência que o cliente definir, poderão ser ativadas regras adicionais na base de regras Fuzzy.

Os atributos de QoS dos recursos são normalizados e submetidos como entrada em um sistema Fuzzy Tipo-2, sendo fuzzificados por meio de funções de pertinência trapezoidais e triangulares. A utilização de funções de pertinência Fuzzy Tipo-2 na entrada de controladores Fuzzy contribui para a redução do número de regras, quando comparados com um sistema Fuzzy convencional. A arquitetura do Framework Fuzzy proposto é apresentado na Figura 1 e suas etapas são explicadas a seguir.

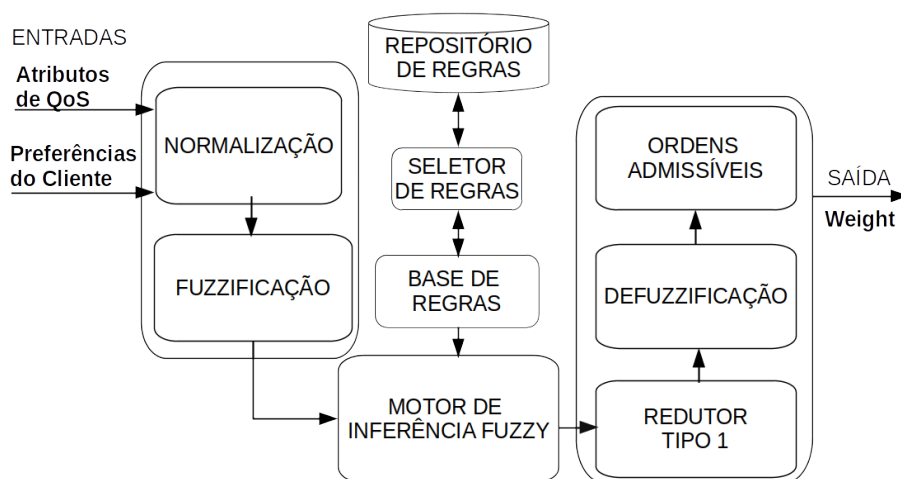


Figura 1. Arquitetura do Framework Fuzzy

3.1. Normalização e Fuzzificação

Os recursos são normalizados antes de serem processados pelo Framework Fuzzy. No processo de normalização, o melhor valor de atributo de QoS é normalizado em 100. Isto é feito para cada um dos atributos. Os atributos que já possuem sua escala de valores (0 - 100) não são normalizados, como é o caso do atributo AV. A normalização é realizada por meio da Eq.(1) se o critério deve ser maximizado ou Eq.(2) se o critério deve ser minimizado.

$$h_{i,j} = \left[\frac{v_{i,j} * 100}{max(v_j)} \right] \quad (1)$$

$$h_{i,j} = \left[\frac{min(v_j) * 100}{v_{i,j}} \right] \quad (2)$$

Após a normalização, o processo de fuzzificação transforma uma entrada crisp em um valor Fuzzy \tilde{A} indicando o grau de pertinência do elemento $x \in U$ no conjunto Fuzzy A , frequentemente dado na forma triangular ($\tilde{A} = (a, b, c)$) ou trapezoidal ($\tilde{A} = (a, b, c, d)$) [Barros and Bassanezi 2010].

Para cada atributo é adicionada uma entrada no sistema Fuzzy Tipo-2 intervalar. O sistema foi modelado com uma única saída denominada de “*Weight*”, variável linguística que representa o peso para o recurso avaliado.

A nomenclatura das variáveis linguísticas RT, AV e TH, foram definidas com base nos atributos Response Time, Availability e Throughput, respectivamente.

Na Tabela 2 são apresentados os valores dos Termos Linguísticos H, M e L, para as variáveis RT, AV e TH. Na primeira coluna estão os Termos Linguísticos. O Termo High foi modelado com função de pertinência triangular e os Termos Linguísticos Medium e Low foram modelados com função de pertinência trapezoidal. Os limites inferiores e superiores são apresentados nas colunas FP Superior e FP Inferior. Estes valores são usualmente especificados por um especialista na área. Particularmente neste artigo estes valores foram definidos a partir da revisão de literatura realizada.

