

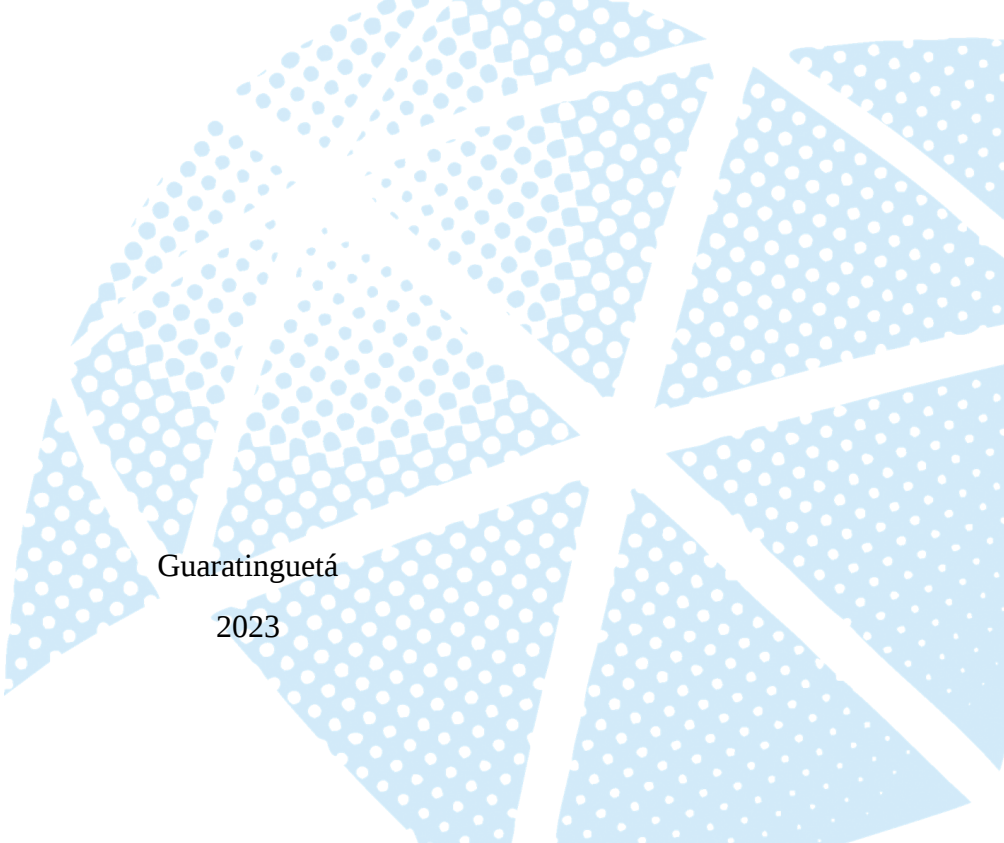
UNIVERSIDADE ESTADUAL PAULISTA – UNESP
Faculdade de Engenharia e Ciências – Campus de Guaratinguetá

RAQUEL DE OLIVEIRA PAULA ANDRADE

**COMPARAÇÃO ENTRE MODELOS *FUZZY* E *FUZZY* TIPO-2 INTERVALAR PARA
VERIFICAR OS IMPACTOS DA POLUIÇÃO DO AR NA SAÚDE INFANTIL**

Guaratinguetá

2023



RAQUEL DE OLIVEIRA PAULA ANDRADE

**COMPARAÇÃO ENTRE MODELOS *FUZZY* E *FUZZY* TIPO-2 INTERVALAR PARA
VERIFICAR OS IMPACTOS DA POLUIÇÃO DO AR NA SAÚDE INFANTIL**

Trabalho de Graduação apresentado ao
Conselho de Curso de Graduação em
Engenharia Elétrica da Faculdade de
Engenharia do Campus de Guaratinguetá,
Universidade Estadual Paulista, como parte dos
requisitos para obtenção do diploma de
Graduação em Engenharia Elétrica

Orientador (a): Paloma Maria Silva Rocha
Rizol

Coorientador (a): Luiz Fernando Costa
Nascimento

Guaratinguetá

2023

A554c	Andrade, Raquel de Oliveira Paula Comparação entre modelos fuzzy e fuzzy tipo-2 intervalar para verificar os impactos da poluição do ar na saúde infantil / Raquel de Oliveira Paula Andrade - Guaratinguetá, 2023. 63 f : il. Bibliografia: f. 62-63 Trabalho de Graduação em Engenharia Elétrica - Universidade Estadual Paulista, Faculdade de Engenharia e Ciências de Guaratinguetá, 2023. Orientadora: Prof.^a Dr.^a Paloma Maria Silva Rocha Rizol Coorientador: Prof. Dr. Luiz Fernando Costa Nascimento 1. Lógica difusa. 2. Ar - Poluição. 3. Doenças respiratórias em crianças. 4. Sistemas difusos. I. Título. CDU 504.05
-------	---

RAQUEL DE OLIVEIRA PAULA ANDRADE

**COMPARAÇÃO ENTRE MODELOS *FUZZY* E *FUZZY* TIPO-2 INTERVALAR PARA
VERIFICAR OS IMPACTOS DA POLUIÇÃO DO AR NA SAÚDE INFANTIL**

TCC apresentada à Universidade Estadual Paulista (UNESP), Faculdade de Engenharia e Ciências, Guaratinguetá, para obtenção do título de Grau acadêmico Bacharel em Engenharia Elétrica

Data da defesa: 18/12/2023

Banca Examinadora:

Documento assinado digitalmente
 **PALOMA MARIA SILVA ROCHA RIZOL**
Data: 30/04/2024 16:55:48-0300
Verifique em <https://validar.itl.gov.br>

Profa. Dra. **PALOMA MARIA SILVA ROCHA RIZOL**
Orientador/UNESP-FEG

Documento assinado digitalmente
 **LUCAS FURTADO RODRIGUES**
Data: 02/05/2024 22:32:26-0300
Verifique em <https://validar.itl.gov.br>

Eng. **LUCAS FURTADO RODRIGUES**
Membro Externo

Documento assinado digitalmente
 **PEDRO NUNES LOPES NETO**
Data: 06/05/2024 11:05:21-0300
Verifique em <https://validar.itl.gov.br>

Eng. **PEDRO NUNES LOPES**
Membro Externo

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, fonte da vida e da graça. Agradeço pela minha vida, minha inteligência, minha família e meus amigos;

À minha orientadora, Profa. Dra. Paloma Maria Silva Rocha Rizol que me acompanhou durante boa parte da minha graduação e sempre teve muita paciência para me aconselhar e dedicação em compartilhar seu conhecimento.

Ao meu co-orientador, Prof. Dr. Luiz Fernando Costa Nascimento por sempre me atender com muito carisma e disposição

Aos meus pais Viviane Patricia de Oliveira Paula Andrade e Jefferson Andrade, que, apesar das dificuldades, nunca mediram esforços para me proporcionar uma educação de qualidade;

Aos meus amigos, Maria Caroline Pereira Salmi, Ana Flávia Oliveira Vieira de Abreu, Brenda Pinheiro de Barros, Victor Vinci Fantucci e Sara Rasas por sempre me apoiarem.

Aos funcionários da Faculdade de Engenharia do Campos de Guaratinguetá pela dedicação e alegria no atendimento.

RESUMO

Crianças se encontram dentro do grupo que apresenta maior suscetibilidade a exposição de poluente atmosféricos. Este estudo tem por objetivo desenvolver um modelo preditivo que seja capaz de correlacionar o número de internações de crianças por doenças respiratórias e a exposição a poluentes atmosféricos utilizando lógica *fuzzy* tipo-1 e tipo-2 e comparar o desempenho de ambas abordagens. Os dados utilizados são de residentes da cidade de São José dos Campos no período de 2015 a 2019. As variáveis de entrada utilizadas no modelo foram concentração de material particulado (PM₁₀), concentração de dióxido de nitrogênio (NO₂), temperatura mínima no dia e velocidade do vento. As informações sobre a concentração dos poluentes e os fatores climáticos foram adquiridas da CETESB enquanto que os números de internações foram obtidos do DATASUS. Dentre os dados analisados foram identificadas 3674 internações de crianças até 10 anos de idade, os valores médio de concentrações de PM₁₀ e NO₂, temperatura mínima e velocidade do vento para o período considerado foram iguais a 20,74 µg/m³ e 18,59 µg/m³, 17,48 °C e 1,37 m/s, respectivamente. Para avaliar o desempenho dos modelos foram utilizados valor de correlação de pearson (r) e erro quadrático médio (MSE), o modelo desenvolvido utilizando lógica *fuzzy* tipo-2 foi o que apresentou melhor desempenho com as seguintes métricas: $r = 0,35$ e $MSE = 7,31$. Para *fuzzy* tipo-1 teve-se como resultado $r = 0,31$ e $MSE = 9,33$. Dentro da literatura, os trabalhos que aplicam modelos *fuzzy* na medicina apresentam correlação de pearson com valores entre 0,2 e 0,4, portanto, os modelos desenvolvidos ficaram dentro do desempenho esperado. O modelo desenvolvido corresponde a uma alternativa de baixo custo para auxiliar no gerenciamento hospitalar, ao estimar quantas internações ocorrerão no dia é possível preparar todos os recursos necessários para o atendimento de crianças por problemas respiratórios.

PALAVRAS-CHAVE: Lógica *Fuzzy*; Doenças Respiratórias; Saúde Infantil; Poluição do Ar.

ABSTRACT

Children are amongst the groups most susceptible to exposure to air pollutants. This study aims to develop a predictive model that is able to correlate the number of hospitalizations of children due to respiratory diseases and exposure to air pollutants using type-1 and type-2 fuzzy logic and to compare the performance of both approaches. The data used are from residents of the city of São José dos Campos from 2015 to 2019. The input variables used in the model were particulate matter concentration (PM_{10}), nitrogen dioxide concentration (NO_2), minimum temperature on the day and wind speed. Information on the concentration of pollutants and climatic factors was acquired from CETESB, while the number of hospitalizations was obtained from DATASUS. Among the data analyzed, 3674 hospitalizations were identified. The mean values of PM_{10} and NO_2 concentrations, minimum temperature and wind speed for the considered period were equal to $20.74 \mu g/m^3$ and $18.59 \mu g/m^3$, $17.48 ^\circ C$ and $1.37 m/s$, respectively. To assess the performance of the models, Pearson's correlation value (r) and mean square error (MSE) were used. The model developed using type-2 fuzzy logic was the best performer with the following metrics: $r = 0.35$ and $MSE = 7.31$. The results for type-1 fuzzy logic were $r = 0.31$ and $MSE = 9.33$. In the literature, studies applying fuzzy models in medicine show a Pearson correlation of between 0.2 and 0.4, so the models developed were within the expected performance. The developed model corresponds to a low-cost alternative to assist hospital management. By estimating how many hospitalizations will occur during the day, it is possible to prepare all the necessary resources for the care of children with respiratory problems.

KEYWORDS: Fuzzy Logic; Respiratory Diseases; Child Health; Air Pollution.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplos de função de pertinência: (a) triangular; (b) trapezoidal; (c) sino; (d) gaussiana; (e) tipo s; (f) tipo z	16
Figura 2 – Operações entre conjuntos <i>fuzzy</i> : (a) união (máximo); (b) intersecção (mínimo) .	19
Figura 3 – Etapas de um sistema <i>fuzzy</i> tipo-1	19
Figura 4 – Método de inferência gráfica Mamdani (max-min) com entradas <i>crisp</i>	21
Figura 5 – Função de pertinência triangular com mancha de incerteza (FOU).....	22
Figura 6 – Conjunto <i>fuzzy</i> tipo 2 representado por conjunto <i>fuzzy</i> tipo 1 com incerteza no desvio padrão	23
Figura 7 – Um conjunto <i>fuzzy</i> tipo 2 representado por conjunto <i>fuzzy</i> tipo 1 com incerteza na média. A média é incerta no intervalo [0.4, 0.6].....	23
Figura 8 – Operações entre conjuntos <i>fuzzy</i> tipo 2: (a) duas funções de pertinência do tipo gaussiana; (b) união de duas funções de pertinência gaussiana; (c) intersecção de duas funções de pertinência gaussiana	24
Figura 9 – Etapas de um sistema <i>fuzzy</i> tipo-2.....	24
Quadro 1 – Comandos do sub-pacote <i>skfuzzy.control</i>	27
Quadro 2 – Comandos da biblioteca <i>skfuzzy</i> para criação de função de pertinência.....	28
Quadro 3 – Comandos da biblioteca <i>PyIT2FLS</i> para criação de antecedentes e consequentes	28
Quadro 4 - Comandos da classe Mamdani	29
Figura 10 – Acesso a base de dados de qualidade do ar	33
Figura 11 – Janela de seleção de dados de qualidade do ar a serem exportados	34
Figura 12 – Download do TABWIN	35
Figura 13 – Download da base de dados de internações	35
Figura 14 – Aplicativo TABWIN	36
Figura 15 – Expandindo arquivo <i>dbc</i> com o aplicativo TABWIN	36
Figura 16 – Base da dados de internações baixada do DATASUS.....	37
Quadro 5 – Bibliotecas usadas para análise de dados	38
Quadro 6 – Importando base de dados para análise	38
Quadro 7 – Plotando <i>boxplot</i> das variáveis de entrada.....	38
Figura 17 – <i>Boxplot</i> das variáveis de entrada: (a) PM ₁₀ ; (b) NO ₂ ; (c) temperatura mínima; (d) velocidade do vento	39

Quadro 8 – Retirando <i>outliers</i> da base de dados	40
Figura 18 – <i>Boxplot</i> das variáveis de entrada após eliminar <i>outliers</i> : (a) PM ₁₀ ; (b) NO ₂ ; (c) temperatura mínima; (d) velocidade do vento	40
Quadro 9 – Bibliotecas importadas para o modelo <i>fuzzy</i> tipo-1	41
Quadro 10 – Definição dos antecedentes e consequentes do modelo <i>fuzzy</i> tipo-1	42
Quadro 11 – Definição das funções de pertinência para o modelo <i>fuzzy</i> tipo-1	43
Figura 19 – Funções de pertinência de entrada do modelo <i>fuzzy</i> tipo-1: (a) PM ₁₀ ; (b) NO ₂ ; (c) temperatura mínima; (d) velocidade do vento	43
Figura 20 – Funções de pertinência de saída do modelo <i>fuzzy</i> tipo-1	44
Quadro 12 – Base de regras para o modelo <i>fuzzy</i>	44
Quadro 13 – Base de regras do modelo <i>fuzzy</i> tipo-1	45
Quadro 14 – Preparando dados para simulação do modelo.....	46
Quadro 15 – Simulação do modelo <i>fuzzy</i> tipo-1	46
Quadro 16 – Código para validação do modelo	47
Quadro 17 – Bibliotecas importadas para o modelo <i>fuzzy</i> tipo-2	48
Quadro 18 - Definição dos antecedentes e consequentes do modelo <i>fuzzy</i> tipo-2	48
Figura 21 - Funções de pertinência de entrada do modelo <i>fuzzy</i> tipo-2: (a) PM ₁₀ ; (b) NO ₂ ; (c) temperatura mínima; (d) velocidade do vento	50
Figura 22 – Funções de pertinência de saída do modelo <i>fuzzy</i> tipo-2	50
Quadro 19 – Criação de um sistema <i>fuzzy</i> tipo-2 com método de inferência Mamdani	51
Quadro 20 – Adicionando as variáveis de entrada e saída no modelo <i>fuzzy</i> tipo	51
Quadro 21 – Criando base de regras do modelo <i>fuzzy</i> tipo-2	52
Quadro 22 – Simulação do modelo <i>fuzzy</i> tipo-2.....	53
Figura 23 – Distribuição dos valores das variáveis PM10 e número de internações por dia ..	54
Figura 24 – Distribuição dos valores das variáveis NO2 e número de internações por dia	55
Figura 25 – Distribuição dos valores das variáveis temperatura mínima e número de internações por dia.....	55
Figura 26 – Distribuição dos valores das variáveis velocidade do vento e número de internações por dia.....	56
Figura 27 – Exemplo de ativação das funções de pertinência de entrada do modelo <i>fuzzy</i> tipo-1: (a) PM ₁₀ ; (b) NO ₂ ; (c) temperatura mínima; (d) velocidade do vento	58
Figura 28 – Exemplo de ativação das funções de pertinência de saída do modelo <i>fuzzy</i> tipo.....	58

Figura 29 – Exemplo de ativação das funções de pertinência de saída do modelo *fuzzy* tipo-2:
(a) saída fuzzy tipo-2; (b) saída com tipo reduzido59

LISTA DE TABELAS

Tabela 1 – Publicações no Scopus por termo de pesquisa.....	13
Tabela 2 – Valores médios, mínimos e máximos com os respectivos desvios padrão das variáveis PM10, NO2, temperatura mínima, velocidade do vento e número de internações. São José dos Campos, 2015-2019	54
Tabela 3 - Comparação entre os modelos <i>fuzzy</i> tipo-1 e tipo-2 com FOU de 0,1	56
Tabela 4 - Comparação entre os modelos <i>fuzzy</i> tipo-1 e tipo-2	57

LISTA DE ABREVIATURAS E SIGLAS

CETESB	Companhia Ambiental do Estado de São Paulo
CONAMA	Conselho Nacional do Meio Ambiente
DATASUS	Departamento de Informática do Sistema Único de Saúde
dbc	Data base CAN
dbf	Data base file
FOU	Mancha de incerteza
IDH	Índice de Desenvolvimento Humano
INPE	Instituto Nacional de Pesquisas Espaciais
ITA	Instituto Tecnológico da Aeronáutica
MATLAB	<i>Matrix Laboratory</i>
MSE	Erro Quadrático Médio
OMS	Organização Mundial da Saúde
QUALAR	Qualidade do Ar
SUS	Sistema Único de Saúde
TABWIN	Tab para Windows
UNESP	Universidade Estadual Paulista

LISTA DE SÍMBOLOS

O ₃	ozônio
SO ₂	dióxido de enxofre
NO ₂	dióxido de nitrogênio
PM ₁₀	material particulado com diâmetro inferior a 10 micrômetros
µm	micrômetro
km ²	quilômetro quadrado
µg / m ³	micrograma por metro cúbico
°C	graus Celsius
m/s	metros segundo

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVO GERAL	13
1.2	OBJETIVOS ESPECÍFICOS	14
1.3	ESTRUTURA DO TRABALHO	14
2	REFERENCIAL TEÓRICO	15
2.1	INTRODUÇÃO A LÓGICA <i>FUZZY</i>	15
2.1.1	Conceitos Básicos dos Conjuntos <i>Fuzzy</i>	16
2.2	SISTEMA <i>FUZZY</i> TIPO 1	19
2.2.1	Fuzzificador	19
2.2.2	Base de Regras	20
2.2.3	Inferência	20
2.2.4	Defuzzificador	21
2.3	SISTEMA <i>FUZZY</i> TIPO-2	22
2.3.1	Redutor de Tipo	25
2.4	SISTEMAS <i>FUZZY</i> COM PYTHON	26
2.4.1	Sistemas <i>Fuzzy</i> do Tipo-1 com Python	27
2.4.2	Sistemas <i>Fuzzy</i> do Tipo-2 com Python	28
2.5	MÉTRICAS DE AVALIAÇÃO	29
2.5.1	Correlação de Pearson	29
2.5.2	Erro Quadrático Médio (MSE)	30
2.5.3	Raiz Quadrática do Erro Médio (RMSE)	30
2.6	POLUENTES ATMOSFÉRICOS	30
3	MATERIAIS E MÉTODOS	32
3.1	BASE DE DADOS	33
3.1.1	Cetesb	33
3.1.2	Datasus	34
3.1.3	Tratamento dos dados	37
3.2	MODELO <i>FUZZY</i> TIPO-1	40
3.3	MODELO <i>FUZZY</i> TIPO-2	47
4	RESULTADOS	54
4.1	SIMULAÇÃO DO MODELO	57
5	CONSIDERAÇÕES FINAIS	60
	REFERÊNCIAS	62

1 INTRODUÇÃO

A ascensão progressiva de atividades industriais, queima de combustíveis fósseis e outras atividades antrópicas vêm aumentando a concentração de poluentes atmosféricos no ar, tais como o ozônio (O_3), o dióxido de enxofre (SO_2), o dióxido de nitrogênio (NO_2), o material particulado com diâmetro inferior a $10\ \mu m$ (PM_{10}), etc. Atualmente, existem diversas evidências dos impactos negativos da exposição a poluentes no sistema respiratórios, sendo crianças representam o grupo que é mais afetado.

Em uma revisão realizada pelo Fórum Internacional das Sociedades (2017), constatou-se que infecções no trato respiratório inferior causam aproximadamente 4 milhões de mortes por ano e são a principal causa de morte em crianças abaixo de 5 anos. A pneumonia matou 920.136 crianças abaixo de 5 anos de idade em 2015, representando 15% das mortes nesta faixa etária. Além das infecções no trato respiratório inferior, a asma também é uma doença que gera preocupações, afetando 14% das crianças no mundo. Mediante aos riscos envolvidos na contração de doenças respiratórias por crianças e a alta exposição das mesmas à poluentes atmosféricos, é de grande relevância estudar como esses fatores se relacionam e prever seus efeitos.

Em 2015 o Departamento de Informática do Sistema Único de Saúde (DATASUS) registrou cerca de 812 mil internações por doenças respiratórias tais como asma, pneumonia, bronquite aguda, bronquiolite aguda, laringite e traqueíte aguda, gerando um custo total de aproximadamente 705 milhões de reais. Destas internações aproximadamente 72% correspondem a crianças com idade entre 0 e 10 anos (MINISTÉRIO DA SAÚDE, 2017).

Além da emissão de fatores contaminantes, ainda há outros dados que corroboram para uma maior exposição da população à poluentes. O município de São José dos Campos tem a indústria como uma das principais atividades econômicas e, por estar localizado em um vale, possui dificuldade para dispersão dos poluentes. A direção predominante do vento no município também é um fator que deixa os moradores de determinados bairros mais expostos a poluição. Segundo Carvalho (2004) a poluição gerada a leste de São José dos Campos tende a ser levada para o Centro e os bairros Jardim Satélite, Bosque dos Eucaliptos, Jardim Morumbi, Parque Industrial, e outros, potencializando assim a poluição gerada no Centro da cidade e nestes bairros, onde estão localizados os maiores aglomerados urbanos da cidade.

Apesar de já estar evidente para a medicina que a exposição a poluentes atmosféricos é prejudicial para o trato respiratório, é difícil traçar uma relação linear entre estes fatores. Portanto, optou-se por trabalhar com Lógica *Fuzzy* por considerar incertezas inerentes à representação humana dos fenômenos da natureza.

Em uma lógica que trabalha com conjuntos clássicos, tem-se que uma variável pertence ou não a um conjunto. Enquanto isso, a lógica *fuzzy* utiliza da teoria dos conjuntos nebulosos, introduzida por Zadeh (1965) para tratar do aspecto vago das informações. Um conjunto nebuloso permite que uma variável seja caracterizada com um termo linguístico que possui um grau de pertinência variando entre 0 e 1. Dessa forma, diferente do que ocorre na lógica clássica na qual um elemento pertence a um conjunto, na lógica *fuzzy* o elemento pode pertencer a dois conjuntos com diferentes graus de pertinência. Isso permite a tradução de converter imprecisões inerentes à linguagem natural em um modelo numérico.

Foi realizada uma revisão bibliométrica dentro do banco de dados Scopus sobre as vertentes abordadas neste trabalho, o resultado está descrito na tabela abaixo.

Tabela 1 – Publicações no Scopus por termo de pesquisa

Termo utilizado na pesquisa	Número de artigos encontrados
"Fuzzy"	397.970
"Fuzzy" "Saúde"	299
"Fuzzy" "Saúde" "Infantil"	13
"Fuzzy" "Saúde" "Infantil" "Poluição"	4

Fonte: Autor (2023)

Dentro os estudos que correspondem às palavras-chave inseridas, dois deles foram realizados utilizando dados do município de São José dos Campos, mas só tratam de casos de asma e pneumonia, deixando espaço para explorar a correlação entre a poluição do ar com outras doenças respiratórias, tais como laringite, traqueíte, bronquite e bronquiolite.

1.1 OBJETIVO GERAL

Mediante os cenários descritos, o trabalho em questão tem por objetivo desenvolver um modelo *fuzzy* para estimar quais os efeitos da poluição do ar na saúde infantil levando em consideração a concentração de poluentes e as características geográficas do município São José dos Campos. Além de analisar como a concentração de poluentes se relaciona com o número de internações de crianças, este trabalho também tem o intuito de comparar o

desempenho de modelos *fuzzy* do tipo-1 e do tipo-2. Para tal, foram desenvolvidos dois modelos utilizando a linguagem de programação Python.

1.2 OBJETIVOS ESPECÍFICOS

- Identificar dados de internações de crianças por doenças respiratórias na cidade de São José dos Campos-SP, nos anos de 2015 a 2019.
- Identificar relatórios diários dos poluentes do ar, PM₁₀ e NO₂; e fatores atmosféricos, temperatura mínima e velocidade do vento, para o mesmo período.
- Desenvolver um modelo baseado na lógica *fuzzy* tipo-1 e tipo-2, relacionando a exposição aos fatores descritos no item anterior com o número de internações de crianças por doenças respiratórias
- Comparar o desempenho dos modelos *fuzzy* tipo-1 e tipo-2.

1.3 ESTRUTURA DO TRABALHO

Este trabalho é dividido em 5 capítulos, no primeiro capítulo é feita uma contextualização do problema estudado e apresentado os objetivos do trabalho.

O segundo capítulo conta com uma revisão teórica para compreender os princípios básicos da teoria dos conjuntos *fuzzy* e como são modelados os sistemas do *fuzzy* do tipo-1 e tipo-2, conjuntamente é realizada uma breve apresentação das bibliotecas Python utilizadas para a criação dos modelos e das métricas utilizadas para validação dos mesmos. Neste capítulo é feita também uma revisão dos principais poluentes atmosféricos que afetam a saúde.

No terceiro capítulo é exposto como foi realizada a coleta de dados de poluentes do ar e internações por doenças respiratórias. Também é apresentado os programas utilizados no desenvolvimento dos modelos *fuzzy* tipo-1 e tipo-2.

O quarto capítulo mostra os resultados obtidos.

No último capítulo são discutidas as limitações dos sistemas desenvolvidos e sugeridas algumas alternativas para aumentar a confiabilidade do modelo.

2 REFERENCIAL TEÓRICO

O capítulo de Referencial Teórico apresenta a revisão literária dos conceitos estudados para desenvolvimento do trabalho.

2.1 INTRODUÇÃO A LÓGICA *FUZZY*

Desde pequeno o ser humano é estimulado a aplicar o método dedutivo para tomada de decisões em situações cotidianas. Quando uma pessoa observa as nuvens com um aspecto um pouco mais escuro, deduz-se que as nuvens estão carregadas e que irá chover naquele dia, com essa informação é possível escolher as roupas e acessórios adequados para utilizar ao sair de casa. Chegar na conclusão de que irá chover naquele dia em específico é o resultado de anos observando que quando chove o céu geralmente fica mais escuro, então pela constatação de situações gerais o ser humano consegue deduzir o resultado de situações específicas.