Tabela 2. Framework Fuzzy - Variáveis Linguísticas Tipo-2: RT, AV e TH

Termo Linguístico	FP Superior	FP Inferior
H - High	(56,100,100)	(68,100,100)
M - Medium	(31,46,68,83)	(43,58,56,71)
L - Low	(0,0,38,58)	(0,0,26,46)

A variável de saída, Weight, foi modelada empregando funções de pertinência trapezoidal. Os Termos Linguísticos e seus valores são apresentados na Tabela 3.

Tabela 3. Framework Fuzzy - Variável Linguística Tipo-2: Weight

Termo Linguístico	FP Superior	FP Inferior
H - High	(46,76,100,100)	(58,88,100,100)
M - Medium	(6,36,63,93)	(18,48,51,81)
L - Low	(0,0,28,53)	(0,0,16,41)

3.2. Repositório, Seletor e Base de Regras

O Repositório de Regras tem o objetivo de armazenar todos os conjuntos de regras e as funções de pertinência desenvolvidas pelo especialista. Com isso, o Seletor de Regras filtra o conjunto que será utilizado no processo de classificação.

Os sistemas baseados em regras Fuzzy são executados mais rapidamente do que os sistemas convencionais baseados em regras, sendo mais fáceis de entender, ler, adicionar, modificar serviços, apresentam conhecimento natural em representação uniforme e também separam conhecimento do processamento, reduzindo a complexidade do tempo [Priya and Chandramathi 2014].

A Base de Regras descreve como as variáveis são processadas em Termos Linguísticos no Motor de Inferência. No controlador desenvolvido por Mamdani [Mamdani and Assilian 1975] foi adotado um método para o processo de decisão baseado em regras do tipo “If X Then Y”, nas quais tanto o antecedente quanto o consequente são valores de variáveis linguísticas, expressos por meio de conjuntos Fuzzy.

Os principais operadores Fuzzy utilizados são a união Fuzzy e a intersecção Fuzzy. O método Mamdani agrega as regras através do operador OU (união), que é modelado pelo operador máximo e, em cada regra, o operador lógico E, modelado pelo operador mínimo (intersecção).

A Tabela 4 exhibe o conjunto de regras utilizadas pelo motor de inferência Fuzzy, no perfil “Default”. Além deste perfil, é possível criar e configurar outros perfis, empregando os seguintes procedimentos: (i) cadastrando o nome do perfil; (ii) fazendo um *upload* para o *framework* do conjunto de regras Fuzzy a ser empregado; (iii) bem como o *upload* das respectivas funções de pertinência (vide Figura 2).

As regras Fuzzy devem ser do tipo Mamdani e empregar o formato definido na Tabela 4 (por exemplo, *LowRT and LowAV and LowTH LowWeight*). Por sua vez, as funções de pertinência precisam estar no formato trapezoidal (*trapezoidal,highRT,superior,40,55,59,74*) ou triangular (*triangular,highRT,inferior,65,100,100*). Caso não sejam selecionadas as funções de pertinência, ou não seja definido o conjunto de regras a serem utilizados, o perfil “Default” será considerado.