Esta é uma situação simples, mas que descreve uma problemática que surgiu com o advento de modelos computacionais: como traduzir de forma matemática as informações constatadas por meio da observação? “Céu escuro” é uma descrição vaga que depende muito da interpretação de cada observador. Além disso, entre o claro e o escuro existem outras tonalidades, o que pode interferir na tomada de decisão implicando também em um certo grau de imprecisão na informação coletada. Dentro deste contexto, a aplicação de modelos *fuzzy* surgiu como uma forma de mapear entrada e saída considerando a incerteza e imprecisão presentes nos dados coletados.

Um sistema *fuzzy* tenta compreender sistemas complexos para os quais ainda não há modelos por conta de informações incertas, no sentido de serem vagas, imprecisas ou completamente inexistentes. Ou seja, sistemas nos quais a relação entre causa e efeito ainda não são completamente compreendidas, isso inclui sistemas envolvendo condição humana como sistemas médicos, sociais, econômicos ou políticos, nos quais os inúmeros conjuntos de entrada e saída não podem ser todos captados analiticamente ou controlados em qualquer sentido convencional (ROSS, 2010).

2.1.1 Conceitos Básicos dos Conjuntos Fuzzy

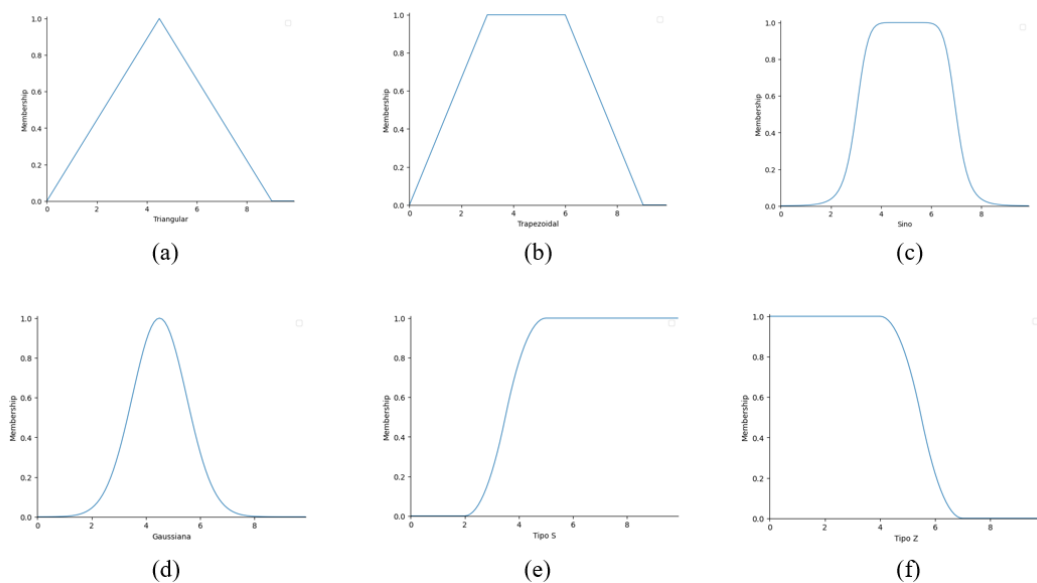
A lógica *fuzzy*, ou lógica nebulosa, utiliza da teoria dos conjuntos nebulosos, introduzida por Lotfi A. Zadeh em 1965. Um conjunto nebuloso permite que uma variável seja caracterizada com um termo linguístico que possui um grau de pertinência variando entre 0 e 1. Dessa forma, diferente do que ocorre na lógica clássica, na qual um elemento pertence ou não a um conjunto, na lógica *fuzzy* o elemento pode pertencer a dois conjuntos com diferentes graus de pertinência.

Para definir um conjunto *fuzzy*, primeiramente é necessário determinar seu domínio, também conhecido como universo de discurso. Os valores presentes no domínio de cada conjunto são chamados de valor *crisp*.

Formalmente, um conjunto nebuloso A do universo de discurso Ω é definido por uma função de pertinência $\mu_A : \Omega \rightarrow [0,1]$. Essa função associa cada elemento x de Ω o grau $\mu_A(x)$, com o qual o x pertence a A . A função de pertinência $\mu_A(x)$ indica o grau de compatibilidade entre x e o conceito expresso por A (SANDRI e CORREA, 1999).

Dessa forma, um conjunto *fuzzy* liga cada valor *crisp* a um grau de pertencimento àquele conjunto e a função de pertinência utilizada depende dos dados adquiridos, podendo ser essa trapezoidal, triangular, gaussiana, entre outras.

Figura 1 – Exemplos de função de pertinência: (a) triangular; (b) trapezoidal; (c) sino; (d) gaussiana; (e) tipo s; (f) tipo z



Fonte: Autor (2023)

Uma função de pertinência triangular é definida por três parâmetros $\{a,b,c\}$ que determinam as coordenadas dos três pontos que estão sob a função triangular, sendo $a < b < c$ (CASTILLO e MELIN, 2008). O ponto ‘a’ indica o início da reta crescente que forma o triângulo, o ponto ‘b’ é onde a reta crescente termina e inicia a reta decrescente (ponto máximo com grau de pertinência é igual a 1) e o ponto ‘c’ sinaliza o fim da reta decrescente. Na Figura 1 (a) tem-se o exemplo de uma função de pertinência triangular definida pelos parâmetros $\{0, 4.5, 9\}$. A equação (1) descreve matematicamente uma função de pertinência triangular.

$$y = \text{triângulo}(x; a, b, c) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x = b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x > c \end{cases} \quad (1)$$

A função de pertinência trapezoidal é definida pelos parâmetros $\{a,b,c,d\}$ com $a < b < c < d$. Sendo ‘a’ o início da reta crescente, ‘b’ e ‘c’ delimita a base menor (região com grau de pertinência igual a 1) e ‘d’ define o fim do trapézio. A equação (2) descreve a função de pertinência trapezoidal e na Figura 1 (b) tem-se o exemplo de uma função com parâmetros definidos em $\{0, 3, 6, 9\}$

$$y = \text{trapézio}(x; a, b, c, d) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \\ 0, & x > d \end{cases} \quad (2)$$

A função do tipo sino é definida segundo os parâmetros $\{a,b,c\}$ que indicam o controle de largura, centro e inclinação, respectivamente. Na Figura 1 (c) tem-se o exemplo de uma função com os parâmetros definidos em $\{2, 4, 5\}$ e a equação (3) mostra como a função de pertinência do tipo sino é descrita em termos de equação.

$$\text{sino}(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (3)$$

A função de pertinência do tipo gaussiana é definida pelos parâmetros $\{c,\sigma\}$ indicando o valor central e o desvio padrão, respectivamente. A equação (4) descreve esta função e na Figura 1 (d) tem-se como exemplo uma função de pertinência gaussiana definida com os parâmetros $\{2, 1\}$.

$$gaussiana(x; c, \sigma) = e^{-\left(\frac{x-c}{\sigma}\right)^2} \quad (4)$$

A função do tipo S é definida pelos parâmetros $\{a, b\}$, sendo ‘a’ o indicador do pé da função (onde a função começa a subir do 0) e ‘b’ o teto (onde a função é nivelada em 1). Na Figura 1 (e) tem-se o exemplo de uma função de pertinência do tipo S com parâmetros $\{2, 5\}$ e a equação desta função é descrita na equação (5).

$$s(x; a, b) = \begin{cases} 0, & x < a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \text{ e } x \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \text{ e } x \leq b \end{cases} \quad (5)$$

A função do tipo Z também é definida pelos parâmetros $\{a, b\}$, mas, neste caso, ‘a’ indica o teto e ‘b’ define onde será o pé da função. Na Figura 1 (f) tem-se o exemplo de uma função de pertinência do tipo Z com parâmetros $\{4, 7\}$ e a equação desta função é descrita na equação (6).

$$z(x; a, b) = \begin{cases} 1 - 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \text{ e } x < \frac{a+b}{2} \\ 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \text{ e } x \leq b \\ 0, & x > b \end{cases} \quad (6)$$

É também possível realizar operações entre conjuntos *fuzzy*. Dado um conjunto A e B descritos em um universo de discurso Ω , é possível definir a intersecção ($A \cap B$) e união ($A \cup B$) destes conjuntos.

A intersecção de conjuntos *fuzzy* é ditada pela família de operadores t-norma, sendo os operadores de mínimo e produto os mais utilizados. Essas duas operações são descritas abaixo, respectivamente.

$$A \cap B = \min(\mu_{A(x)}, \mu_{B(x)}) \quad (7)$$

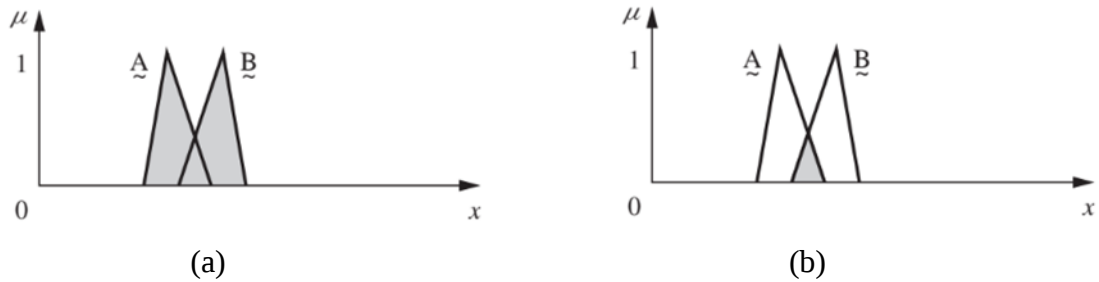
$$A \cap B = \mu_{A(x)} \times \mu_{B(x)} \quad (8)$$

Já as operações de união são definidas pela família t-conorma, sendo o operador de máximo o mais aplicado

$$A \cup B = \max(\mu_{A(x)}, \mu_{B(x)}) \quad (9)$$

Na Figura (2) é ilustrado um exemplo de união e intersecção de conjuntos *fuzzy* utilizando os operadores de máximo e mínimo.

Figura 2 – Operações entre conjuntos *fuzzy*: (a) união (máximo); (b) intersecção (mínimo)



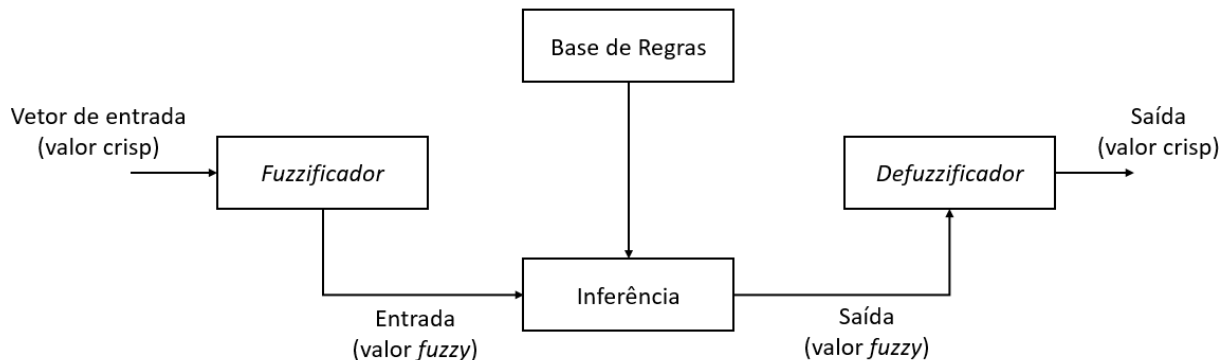
Fonte: (ROSS, 2010)

2.2 SISTEMA FUZZY TIPO 1

Um sistema *fuzzy* é um modelo que utiliza da teoria dos conjuntos *fuzzy* para tentar simular a forma de raciocínio humana para tomada de decisões. De maneira geral, um sistema de lógica *fuzzy* é um mapeamento não linear de um vetor de dados de entrada em uma saída escalar (MENDEL, 1995).

Um modelo *fuzzy* é composto por 4 blocos: fuzzificador, base de regras, inferência e defuzzificador. A forma como esses blocos se relacionam é expressa na Figura 3.

Figura 3 – Etapas de um sistema *fuzzy* tipo-1



Fonte: Autor (2023)

2.2.1 Fuzzificador

Esta etapa consiste em transformar o valor *crisp* de entrada em um valor *fuzzy*. Ou seja, o valor de entrada será mapeado e transformado em uma variável linguística com um

determinado grau de pertinência. Para tal, é identificado com qual grau de pertinência a entrada *crisp* pertence a cada uma das funções desenvolvidas.

2.2.2 Base de Regras

A base de regras estabelece a relação entre as variáveis de entrada e saída de modelo e cada regra apresenta o seguinte formato:

Se <antecedente> **então** <consequente>

O antecedente é composto por um conjunto de condições que, quando satisfeitas, ativam a regra e determinam um conjunto de ações que serão tomadas (consequente) através do método de inferência.

Abaixo tem-se um exemplo de regra considerando o método mencionado e um sistema *fuzzy* genérico de múltiplas entradas e uma saída.

Se ($x_1 = A_i$ **E** $x_2 = A_p$ **E** ... **E** $x_n = A_j$) **então** ($y_1 = B_i$)

sendo:

$x_1, x_2 \dots x_n$: entradas do sistema

y_1 : saída do sistema

A_i, A_p, A_j : função de pertinência ativada para cada uma das variáveis

B_i : função de saída ativada

O número de regras de um modelo é definido pela quantidade de combinações possíveis entre as funções de entrada. Vale ressaltar que a definição das regras requer um reconhecimento prévio do comportamento das variáveis e deve, portanto, procurar o auxílio de um especialista na área de aplicação.

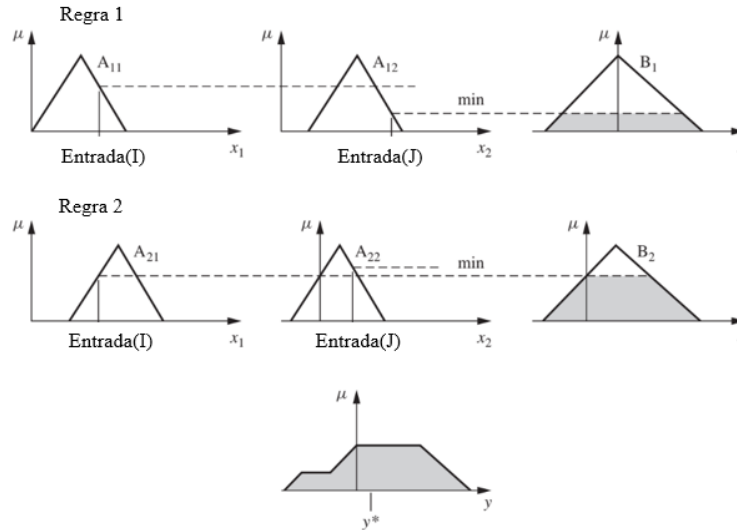
2.2.3 Inferência

A inferência é o módulo do sistema *fuzzy* responsável por fazer o mapeamento de entrada e saída. Esta é então a etapa onde é formulada a base para tomada de decisão. Existem diferentes formas de tratar as variáveis de um modelo *fuzzy*, sendo os métodos Mamdani e Sugeno os mais conhecidos. No trabalho em questão foi aplicado o método de inferência Mamdani, também conhecido como o método dos máximos e mínimos (max-min).

O método Mamdani consiste em realizar intersecção das variáveis ativadas na regra, o resultado será a função de pertinência de saída que é ativada na regra delimitada pelo valor

mínimo resultado da intersecção. Por fim, é feita a união da conclusão de cada regra e esta será a saída *fuzzy* do sistema. Na Figura (4) tem-se o exemplo das operações aplicadas no método de inferência Mamdani.

Figura 4 – Método de inferência gráfica Mamdani (max-min) com entradas *crisp*



Fonte: (ROSS, 2010)

2.2.4 Defuzzificador

Como resultado do processo de inferência, tem-se na saída do modelo um valor *fuzzy* definido pela união das funções de pertinência de saída ativadas com seus respectivos graus de pertinência. Dessa forma, a etapa final é a transformação desse valor *fuzzy* em um valor *crisp*.

Para o projeto em questão foi aplicado o método centro de área (centróide) para a defuzzificação, neste caso a saída será o centro de gravidade da função resultante da inferência.

No caso discreto, o resultado é obtido pela equação (10) e para o caso contínuo pela equação (11).

$$y(x) = \frac{\sum_{i=1}^N x_i \mu_x(x_i)}{\sum_{i=1}^N \mu_x(x_i)} \quad (10)$$

$$y(x) = \frac{\int x \cdot \mu_x(x) dx}{\int \mu_x(x) dx} \quad (11)$$

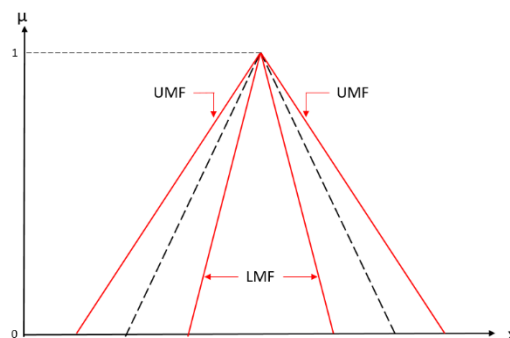
2.3 SISTEMA FUZZY TIPO-2

Apesar de a teoria dos conjuntos *fuzzy* ter sido suficiente para lidar com certas incertezas e, assim, permitindo o desenvolvimento de modelos preditivos e controladores confiáveis, ainda existem algumas questões que não são contemplados por sistemas *fuzzy* tipo-1.

Muitas vezes a base de dados e o conhecimento utilizado para construir sistemas *fuzzy* apresentam incertezas. Por exemplo, caso fosse perguntado para 50 pessoas diferentes definirem uma nota de 0 a 10 para classificar o quão escuro está o céu, ter-se-ia diferentes respostas para uma mesma condição porque palavras tem significados diferentes para diferentes pessoas e cada um tem a sua percepção do que seria um céu escuro. Este mesmo problema é encontrado ao analisar uma base de dados para construção de um modelo *fuzzy*, muitas vezes devido a subjetividade e incerteza da informação analisada, é difícil definir com qual grau de pertinência determinado valor pertence a função de pertinência. Um modelo *fuzzy* tipo-2 apresenta uma forma minimizar os efeitos das incertezas presentes nas variáveis do modelo quando não é possível determinar uma função de pertinência exata.

Na Figura 5 tem-se o exemplo de uma função de pertinência *fuzzy* tipo-2 triangular. Nota-se que a função é delimitada por uma função de pertinência inferior (LMF, do inglês *low membership function*) e uma função de pertinência superior (UMF, do inglês *upper membership function*). A função de pertinência primária (representada pela linha tracejada) passa então a ser cercada por uma mancha de incerteza (FOU, do inglês *footprint of uncertainty*) que contempla a incerteza com qual cada valor pertence ao conjunto. Enquanto sistemas *fuzzy* tipo-1 possuem graus de pertinência definidos por um valor *crisp*, para sistemas *fuzzy* tipo-2 o grau de pertinência é um valor *fuzzy* e poderia então ser chamando de “conjunto *fuzzy-fuzzy*” (MENDEL, 2007).

Figura 5 – Função de pertinência triangular com mancha de incerteza (FOU)

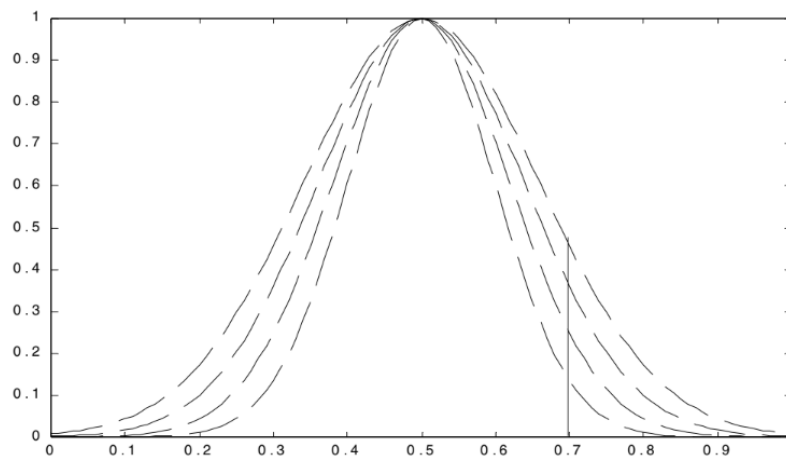


Fonte: Autor (2023)

Um conjunto *fuzzy* do tipo-2 pode ser interpretado como a união de todas as funções de pertinência possíveis para o conjunto em questão (MENDEL, JOHN, LIU, 2007). A função de pertinência pode apresentar incerteza no desvio padrão ou na média.

No caso de um conjunto *fuzzy* tipo-2 com incerteza no desvio padrão, para cada valor de σ tem-se uma função de pertinência diferente. A Figura 6 mostra o domínio *fuzzy* associado com $x = 0,7$.

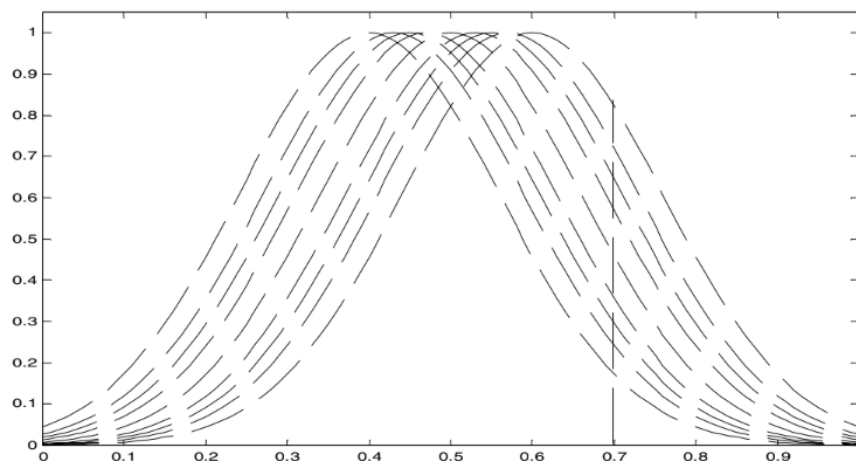
Figura 6 – Conjunto *fuzzy* tipo 2 representado por conjunto *fuzzy* tipo 1 com incerteza no desvio padrão



Fonte: (CASTILLO e MELLIN, 2008)

Para uma função de pertinência com incerteza na média, o desvio padrão é fixo e, como resultado, tem-se uma função de pertinência para cada valor de média, conforme ilustrado na Figura 7.

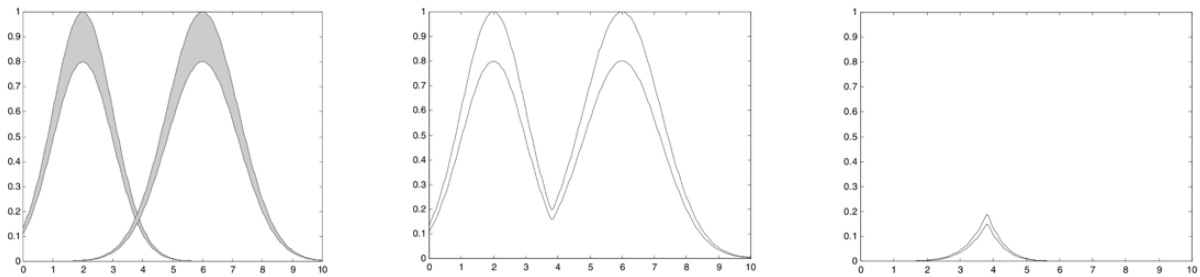
Figura 7 – Um conjunto *fuzzy* tipo-2 representado por conjunto *fuzzy* tipo-1 com incerteza na média. A média é incerta no intervalo $[0,4, 0,6]$



Fonte: (CASTILLO e MELLIN, 2008)

Os operadores de união e intersecção para conjuntos *fuzzy* tipo-2 seguem os mesmos princípios aplicados ao tipo-1. A Figura 8 apresenta um exemplo de união e intersecção de funções de pertinência *fuzzy* tipo-2 utilizando os operadores de máximo e mínimo.

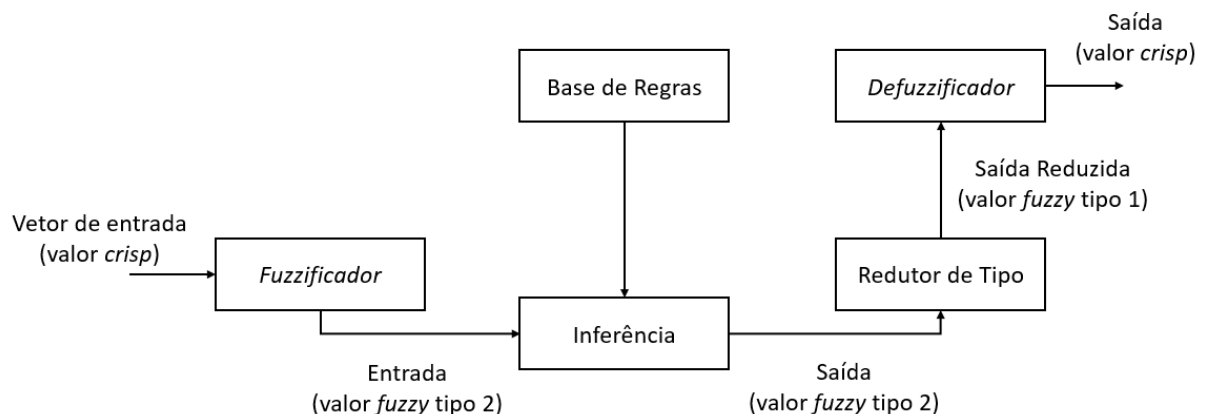
Figura 8 – Operações entre conjuntos *fuzzy* tipo-2: (a) duas funções de pertinência do tipo gaussiana; (b) união de duas funções de pertinência gaussiana; (c) intersecção de duas funções de pertinência gaussiana



Fonte: (CASTILLO e MELLIN, 2008)

Os sistemas *fuzzy* do tipo-2 apresentam as mesmas etapas dos sistemas do tipo-1, a diferença é que na saída do bloco de inferência é um valor *fuzzy* do tipo-2 e, portanto, para obter um valor *crisp* de saída é necessário reduzir o valor na saída de inferência de tipo-2 para tipo-1 (MENDEL, JOHN, LIU, 2007). Dessa forma, após a inferência é adicionado um bloco redutor de tipo conforme mostrado na Figura 9.