Tabela 4. Framework Fuzzy - Base de Regras Mamdani

Regra	Antecedente	Consequente
01	LowRT and LowAV and LowTH	LowWeight
02	LowRT and LowAV and MediumTH	LowWeight
03	LowRT and LowAV and HighTH	LowWeight
04	LowRT and MediumAV and LowTH	LowWeight
05	LowRT and MediumAV and MediumTH	LowWeight
06	LowRT and MediumAV and HighTH	LowWeight
07	LowRT and HighAV and LowTH	LowWeight
08	LowRT and HighAV and MediumTH	LowWeight
09	LowRT and HighAV and HighTH	LowWeight
10	MediumRT and LowAV and LowTH	LowWeight
11	MediumRT and LowAV and MediumTH	LowWeight
12	MediumRT and LowAV and HighTH	LowWeight
13	MediumRT and MediumAV and LowTH	LowWeight
14	MediumRT and MediumAV and MediumTH	MediumWeight
15	MediumRT and MediumAV and HighTH	MediumWeight
16	MediumRT and HighAV and LowTH	LowWeight
17	MediumRT and HighAV and MediumTH	MediumWeight
18	MediumRT and HighAV and HighTH	MediumWeight
19	HighRT and LowAV and LowTH	LowWeight
20	HighRT and LowAV and MediumTH	LowWeight
21	HighRT and LowAV and HighTH	LowWeight
22	HighRT and MediumAV and LowTH	LowWeight
23	HighRT and MediumAV and MediumTH	MediumWeight
24	HighRT and MediumAV and HighTH	MediumWeight
25	HighRT and HighAV and LowTH	LowWeight
26	HighRT and HighAV and MediumTH	MediumWeight
27	HighRT and HighAV and HighTH	HighWeight



Figura 2. Modo de Administração - Cadastro de Perfil

A preferência do cliente é informada para os três atributos definidos. O framework possibilita a utilização de mais atributos, entretanto, neste caso a base de regras deve ser aumentada para contemplar todas as diferentes condições.

A lógica utilizada na construção das regras é que o menor valor dos Termos Linguísticos entre as variáveis de entrada, seja utilizada na variável de saída. Ex. A regra 08 pode ser desmembrada da seguinte forma: **If** RT = Low **and** AV = High **and** TH = Medium **then** Weight = Low. A variável de entrada RT possui o menor termo linguístico (Low), este termo é o mesmo utilizado na variável de saída (Weight).

3.3. Motor de Inferência Fuzzy

Este bloco fornece a saída de cada entrada, de acordo com a base de regras, construídas por operadores de composição e conectivos LF. Então, o motor de inferência Fuzzy processa essas regras, utilizando o método de Inferência de Mamdani para Conjuntos Fuzzy Tipo-2. Após concluída a inferência, é preciso transformar o Conjunto Fuzzy Tipo-2 em Conjunto Fuzzy Tipo-1, sendo empregado para isto um redutor para Fuzzy Tipo-1.

3.4. Redutor Tipo-1, Defuzzificação e Ordens Admissíveis

Considerando a melhor escolha (de granulosidade), para transformação de um Conjunto Fuzzy Tipo-2 em um Conjunto Fuzzy Tipo-1. O bloco redutor de tipo tem como objetivo utilizar o algoritmo de Karnik-Mendel (KM) que determina o mínimo (C_L) e o máximo (C_R) dos centroides de conjuntos Fuzzy Tipo-1. [Mendel 2017]

Observou-se que ao defuzzificar os intervalos para obter um valor crisp com o propósito de ordenar para gerar uma classificação final em alguns casos o valor defuzzificado de intervalos diferentes é o mesmo, dificultando o processo de classificação. Neste trabalho não está sendo realizada a defuzzificação, como a média entre dois valores, onde o primeiro calcula o ponto final esquerdo do intervalo (C_L) e o último calcula o ponto final direito (C_R). Optou-se por preservar os intervalos e ordená-los por meio de Ordens Admissíveis.

Concluindo a etapa de defuzzificação, com muitos elementos em uma lista de intervalos gerados na abordagem de conjuntos Fuzzy Tipo-2 Intervalar podem não ser comparáveis pela ordem parcial usual, uma ordem total de intervalos é considerada com base nas ordens admissíveis, conforme proposto em [Xu and Yager 2006, Zumelzu et al. 2020].

Assim, nesta proposta, o desenvolvimento do Framework Fuzzy não se restringiu ao uso de operadores de agregação (como média aritmética) executados sobre os limites superior e limite dos dados do intervalo no procedimento final da etapa de defuzzificação.