Figura 9 – Etapas de um sistema *fuzzy* tipo-2



Fonte: Autor (2023)

2.3.1 Redutor de Tipo

Dado um conjunto *fuzzy* do tipo-2 é possível representa-lo por um conjunto *fuzzy* do tipo-1. O tipo-redutor é obtido aplicando-se o princípio da extensão proposto por Zadeh na defuzzificação dos conjuntos tipo-1. Desta forma, existe um método de tipo-redutor equivalente para cada método de defuzzificação (RIZOL, MESQUITA, SAOTOME, 2011).

O método tipo-redutor mais aplicado é o de centro de área (centroide). O centroide de um conjunto $\tilde{B}$ ($C_{\tilde{B}}$) é definido como a coleção de todos os centroides dos conjuntos *fuzzy* do tipo-1 que compõe a FOU($\tilde{B}$). O resultado será então uma coleção de números finitos que tem, portanto, um valor máximo que está à direita (y_r) e um valor mínimo que está à esquerda (y_l). Neste trabalho será descrito o método iterativo proposto por Karnik-Mendel (KM) para encontrar os valores de y_r e y_l (RIZOL, MESQUITA, SAOTOME, 2011).

Algoritmo KM para y_l :

$$y_l = \min_{\forall \theta \in [\underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_l}), \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_l})]} \frac{\sum_{l=1}^N y^l \theta_l}{\sum_{l=1}^N \theta_l} \quad (12)$$

1º passo: inicialize θ_i com o resultado da equação (13)

$$\theta_i = \frac{\underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_l}) + \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_l})}{2} \quad (13)$$

2º passo: calcule c'

$$c' = \frac{\sum_{l=1}^N y^l \theta_l}{\sum_{l=1}^N \theta_l} \quad (14)$$

3º passo: encontre o valor de k ($k=1, \dots, N-1$), tal que $y_k \leq c' \leq y_{k+1}$

4º passo: calcular conjunto θ_i

$$\theta_i = \begin{cases} \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}), & i \leq k \\ \underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}), & i \geq k + 1 \end{cases} \quad (15)$$

5º passo: atribuir $c' \equiv c''$ no passo 2 utilizando o valor de θ_i do passo 4

6º passo: se $c'' = c'$ FIM, atribuir o valor de c'' à L . Se não, ir para o passo 7

7º passo: se $c' = c''$, ir para o passo 3

Algoritmo KM para y_r :

$$y_r = \max_{\forall \theta \in [\underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}), \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i})]} \frac{\sum_{l=1}^N y^l \theta_l}{\sum_{l=1}^N \theta_l} \quad (16)$$

1º passo: inicialize θ_i com o resultado da equação (13)

2º passo: calcule c' conforme a equação (14)

3º passo: encontre o valor de k ($k=1, \dots, N-1$), tal que $y_k \leq c' \leq y_{k+1}$

4º passo: calcular conjunto θ_i

$$\theta_i = \begin{cases} \underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}), & i \geq k + 1 \\ \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}), & i \leq k \end{cases} \quad (17)$$

5º passo: atribuir $c' \equiv c''$ no passo 2 utilizando o valor de θ_i do passo 4

6º passo: se $c'' = c'$ FIM, atribuir o valor de c'' à L . Se não, ir para o passo 7

7º passo: se $c' = c''$, ir para o passo 3

Uma vez determinado os pontos R e L , é possível determinar os valores de y_r e y_l .

$$y_r = y_r(R) = \frac{\sum_{i=1}^L y_i \underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}) + \sum_{i=L+1}^N y_i \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i})}{\sum_{i=1}^L \underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}) + \sum_{i=L+1}^N \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i})} \quad (18)$$

$$y_l = y_l(L) = \frac{\sum_{i=1}^L y_i \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}) + \sum_{i=L+1}^N y_i \underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i})}{\sum_{i=1}^L \overline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i}) + \sum_{i=L+1}^N \underline{\mu}_{\tilde{B}}(\frac{\tilde{B}}{y_i})} \quad (19)$$

Por fim, a saída *crisp* do sistema é dada pela equação (17).

$$y(x) = \frac{y_r + y_l}{2} \quad (20)$$

2.4 SISTEMAS FUZZY COM PYTHON

Python é uma linguagem de programação amplamente aplicada em análise de dados e criação de modelos de inteligência artificial e *machine learning* pela sua ampla quantidade de bibliotecas que possibilitam solucionar problemas complexos de forma simplificada

(RASCHKA, PETTERSON, NOLET, 2020). Para o desenvolvimento do trabalho em questão foram utilizadas as bibliotecas *skfuzzy* e *PyIT2FLS* para desenvolvimento dos modelos *fuzzy* tipo-1 e tipo-2, respectivamente.

2.4.1 Sistemas *Fuzzy* do Tipo-1 com Python

A biblioteca *skfuzzy*, desenvolvida por Warner e Sexauer (2019), é disponibilizada no Python para desenvolvimento de modelos *fuzzy* tipo-1, nela existem diversas funções para desenvolvimento de projetos envolvendo lógica *fuzzy*. Para criar um sistema *fuzzy* utilizando Python é então necessário importar esta biblioteca e sua classe *control*.

É através de funções presentes na classe *skfuzzy.control* que tem-se os recursos necessários para definição de antecedentes, consequentes, regras e de comandos que possibilitam a simulação do modelo. O Quadro 1 apresenta os principais comandos desta classe.

Quadro 1 – Comandos do sub-pacote *skfuzzy.control*

Comando	Descrição
<code>skfuzzy.control.Antecedent(universo de discurso, nome do conjunto)</code>	Variável antecedente (entrada) de um sistema de controle <i>fuzzy</i>
<code>skfuzzy.control.Consequent(universo de discurso, nome do conjunto)</code>	Variável consequente (saída) de um sistema de controle <i>fuzzy</i>
<code>skfuzzy.control.Rule([antecedente, ...])</code>	Cria regra de um sistema de controle <i>fuzzy</i> , conecta antecedente com consequente
<code>skfuzzy.control.ControlSystem([regra])</code>	Base de regras de um controle de sistema <i>fuzzy</i>
<code>skfuzzy.control.ControlSystemSimulation(...)</code>	Cálcula o resultado de um <i>ControlSystem</i>

Fonte: Autor (2023)

Após definido os antecedentes e consequentes, é necessário definir as funções de pertinência que os vão compor. O Quadro 2 apresenta os comandos para criação das funções de pertinência mais utilizadas, onde *x* é o antecedente/consequente ao qual aquela função pertence e os outros parâmetros correspondentes de cada função de pertinência definidos nas equações de 1 a 6.

Quadro 2 – Comandos da biblioteca *skfuzzy* para criação de função de pertinência

Comando	Descrição
<code>skfuzzy.membership.trimf(x, a, b, c)</code>	Função de pertinência triangular
<code>skfuzzy.membership.trapmf(x, a, b, c, d)</code>	Função de pertinência trapezoidal
<code>skfuzzy.membership.gbellmf(x, a, b, c)</code>	Função de pertinência sino
<code>skfuzzy.membership.gaussmf(x, c, σ)</code>	Função de pertinência gaussiana
<code>skfuzzy.membership.smf(x, a, b)</code>	Função de pertinência do tipo S
<code>skfuzzy.membership.zmf(x, a, b)</code>	Função de pertinência do tipo Z

Fonte: Autor (2023)

2.4.2 Sistemas *Fuzzy* do Tipo-2 com Python

O Python disponibiliza a biblioteca *PyIT2FLS* desenvolvida por Haghrh e Ghaemi (2019) para desenvolvimento de modelos *fuzzy* do tipo-2. Nos quadros a seguir são apresentados os comandos aplicados na criação do modelo.

Quadro 3 – Comandos da biblioteca *PyIT2FLS* para criação de antecedentes e consequentes

Comando	Descrição
<code>IT2FS(universo de discurso, tipo da UMF, coordenadas da UMF, tipo da LMF, coordenadas da LMF)</code>	Variável antecedente/consequente
<code>tri_mf</code>	Função de pertinência triangular
<code>trapezoid_mf</code>	Função de pertinência trapezoidal
<code>gbell_mf</code>	Função de pertinência sino
<code>gaussian_mf</code>	Função de pertinência gaussiana

Fonte: Autor (2023)

Dentro da biblioteca *PYIT2FLS* existe a classe Mamdani que possui comandos prontos para criação de um sistema *fuzzy* tipo-2 com método de inferência Mamdani. O Quadro 4 descreve os comandos necessários para criação do sistema.

Quadro 4 - Comandos da classe Mamdani

Comando	Descrição
IT2Mamdani(t_norm, s_norm, método de defuzzificação, algoritmo redutor de tipo)	Cria objeto de classe Mamdani
add_input_variable(antecedente)	Adiciona variável de entrada
add_input_variable(consequente)	Adiciona variável de saída
add_rule([antecedente],[consequente])	Adiciona regra no modelo
evaluate({vetor de entrada})	Avalia saída do modelo para determinado vetor de entrada

Fonte: Autor (2023)

2.5 MÉTRICAS DE AVALIAÇÃO

Nesta seção serão apresentados os cálculos por trás das métricas utilizadas para avaliação dos modelos.

2.5.1 Correlação de Pearson

A correlação de pearson é aplicada para validação do modelo para evidenciar a relação estatística entre os dados reais e os simulados pelo modelo. O cálculo da correlação é feito aplicando a equação (21).

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left[\sum_{i=1}^n (x_i - \bar{x})^2\right] \left[\sum_{i=1}^n (y_i - \bar{y})^2\right]}} \quad (21)$$

sendo n a quantidade de amostra, x_i e y_i as variáveis, $\bar{x}$ e $\bar{y}$ as médias das amostras.

O resultado deste cálculo indica o quão as variáveis estão linearmente relacionadas. Dessa forma, quanto mais esse valor se aproxima de 1, maior é a relação entre os dados analisados. Contudo, na revisão bibliométrica notou-se que os artigos que utilizam *fuzzy* para aplicações médicas apresentam correlação entre 0,2 e 0,4. Portanto, adotou-se essa faixa de valor como esperado para os modelos.

2.5.2 Erro Quadrático Médio (MSE)

O Erro Quadrático Médio (MSE, do inglês *mean squared error*) tem como intuito analisar a diferença média dos valores reais e simulados. O cálculo realizado é descrito na equação (22).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (22)$$

O MSE avalia a diferença quadrática média entre os valores simulados e os valores reais. O fato de o erro ser elevado ao quadrado traz as seguintes vantagens para a análise:

- Elimina diferenças com sinais negativos e, logo, ter-se-á sempre um resultado maior ou igual a zero. O ideal é que o MSE chegue a zero, mas na realidade isto nunca ocorre, pois é impossível criar um modelo sem erros.
- Elevar o erro ao quadrado penaliza valores que estão muito distantes do esperado. Logo, um alto valor de MSE indica resultados muito discrepantes.

2.5.3 Raiz Quadrática do Erro Médio (RMSE)

A Raiz Quadrática do Erro Médio (RMSE) mede qual é a diferença média entre o valor previsto e o valor real e seu cálculo é feito conforme mostrado na equação (23).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (23)$$

O benefício de usar essa métrica se dá ao fato de o valor ser apresentado na mesma unidade de medida que a variável de saída. Ou seja, ao dizer que o RMSE de um modelo é igual a 4 é equivalente ao dizer que em média os valores do modelo estão se distanciam do valor real em 4 unidades. Dessa forma, o RMSE facilita a interpretação da métrica de erro.

2.6 POLUENTES ATMOSFÉRICOS

Segundo o Conselho Nacional do Meio Ambiente (Conama) nº 491/2018 é considerado poluente atmosférico qualquer forma de matéria que por alguma característica (quantidade, concentração, tempo, etc) tenha a capacidade de tornar o ar impróprio ou nocivo à saúde, danosos à fauna e flora e prejudicial à segurança e ao bem-estar público. (BRASIL, 2018) A poluição atmosférica pode ser resultado de processos naturais ou atividade humana.

Os principais poluentes atmosféricos que mais causam danos à saúde são o material particulado (PM_{10}), dióxido de enxofre (SO_2), óxidos de nitrogênio (NO_x) e o ozônio (O_3) (CANÇADO et. al, 2006).

O PM_{10} é composto por uma mistura de partículas sólidas e líquidas com diâmetro inferior a $10\mu m$ proveniente da queima de combustíveis e biomassa, este material é capaz de atingir as vias respiratórias inferiores transportando gases adsorvidos em sua superfície. Essas partículas também são chamadas de partículas inaláveis, pois elas possuem a capacidade de adsorver gases e transportá-los até as vias respiratórias inferiores, onde ocorre a troca de gases no pulmão. O SO_2 é um gás resultante principalmente da queima de combustíveis que contém enxofre e é um dos principais formadores da chuva ácida. Os NO_x são agentes oxidantes repercutidos majoritariamente por motores de automóveis, o dióxido de nitrogênio (NO_2), devido a sua baixa solubilidade em água, quando inalado, atinge as porções mais periféricas do pulmão. A formação de O_3 é resultado de uma série de reações catalisadas pela luz solar e é um potente oxidante e citotóxico que atinge as porções mais distais das vias aéreas (CANÇADO et. al, 2006).

À medida que o mundo fica mais populoso, a emissão de poluentes atmosféricos por atividade humana aumenta de forma considerável e o ar respirado se torna cada vez mais nocivo à saúde, nove em cada dez pessoas respiram agora ar poluído, o que mata 7 milhões de pessoas todos os anos (OMS, 2018). O estudo da concentração de poluentes atmosféricos é, portanto, essencial para manutenção da saúde e bem-estar público.