Essa nova estratégia de preservação de dados de intervalo nos permite aplicar métricas (por exemplo, diâmetro) estendendo a análise dos resultados do intervalo, considerando não apenas a incerteza associada aos dados de entrada, mas também a imprecisão relacionada aos cálculos do processo de classificação. Essa proposta modela as incertezas relacionadas a possível indecisão das preferências do cliente e a imprecisão dos cálculos computacionais. Assim, é possível garantir a comparação dos dados do intervalo de saída.

Portanto, esta proposta considera a ordem de Xu e Yager, onde os recursos são classificados em ordem decrescente de intervalos através da ordem admissível proposta por [Xu and Yager 2006].

4. Avaliação do Framework Proposto

Neste cenário, avalia-se as funcionalidades do Classificador Fuzzy no tratamento da incerteza na especificação das preferências do cliente por meio de variáveis linguísticas e na possibilidade do especialista adequar as funções de pertinência para que os termos *Low*, *Medium* e *High* aproximem-se da necessidade do cliente.

Oportuno registrar que o algoritmo MCDA WSRF é entendido pela literatura da área como de referência em termos de classificação de recursos [Branke 2016]. Neste sentido, é comparado o resultado da classificação obtida pelo Framework Fuzzy proposto, com a obtida pelo algoritmo MCDA WsRF, ao qual são realizadas requisições por recursos com diferentes atributos de QoS. Estes recursos IoT considerados, são disponibilizados pelo *QWS Dataset* versão 2.0 [Al-Masri and Mahmoud 2008] com 2.507 recursos atualizados em novembro de 2019. O Dataset contempla ao todo nove atributos e destes foram selecionados os três entendidos como mais significativos nas demandas da IoT: Response Time (RT), Availability (AV) e Throughput (TH).

O Classificador MCDA WsRF foi desenvolvido na linguagem Python e o Framework Fuzzy proposto, foi desenvolvido em Java, com o emprego da ferramenta Juzzy, proposta por [Wagner 2013].

A Figura 3 apresenta a interface do Framework proposto para a especificação das preferências pelos clientes.

EXEHDA

EXEHDA-RR - Preferências do Cliente

Início

Response Time (RT):

Availability (AV):

Throughput (TH):

Perfil:

Aplicar preferências

© EXEHDA

Figura 3. Preferências do Cliente

Foi realizada uma avaliação classificando 10 recursos (Tabela 5) com as preferências do cliente da seguinte forma: HML, ou seja, o atributo RT (High), AV (Medium) e TH (Low), com o perfil de regras “*Default*”. Estas preferências foram adaptadas ao algoritmo MCDA WsRF, com os seguintes pesos para os atributos: RT = 1, AV = 0.6 e TH = 0.3. Após o cliente clicar em “Aplicar preferências”, é então exibida a tela com o resultado da classificação para a preferência selecionada, HML, Tabela 6. O resultado é

Tabela 5. Amostra com 10 Recursos

ID	Original			Normalizado		
	RT	AV	TH	RT	AV	TH
01	220,88	93	11,1	99,34	93	90,24
02	240,03	95	7,6	91,42	95	61,79
03	260,84	80	5,2	84,12	80	42,28
04	254,97	70	4,9	86,06	70	39,84
05	219,43	72	12,3	100	72	100
06	244,11	85	7,2	89,89	85	58,54
07	223,18	50	10,9	98,32	50	88,62
08	251,31	86	6,8	87,31	86	55,28
09	248,21	85	9,7	88,40	85	78,86
10	281,13	65	6,4	78,05	65	52,03

ordenado por ID do recurso e os valores dos atributos RT, AV e TH estão normalizados. O Centroe inferior (CI) e o Superior (CS) representam o intervalo resultante da defuzzificação e a última coluna apresenta a classificação, sendo valor 1 para o melhor e 10 para o pior recurso.

Tabela 6. Resultado da Classificação dos 10 Recursos

ID	RT	AV	TH	CI	CS	Classificação
1	99,34	93	90,24	61,36	68,21	1
2	91,42	95	61,79	59,01	63,72	2
3	84,12	80	42,28	36,70	47,45	7
4	86,06	70	39,84	33,11	46,36	9
5	100	72	100	43,01	52,80	6
6	89,89	85	58,54	48,56	57,87	5
7	98,32	50	88,62	19,13	23,20	10
8	87,31	86	55,28	49,95	58,80	3
9	88,40	85	78,86	48,70	57,82	4
10	78,05	65	52,03	31,88	51,81	8

A Figura 4 mostra a classificação dos 10 recursos, o eixo horizontal representa a ID do recurso e, o eixo vertical a classificação obtida com ambos os classificadores.

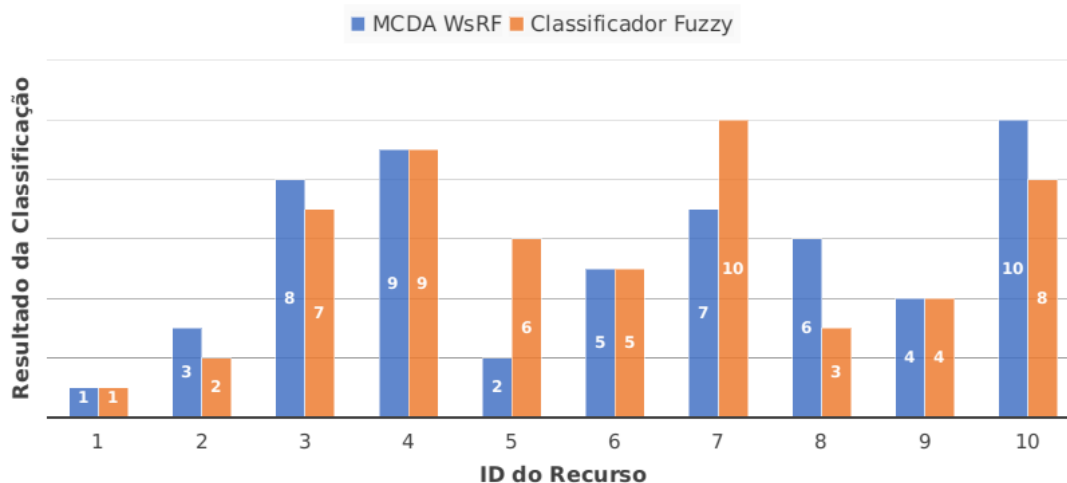


Figura 4. Comparação dos Classificadores Fuzzy e WsRF

Observa-se que a maior diferença na classificação ocorreu com o recurso ID 05. Este recurso possui excelente RT (219.43), atende a preferência RT - High, mas o valor de AV (72) não atende a preferência AV - Medium, conforme a função de pertinência especificada pelo especialista. Isto justifica a grande diferença na classificação do recurso ID 05, com o algoritmo MCDA WsRF o recurso recebeu a classificação 2, com o Classificador Fuzzy recebeu a classificação 6.

5. Considerações Finais

O processo de seleção de recursos na Internet das Coisas, considerando as preferências do cliente, pode tornar-se uma tarefa complexa, pois além de ser necessário manipular uma grande quantidade de recursos, o cliente pode ter incertezas quanto aos atributos de QoS.

Este trabalho teve como contribuição central a concepção de um Framework para gerenciamento de regras Fuzzy, agregando funções para criação de regras e definição de preferências dos clientes. As funcionalidades são disponibilizadas por uma aplicação web de uso intuitivo.

O Framework proposto se integra a arquitetura de software do EXEHDA-RR, utilizando Lógica Fuzzy Tipo-2 Intervalar para tratamento de incertezas associadas ao processo de seleção de recursos na IoT.

Os recursos empregados na avaliação foram extraídos do *QWS Dataset*, considerado internacionalmente para esta finalidade. Foram empregados na avaliação dos classificadores os três atributos considerados mais significativos para o cenário da IoT: Response Time; Availability e Throughput.