3 MATERIAIS E MÉTODOS

São José dos Campos, com uma área de 1.099 km², é uma cidade localizada na região sudeste do Brasil entre as cidades de São Paulo e Rio de Janeiro (23° 11' S e 45° 53' W) e tem uma população de aproximadamente 700.000 habitantes e conta com 12 hospitais. A cidade é um polo industrial, comercial e de serviços que atende as cidades do leste de São Paulo e do sul de Minas Gerais, com uma população de aproximadamente dois milhões de habitantes. Importantes centros de pesquisa estão instalados no município, como o Instituto Nacional de Pesquisas Espaciais (INPE), o Instituto Tecnológico da Aeronáutica (ITA) e a Universidade Estadual Paulista (UNESP), com índice de desenvolvimento humano (IDH) de 0,81. A Rodovia Dutra, considerada a rodovia mais importante do Brasil, atravessa a cidade e tem um fluxo de aproximadamente 80.000 veículos por dia, incluindo grande quantidade de veículos pesados e ônibus (BRASIL, 2023).

Serão desenvolvidos modelos computacionais *fuzzy* tipo-1 e outro tipo-2 desenvolvidos com o objetivo de estimar o número de internações em crianças de até 10 anos por doenças respiratórias, por fim será avaliado a diferença de desempenho entre os dois modelos.

Para desenvolvimento dos modelos foram selecionadas, com auxílio de um profissional da área da saúde, quatro variáveis de entrada: concentração de PM₁₀ em µg/m³, concentração de NO₂ em µg/m³, temperatura mínima no dia em °C e velocidade do vento em m/s. Baseado nestes parâmetros os modelos *fuzzy* devem então ser capaz de prever o número de internações de crianças de 0 a 10 anos por doenças respiratórias no município de São José dos Campos.

Para construção das funções de pertinência dos modelos é primeiro necessário analisar como as variáveis de entrada se relacionam com a variável de saída. O trabalho foi então dividido em quatro etapas: coleta de dados, tratamento dos dados, desenvolvimento dos modelos *fuzzy* tipo-1 e tipo-2 e validação dos mesmos.

Os dados de entrada foram coletados da Companhia Ambiental do Estado de São Paulo (CETESB) através do Sistema de Informações da Qualidade do Ar (QUALAR). O número de internações de crianças por doenças respiratórias foi adquirido no TABWIN, aplicativo de tabulação de dados desenvolvido pelo DATASUS. Foram analisados dados de 2015 a 2019 para as doenças diagnósticas Laringite e Traqueíte (J04), Pneumonias (J12 a J18), Bronquite (J20), Bronquiolite (J21) e Asma (J45) segundo o CID – 10ª revisão.

3.1 BASE DE DADOS

Esta seção apresenta o passo a passo aplicado para coleta e limpeza da base de dados utilizada.

3.1.1 Cetesb

Para coletar as informações de poluentes atmosféricos e elementos climáticos primeiro é necessário acessar o sistema Qualar disponível do site da CETESB. Deve-se então clicar na opção ‘Consultas’ presente no canto esquerdo da tela e então selecionar a opção ‘Exportar Dados’, conforme mostrado na Figura 10.

Figura 10 – Acesso a base de dados de qualidade do ar

Fonte: Autor (2023)

É então aberta a tela mostrada na Figura 11, deve-se então inserir as datas inicial e final de quando dos dados a serem extraídos, assim como a estação e o parâmetro desejados. Para o trabalho em questão foram exportados dados de 01 de janeiro de 2015 a 31 de dezembro de 2019. Os parâmetros utilizados foram: MP₁₀ (Partículas Inaláveis), NO₂ (Dióxido de Nitrogênio), TEMP (Temperatura do Ar) e VV (Velocidade do Vento). São José dos Campos conta com três estações: S.José Campos, S.José Campos-Jd.Satelite e S.José Campos-Vista Verde

Figura 11 – Janela de seleção de dados de qualidade do ar a serem exportados

SISTEMA DE INFORMAÇÕES DA QUALIDADE DO AR V 3.83

[ESTAÇÃO](#)
[PARÂMETRO](#)
[ENTRADA DE DADOS](#)
[CONSULTAS](#)
[RELATÓRIOS](#)
[SAIR](#)

Processos - Exportar Dados Raquel de Oliveira Paula Andrade

EXPORTAR DADOS

Tipo de Rede: ☒ Automática ☐ Manual ☐ Passivo

Data Inicial*: 01/01/2015 **Data Final*:** 31/12/2019

Tipo de Dados: ☒ Média Horária ☐ Média Móvel

☐ Incluir Dados Inválidos

Estação*: S.José Campos

Parâmetro*: MP10 (Partículas Inaláveis)

Pesquisar

Fonte: Autor (2023)

Os dados são baixados e apresentados em média horária, dessa forma tem-se a média do parâmetro selecionado dentro de cada hora do dia (desde 01:00 até 24:00).

Para obter dados que sejam capazes de representar a média na cidade foram coletados os dados das três estações e feitas algumas aproximações. Foram extraídos os dados de média horária para as três estações de São José dos Campos e para as concentrações de PM₁₀, NO₂ e velocidade do vento foram calculadas as médias diárias para cada estação, para a temperatura mínima foi retirado o valor mínimo de cada dia. Ao fim, foi feita uma média do valor das três estações e o resultado foi adicionado a base de dados utilizada no desenvolvimento do trabalho.

3.1.2 Datasus

As internações de crianças por doenças respiratórias foram coletadas do Departamento de Informática do Sistema Único de Saúde (DATASUS) através do TABWIN, um aplicativo que possibilita a tabulação rápida a partir de dados do sistema de informação do SUS.

O primeiro passo para obtenção de dados de internação é realizar o download do TABWIN. Para tal, basta pesquisar a ferramenta no Google e será encontrada a tela ilustrada na Figura 12. Para realizar o download basta realizar seguintes seleções nos quadros disponíveis no site:

- Fonte: Aplicativos – TABWIN/TABNET – Ferramentas para tabulação de dados
- Modalidade: Programas
- Tipo de Arquivo: TABWIN – Tabulador de dados para Windows

Em seguida, clicar em enviar e o aplicativo estará disponível para download.

Figura 12 – Download do TABWIN

Download de arquivos

Fonte

Aplicativos - TABWIN/TABNET - Ferramentas para tabulação de dados
Base Populacional - IBGE
Base Territorial - Mapas e conversões para tabulação
CIH - Sistema de Comunicação de Informação Hospitalar

Modalidade

Documentação
Programas

Tipo de Arquivo

TABWIN - Tabulador de dados para Windows
TABNET - Tabulador de dados para ambiente internet
TABDOS - Tabulador de dados para DOS

Enviar

Fonte: Autor (2023)

O próximo passo consiste em baixar o arquivo com os dados de internações. Na mesma página do site do TABWIN deve-se selecionar as seguintes seleções.

- Fonte: SIHSUS – Sistemas de Informações Hospitalares do SUS
- Modalidade: Dados
- Tipo de Arquivo: RD - AIH Reduzida

Surgirão então novos quadros para seleção do ano, mês e unidade federativa (UF) da qual deseja-se obter os dados. Para o trabalho em questão foram selecionados todos os meses dos anos de 2015 a 2019 e São Paulo como UF.

Figura 13 – Download da base de dados de internações

Download de arquivos

Fonte

PO - Painel de Oncologia – desde 2013
RESP - Notificações de casos suspeitos de SCZ – desde 2015
SIASUS - Sistema de Informações Ambulatoriais do SUS
SIHSUS - Sistema de Informações Hospitalares do SUS
SIM - Sistema de Informações de Mortalidade

Modalidade

Arquivos auxiliares para tabulação
Dados
Documentação

Tipo de Arquivo

ER - AIH Rejeitadas com código de erro
RD - AIH Reduzida
RJ - AIH Rejeitadas
SP - Serviços Profissionais

Ano

2019
2018
2017
2016
2015

Mês

Abril
Maio
Junho
Julho
Agosto

UF

SC
SE
SP
TO

Enviar

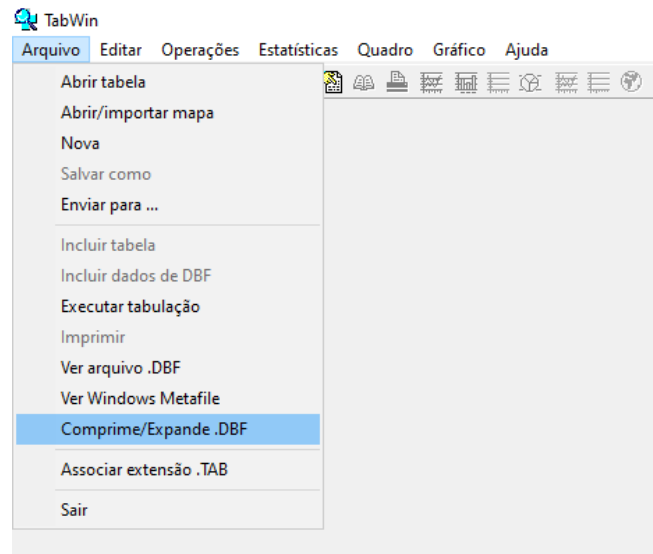
Fonte: Autor (2023)

Clicando em enviar o aplicativo estará disponível para download. A base de dados disponibilizada pelo DATASUS vem no formato dbc que corresponde a um arquivo dbf

compactado, para ler este arquivo em excel é preciso então descompactar o arquivo dbc utilizando o TABWIN.

Ao abrir o aplicativo TABWIN deve-se selecionar o menu Arquivo localizado na parte superior esquerda da tela e então clicar na opção “Comprime/Expande .DBF” conforme mostrado na Figura 14.

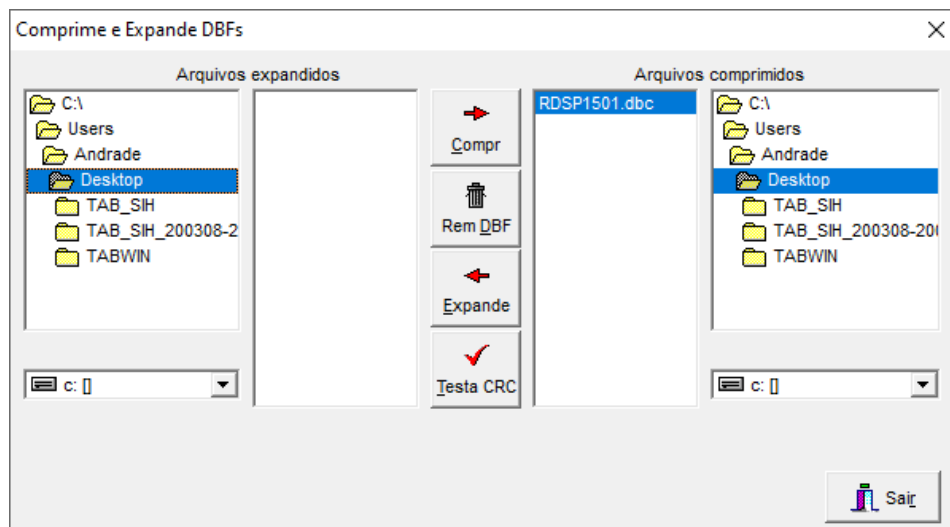
Figura 14 – Aplicativo TABWIN



Fonte: Autor (2023)

Ao finalizar os passos descritos anteriormente, será aberta a janela ilustrada na Figura 15. Na seção de “Arquivos Comprimidos” é preciso encontrar o arquivo dbc baixado e clicar em “Expande”, então o arquivo dbf será salvo na pasta selecionada na seção “Arquivos Expandidos”

Figura 15 – Expandindo arquivo dbc com o aplicativo TABWIN



Fonte: Autor (2023)

O arquivo dbf é então aberto no excel e tem-se acesso as informações de internações no estado de São Paulo no período de 01 de janeiro de 2015 a 31 de dezembro de 2019. A Figura 16 mostra como a base de dados é apresentada dentro do excel, sendo que cada linha corresponde a uma internação.