Os resultados da classificação de um conjunto de recursos usando o algoritmo MCDA WsRF, entendido como referência na área, foi comparado com aqueles produzidos pelo Framework Fuzzy proposto e os resultados se mostram promissores. Entende-se que oferecer para o cliente a possibilidade de manipular a relevância de cada atributo de QoS, empregando uma interface de visualização confortável, se valendo de variáveis linguísticas, pode contribuir para reduzir os níveis de incerteza quando da especificação de suas preferências.

Dentre os trabalhos futuros que poderiam dar continuidade à pesquisa, destacá-riamos a geração automatizada de regras Fuzzy, que além de poder diminuir consideravelmente o número de regras, facilitaria na manipulação destas. Outro item a ser implementado é gerar essas regras tendo como parâmetro o histórico dos dados referentes aos atributos dos serviços providos pelos recursos.

Referências

- Al-Masri, E. and Mahmoud, Q. H. (2008). Investigating web services on the world wide web. In *Proceedings of the 17th international conference on World Wide Web*, pages 795–804.
- Alhadithy, H. and Al-Shargabi, B. (2018). Fuzzy rule based web service composition in cloud. In *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*, pages 1–4.
- Barros, L. and Bassanezi, R. (2010). *Tópicos de lógica fuzzy e biomatemática*. UNICAMP/IMECC.

- Belouaar, H., Kazar, O., and Kabachi, N. (2018). A new model for web services selection based on fuzzy logic. *Courrier du Savoir*, 1(26):393–400.
- Branke, J. (2016). Mcdm and multiobjective evolutionary algorithms. In *Multiple Criteria Decision Analysis*, pages 977–1008. Springer.
- Dilli, R., Argou, A., Pernas, A., Reiser, R., and Yamin, A. (2018). EXEHDA-RR : Uma proposta para tratar incertezas e otimizar o processo de classificação de recursos na IoT. *Simpósio Brasileiro de Computação Ubíqua - CSBC*.
- IHS Markit (2017). The Internet of Things: a movement, not a market.
- Jatoth, C., Gangadharan, G., Fiore, U., and Buyya, R. (2019). Selcloud: a hybrid multi-criteria decision-making model for selection of cloud services. *Soft Computing*, 23(13):4701–4715.
- Khutade, P. A. and Phalnikar, R. (2014). QoS Aware Web Service Selection and Ranking Framework Based on Ontology. *International Journal of Soft Computing and Engineering (IJSCE)*, 4(3):77–81.
- Kumar, R. R., Mishra, S., and Kumar, C. (2017). Prioritizing the solution of cloud service selection using integrated MCDM methods under Fuzzy environment. *Journal of Supercomputing*, 73(11):4652–4682.
- Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*.
- Mendel, J. M. (2017). *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*. Springer International Publishing, 2 edition.
- Priya, N. H. and Chandramathi, S. (2014). QoS Based Optimal Selection of Web Services Using Fuzzy Logic. *Journal of Emerging Technologies in Web Intelligence 6.3 (2014): 331-339*, 6(3):331–339.
- Silva, M., Cardoso, M. A., Machado, M. C., and Ferreira, A. P. L. (2019). Sistema de inferência fuzzy para estimativa de crescimento populacional. *Anais do Salão Internacional de Ensino, Pesquisa e Extensão*, 11(2).
- Wagner, C. (2013). Juzzy - A Java based toolkit for Type-2 Fuzzy Logic. *Proceedings of the 2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems, T2FUZZ 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, 1(April 2013):45–52.
- Xu, Z. and Yager, R. R. (2006). Some geometric aggregation operators based on intuitionistic fuzzy sets. *International Journal of General Systems*, 35(4):417–433.
- Zumelzu, N., Bedregal, B., Mansilla, E., Bustince, H., and Díaz, R. (2020). Admissible orders on fuzzy numbers. *arXiv preprint arXiv:2003.01530*.