Figura 16 – Base da dados de internações baixada do DATASUS

	A	B	C	D	E	F	G	H	I	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD
1	UF	Z	AN		CGC_HOSP	N_AIH	CEP		MUNIC_RES	DT_INTER	DT_SAI	DIAG_PRINC	DI	NA	RUE												
2272	350000	2015	01	04	45915675000107	3508103544935	5	12230000	354990	20080101	20150131	F028	0000	28	61	3999	2	0	0	350870	4	30	31	0	010		
4081	350000	2015	01	05	45474863000139	3508103997750	5	12213010	354990	20080101	20150131	F200	0000	21	20	2062	2	0	0	353780	4	51	31	0	010		
4087	350000	2015	01	05	45474863000139	3508104000050	5	12235270	354990	20080101	20150131	F720	0000	21	20	2062	2	0	0	353780	4	58	31	0	010		
6022	350000	2015	01	03	46374500014810	3515100274851	1	12247210	354990	20150123	20150129	J81	0000	31	40	1023	2	0	0	350600	4	37	6	0	010		
9857	350000	2015	01	02		3515101326407	1	12239847	354990	20150109	20150111	O808	0000	61	50	1210	2	0	1	353620	4	30	2	0	010		
11873	350000	2015	01	05	46374500005153	3508103451754	5	1220540	354990	20080101	20150131	F205	0000	21	40	1023	2	0	0	351080	4	70	31	0	010		
13969	350000	2015	01	01	45186053000187	3515101528488	1	12209320	354990	20150105	20150109	S983	0000	12	60	3999	2	0	0	354990	4	54	4	0	010		
13977	350000	2015	01	03	45186053000187	3515101528587	1	12236170	354990	20150102	20150106	N189	0000	12	60	3999	2	0	0	354990	4	56	4	0	010		
13997	350000	2015	01	01	45186053000187	3515101528444	1	12237828	354990	20150105	20150114	T290	0000	12	60	3999	2	0	0	354990	4	25	9	0	010		
14004	350000	2015	01	03	45186053000187	3514124979332	1	12210030	354990	20141230	20150107	A498	0000	41	60	3999	2	0	0	354990	4	48	8	1	010		
14006	350000	2015	01	03	45186053000187	3515101528554	1	12227750	354990	20150101	20150103	K729	0000	12	60	3999	2	0	0	354990	4	45	2	0	010		
14008	350000	2015	01	07	45186053000187	3515101528895	1	12239060	354990	20150106	20150110	J210	0000	12	60	3999	2	0	0	354990	3	2	4	0	010		
14029	350000	2015	01	01	45186053000187	3514124979277	1	12214435	354990	20141225	20150108	T302	0000	12	60	3999	2	0	0	354990	4	56	14	0	010		
14030	350000	2015	01	01	45186053000187	3515101528433	1	12210110	354990	20141209	20141212	S720	0000	12	60	3999	2	0	0	354990	4	99	3	0	010		
14043	350000	2015	01	03	45186053000187	3515101528950	1	12227750	354990	20150107	20150117	K769	0000	12	60	3999	2	0	0	354990	4	45	10	0	010		
14045	350000	2015	01	03	45186053000187	3515101529247	1	12230020	354990	20150112	20150120	N189	0000	41	60	3999	2	0	0	354990	4	78	8	1	010		
14054	350000	2015	01	01	45186053000187	3515101529203	1	12239150	354990	20150109	20150116	T223	0000	12	60	3999	2	0	0	354990	4	24	7	0	010		
14058	350000	2015	01	01	45186053000187	3515101529346	1	12220200	354990	20150111	20150113	R100	0000	41	60	3999	2	0	0	354990	4	64	2	1	010		
14060	350000	2015	01	03	45186053000187	3515101528741	1	12232380	354990	20150106	20150109	N189	0000	12	60	3999	2	0	0	354990	4	76	3	0	010		

Fonte: Autor (2023)

A coluna da base de dados utilizadas para o desenvolvimento do trabalho foram:

- **MUNIC_RES**: indica o município de residência da pessoa internada. Foram filtradas as linhas com valor igual a 354990, que corresponde ao código IBGE para o município de São José Dos Campos;
- **DT_INTER**: data de internação;
- **DIAG_PRINC**: diagnóstico segundo o CID – 10ª revisão. Foram filtrados os códigos para Laringite e Traqueíte (J04), Pneumonias (J12 a J18), Bronquite (J20), Bronquiolite (J21) e Asma (J45);
- **IDADE**: foram utilizadas apenas as internações de crianças até 10 anos.

3.1.3 Tratamento dos dados

Após coletado todos os dados foi necessário entender como eles se comportavam e analisar quais informações seriam úteis para construção dos modelos. Esta etapa foi também feita com auxílio da linguagem de programação Python.

Primeiramente foram importadas três bibliotecas para auxiliar no processo de análise: *numpy*, a qual possui uma variedade de funções para manipular vetores; *matplotlib.pyplot*, utilizada para visualização e plotagem de gráficos; e *pandas*, amplamente aplicada para

trabalhar com dados tabulares. Foram então definidos os objetos np, plt e pd, respectivamente, para aplicação das bibliotecas em questão.

Quadro 5 – Bibliotecas usadas para análise de dados

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Fonte: Autor (2023)

A base de dados foi salva em um arquivo excel, para ler este arquivo foi utilizada a função *read_excel* da biblioteca pandas, o parâmetro passado para esta função é o caminho do arquivo excel. A tabela importada do arquivo é então armazenada no *Data Frame* dados. A função *dropna* é aplicada para eliminar as linhas que contém algum dos dados vazio.

Quadro 6 – Importando base de dados para análise

```
dados = pd.read_excel(r'F:\Caminho para a pasta\base_de_dados.xlsx')
dados.dropna(inplace = True)
```

Fonte: Autor (2023)

Um dos fatores que prejudicam a qualidade da interpretação de dados é a presença de *outliers*, valores atípicos que fogem da média, ou seja, que se afastam muito do comportamento padrão dos outros dados. Uma forma de visualizar a presença de outliers é através de *boxplot*, ferramenta gráfica que resume a distribuição gráfica de uma amostra. Foi então utilizada as funções presentes na biblioteca *matplotlib.pyplot* para visualizar esta distribuição.

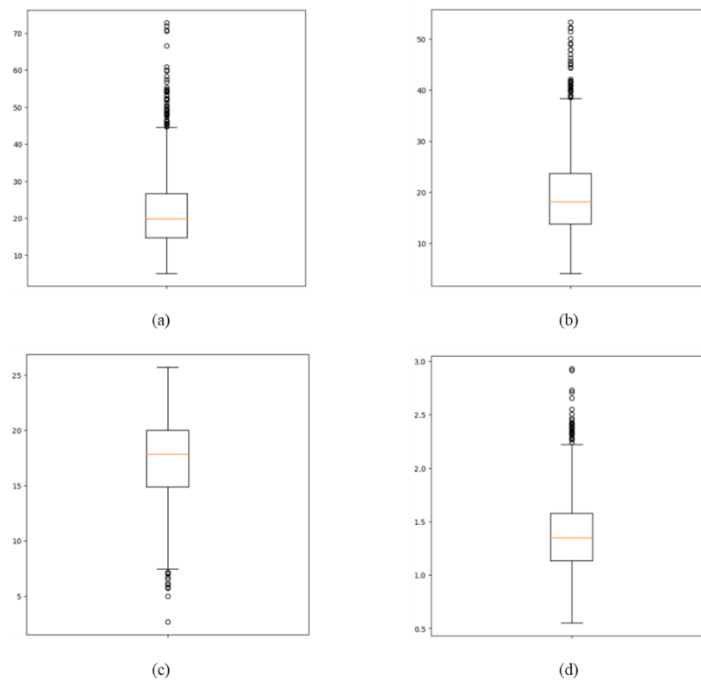
Quadro 7 – Plotando *boxplot* das variáveis de entrada

```
fig = plt.figure(figsize = (15,15))
fig.add_subplot(2,2,1)
plt.boxplot(dados['PM10'])
fig.add_subplot(2,2,2)
plt.boxplot(dados['NO2'])
fig.add_subplot(2,2,3)
plt.boxplot(dados['Temp. min'])
fig.add_subplot(2,2,4)
plt.boxplot(dados['VV'])
```

Fonte: Autor (2023)

Na figura abaixo tem-se o *boxplot* de cada uma das variáveis de entrada. A faixa laranja no meio representa o valor médio dos dados, a linha inferior do bloco indica o 1º quartil e o superior o 3º quartil, ou seja, a área delimitada pelo bloco mostra os valores que representam 50% dos dados do conjunto. Fora do bloco tem-se ainda dois limites, inferior e superior, e os pontos fora desta delimitação são os *outliers*.

Figura 17 – *Boxplot* das variáveis de entrada: (a) PM₁₀; (b) NO₂; (c) temperatura mínima; (d) velocidade do vento



Fonte: Autor (2023)

Pode-se observar que a base de dados coletada apresenta muitos *outliers* que podem ser provenientes de erros, como por exemplo erro no aparelho de medição. Para eliminar esses dados indesejados é necessário identificar os limites inferior e superior, o primeiro passo é encontrar o valor referente ao 1º quartil (q_1) e 3º quartil (q_3), então calcula-se intervalo interquartil (IIQ), que é variação entre q_1 e q_3 . Por fim, os limites inferior e superior são obtidos aplicando as equações (24) e (25), respectivamente.

$$\text{Limite Inferior} = q_1 - 1.5 \times \text{IIQ} \quad (24)$$

$$\text{Limite Superior} = q_3 + 1.5 \times \text{IIQ} \quad (25)$$

Por fim, elimina-se os valores que estão acima do limite superior e abaixo do limite inferior. O programa em utilizado para remover os *outliers* da variável PM₁₀ é apresentado no Quadro 8. O mesmo procedimento foi utilizado para as outras variáveis.

Quadro 8 – Retirando *outliers* da base de dados

```

q1 = dados["PM10"].quantile(.25)
q3 = dados["PM10"].quantile(.75)
IIQ = q3 - q1
limite_inferior = q1 - 1.5*IIQ
limite_superior = q3 + 1.5*IIQ

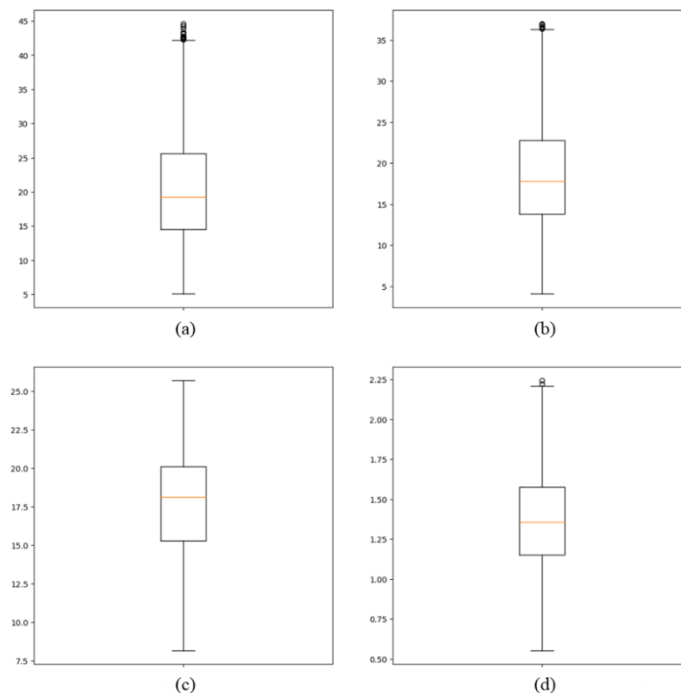
selecao = (dados["PM10"] >= limite_inferior) & (dados["PM10"] <= limite_superior)
dados_novo = pd.DataFrame()
dados_novo = dados[selecao]

```

Fonte: Autor (2023)

A Figura 18 apresenta o boxplot das variáveis de entrada após a exclusão dos *outliers*.

Figura 18 – *Boxplot* das variáveis de entrada após eliminar *outliers*: (a) PM₁₀; (b) NO₂; (c) temperatura mínima; (d) velocidade do vento



Fonte: Autor (2023)

3.2 MODELO FUZZY TIPO-1

Para desenvolvimento de qualquer programa em Python, primeiramente é necessário importar as bibliotecas que contém as funções que serão utilizadas. No caso do

desenvolvimento modelo *fuzzy* do tipo-1 primeiro é importada a biblioteca *skfuzzy* e sua classe *control*. Além disso, também são importadas as bibliotecas *pandas* e *numpy* para manipular os dados utilizados para validar o modelo. A função *mean_squared_error* da biblioteca *sklearn.metrics* foi importada para realização do cálculo de erro quadrático médio utilizado para validação do modelo.

Para facilitar o uso das funções de cada biblioteca, foram definidos os objetos *np*, *pd*, *fuzz* e *ctrl* para fazer referência às bibliotecas *numpy*, *pandas*, *skfuzzy* e classe *control*, respectivamente. O Quadro apresenta as linhas de código utilizadas para importar as bibliotecas utilizadas.

Quadro 9 – Bibliotecas importadas para o modelo *fuzzy* tipo-1

```
import numpy as np
import pandas as pd
import skfuzzy as fuzz
from skfuzzy import control as ctrl
from sklearn.metrics import mean_squared_error
```

Fonte: Autor (2023)

Em seguida são definidos os universos de discurso das variáveis de entrada (antecedentes) e de saída (consequente) utilizando a classe *control*, conforme mostrado no Quadro 1. O universo de discurso é delimitado considerando os valores mínimo e máximo de cada variável na base de dados coletada. A função *arange* da biblioteca *numpy* é utilizada para criar um arranjo de valores uniformemente espaçados, os dois primeiros parâmetros representam os valores mínimo e máximo do arranjo, enquanto que o terceiro parâmetro se refere ao intervalo entre os valores. Após determinar o intervalo que representa o universo de discurso, é estipulado quais serão os rótulos dos antecedente/consequente. No caso, *pm10* corresponde ao material particulado, *no2* ao dióxido de carbono, *temp* a temperatura mínima no dia, *vv* a velocidade do vento e *internacoes* ao número de internações em determinado dia. No Quadro 10 tem-se a parte do programa utilizada para a criação dos antecedentes e consequentes com seus respectivos universo de discurso.

Para cada variável foram definidas duas funções de pertinência Baixo (B) e Alto (A). Para o PM_{10} e NO_2 foi utilizada uma função de pertinência triangular para o subconjunto ‘B’ e uma função trapezoidal para o subconjunto A. Para o a temperatura mínima e velocidade do vento foram utilizadas duas funções trapezoidais.

No caso da variável de saída (número de internações) foram definidos 7 subconjuntos: Muito Baixo (MB), Baixo (B), Baixo Médio (BM), Médio (M), Alto Médio (AM), Alto (A) e Muito Alto (MA). Foi utilizada uma função de pertinência trapezoidal para MA e para o restante foram definidas funções de pertinência triangulares.

A função *automf* é utilizada para criar um conjunto *fuzzy* genérico com o número de subconjuntos definido no primeiro parâmetro, no segundo parâmetro tem-se a variável linguística pela qual cada subconjunto será chamado. Em seguida são utilizados os comandos mostrados no Quadro 2 para definir os parâmetros de cada função de pertinência e a qual universo de discurso cada um pertence. No Quadro 11 tem-se a definição das funções de pertinência.

Quadro 10 – Definição dos antecedentes e consequentes do modelo *fuzzy* tipo-1

```
pm10 = ctrl.Antecedent(np.arange(5,45,0.1),"pm10")
no2 = ctrl.Antecedent(np.arange(4,37,0.1),"no2")
temp = ctrl.Antecedent(np.arange(8,26,0.1),"temp")
vv = ctrl.Antecedent(np.arange(0.5,2.3,0.1),"vv")
internacoes = ctrl.Consequent(np.arange(0,13,0.1),"internacoes")
```

Fonte: Autor (2023)

Quadro 11 – Definição das funções de pertinência para o modelo *fuzzy* tipo-1

```
pm10.automf(2,names = ['B','A'])
pm10['B'] = fuzz.trapmf(pm10.universe, [0, 0, 5, 25])
pm10['A'] = fuzz.trapmf(pm10.universe, [5, 25, 45, 45])

no2.automf(2,names = ['B','A'])
no2['B'] = fuzz.trapmf(no2.universe, [0, 0, 4, 20])
no2['A'] = fuzz.trapmf(no2.universe, [4, 20, 37, 37])

temp.automf(2,names = ['B','A'])
temp['B'] = fuzz.trapmf(temp.universe, [0, 0, 17, 19])
temp['A'] = fuzz.trapmf(temp.universe, [17, 19, 26, 26])

vv.automf(2,names = ['B','A'])
vv['B'] = fuzz.trapmf(vv.universe, [0.5, 0.5, 0.6, 1.5])
```

Quadro 11 – Definição das funções de pertinência para o modelo *fuzzy* tipo-1

```

vv['A'] = fuzz.trapmf(vv.universe, [0.6, 1.5, 2.3, 2.3])

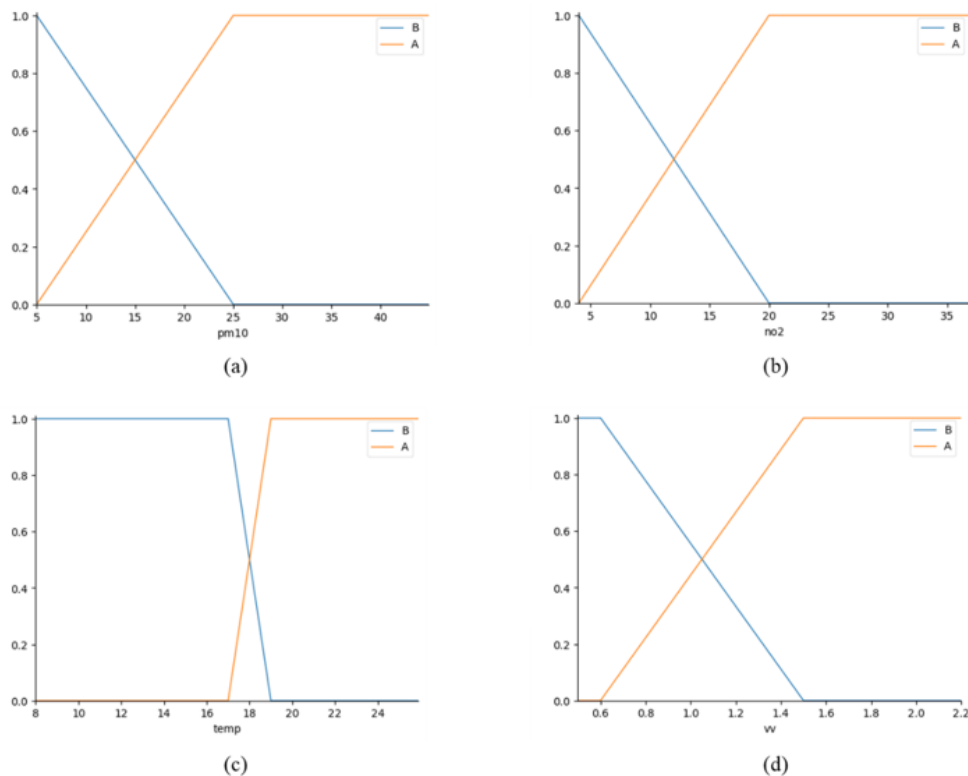
internacoes.automf(7, names = ['MB', 'B', 'BM', 'M', 'AM', 'A', 'MA'])
internacoes['MB'] = fuzz.trimf(internacoes.universe, [0, 0, 0.8])
internacoes['B'] = fuzz.trimf(internacoes.universe, [0.3, 1, 1.7])
internacoes['BM'] = fuzz.trimf(internacoes.universe, [1.5, 2, 2.5])
internacoes['M'] = fuzz.trimf(internacoes.universe, [2.3, 3, 3.7])
internacoes['AM'] = fuzz.trimf(internacoes.universe, [3.2, 4, 4.8])
internacoes['A'] = fuzz.trimf(internacoes.universe, [4, 5, 6])
internacoes['MA'] = fuzz.trapmf(internacoes.universe, [5, 6, 13, 13])

```

Fonte: Autor (2023)

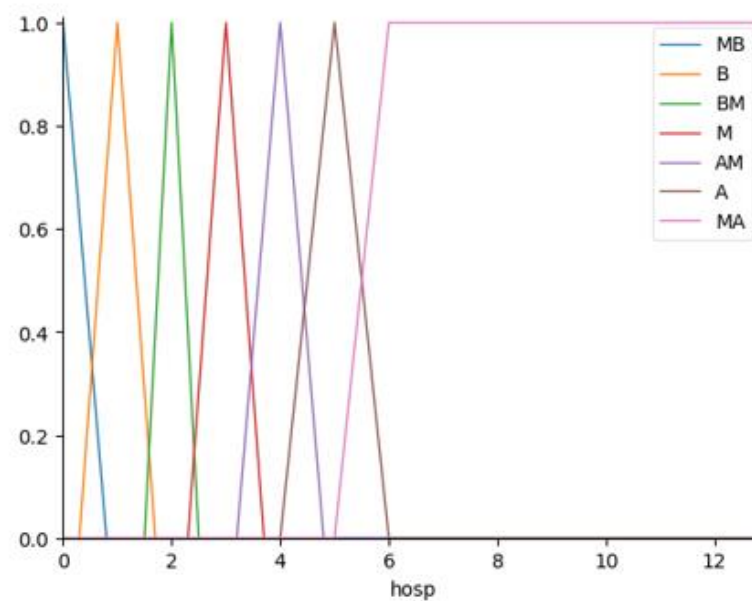
A Figura 19 ilustra como ficaram definidos os conjuntos *fuzzy* de entrada e na Figura 20 observa-se as funções de pertinência de saída.

Figura 19 – Funções de pertinência de entrada do modelo *fuzzy* tipo-1: (a) PM₁₀; (b) NO₂; (c) temperatura mínima; (d) velocidade do vento



Fonte: Autor (2023)

Figura 20 – Funções de pertinência de saída do modelo *fuzzy* tipo-1



Fonte: Autor (2023)

A próxima etapa para a construção do modelo é a definição da base de regras. A base de regras do modelo em questão conta com 16 regras (2x2x2x2) que foram determinadas com o auxílio de um especialista. Um médio com experiência neste ramo de pesquisa foi consultado para desenvolver a base de regras descrita no Quadro 12.

Quadro 12 – Base de regras para o modelo *fuzzy*

Regra	PM ₁₀	NO ₂	Temperatura Mínima	Velocidade do Vento	Número de Internações
Regra 1	B	B	B	B	B
Regra 2	B	B	B	A	B
Regra 3	B	B	A	B	MB
Regra 4	B	B	A	A	MB
Regra 5	B	A	B	B	M
Regra 6	B	A	B	A	BM
Regra 7	B	A	A	B	BM
Regra 8	B	A	A	A	MB
Regra 9	A	B	B	B	AM
Regra 10	A	B	B	A	M
Regra 11	A	B	A	B	M
Regra 12	A	B	A	A	BM
Regra 13	A	A	B	B	MA
Regra 14	A	A	B	A	A
Regra 15	A	A	A	B	AM
Regra 16	A	A	A	A	M

Fonte: Autor (2023)

A transcrição da base de regras em linguagem Python é apresentada no Quadro 13. A função *rule* presente na classe *control* é aplicada para criar cada regra que, por sua vez, são armazenadas nas variáveis *regra_n*. Para que uma determinada saída seja ativada, todas as condições de entrada devem ser satisfeitas, dessa forma, os antecedentes são separados por um operador E lógico (&). Após determinar a condição de ativação é inserida uma vírgula para separar o consequente dos antecedentes.

Após definir as regras, é necessário uni-las em uma única base, para tal é aplicada a função *ControlSystem* e a base de regras é armazenada na variável *base_regras*.

A função *ControlSystemSimulation* calcula o resultado da simulação do modelo composto pela base de regras definida.

Quadro 13 – Base de regras do modelo *fuzzy* tipo-1

```
regra1 = ctrl.Rule(pm10['B'] & no2['B'] & temp['B'] & vv['B'], internacoes['B'])
regra2 = ctrl.Rule(pm10['B'] & no2['B'] & temp['B'] & vv['A'], internacoes['B'])
regra3 = ctrl.Rule(pm10['B'] & no2['B'] & temp['A'] & vv['B'], internacoes['MB'])
regra4 = ctrl.Rule(pm10['B'] & no2['B'] & temp['A'] & vv['A'], internacoes['MB'])
regra5 = ctrl.Rule(pm10['B'] & no2['A'] & temp['B'] & vv['B'], internacoes['M'])
regra6 = ctrl.Rule(pm10['B'] & no2['A'] & temp['B'] & vv['A'], internacoes['BM'])
regra7 = ctrl.Rule(pm10['B'] & no2['A'] & temp['A'] & vv['B'], internacoes['BM'])
regra8 = ctrl.Rule(pm10['B'] & no2['A'] & temp['A'] & vv['A'], internacoes['MB'])
regra9 = ctrl.Rule(pm10['A'] & no2['B'] & temp['B'] & vv['B'], internacoes['AM'])
regra10 = ctrl.Rule(pm10['A'] & no2['B'] & temp['B'] & vv['A'], internacoes['M'])
regra11 = ctrl.Rule(pm10['A'] & no2['B'] & temp['A'] & vv['B'], internacoes['M'])
regra12 = ctrl.Rule(pm10['A'] & no2['B'] & temp['A'] & vv['A'], internacoes['BM'])
regra13 = ctrl.Rule(pm10['A'] & no2['A'] & temp['B'] & vv['B'], internacoes['MA'])
regra14 = ctrl.Rule(pm10['A'] & no2['A'] & temp['B'] & vv['A'], internacoes['A'])
regra15 = ctrl.Rule(pm10['A'] & no2['A'] & temp['A'] & vv['B'], internacoes['AM'])
regra16 = ctrl.Rule(pm10['A'] & no2['A'] & temp['A'] & vv['A'], internacoes['M'])
base_regras = ctrl.ControlSystem([regra1, regra2, regra3, regra4, regra5, regra6, regra7,
                                   regra8, regra9, regra10, regra11, regra12, regra13, regra14, regra15, regra16])

simulacao = ctrl.ControlSystemSimulation(base_regras)
```

Com o modelo já pronto, inicia-se o processo de validação do mesmo. Esse processo é feito aplicando os valores de entrada da base de dados coletada no modelo e observando a correlação e o erro quadrado médio entre o número de internações da base de dados e o resultado do modelo.

Conforme já feito anteriormente, utilizou-se a função *read_excel* da biblioteca *pandas* para importar a base de dados utilizada e armazená-la no *Data Frame* *dados*. Os vetores *pm10_valores*, *no2_valores*, *temp_valores*, e *vv_valores* tem a função de armazenar os valores das entradas que serão utilizadas na simulação. Estas operações são apresentadas no Quadro 14.

Quadro 14 – Preparando dados para simulação do modelo

```
dados = pd.read_excel(r'F:\Caminho para a pasta\base_de_dados.xlsx')
pm10_valores = dados['PM10'].to_numpy()
no2_valores = dados['NO2'].to_numpy()
temp_valores = dados['Temp. min'].to_numpy()
vv_valores = dados['VV'].to_numpy()
```

Fonte: Autor (2023)

Para armazenar os valores de saída da simulação foi criada a variável *saida_simulacao*. Em seguida, utiliza-se a estrutura de repetição *for* para percorrer cada valor dos vetores criados anteriormente e utiliza-los como entrada do modelo. Após atribuído o vetor de entrada, a função *compute* é utilizada para calcular o valor de saída. Este valor, por sua vez, é acrescentado no vetor *saida_simulacao*. Assim, cada linha deste vetor corresponde ao número de internações simulado para dado vetor de entrada. As linhas de código desenvolvidas para simulação do modelo são apresentadas no Quadro 15.

Quadro 15 – Simulação do modelo *fuzzy* tipo-1

```
saida_simulacao = []
for i in range(len(pm10_valores)):
    simulacao.input['pm10'] = pm10_valores[i]
    simulacao.input['no2'] = no2_valores[i]
    simulacao.input['temp'] = temp_valores[i]
    simulacao.input['vv'] = vv_valores[i]
    simulacao.compute()
    saida_simulacao.append(int(simulacao.output['internacoes']))
```

Fonte: Autor (2023)

Por fim, foi adicionada uma coluna na tabela dados com os valores simulados, conforme apresentado no Quadro 16. Dessa forma, é possível obter a correlação entre os dados reais e os simulados pelo modelo e o erro quadrático médio utilizando as funções *corr* e *mean_squared_error*, respectivamente.

Quadro 16 – Código para validação do modelo

```
dados['Internações Simulação'] = saida_simulacao
dados.corr().round(2)
mean_squared_error(dados['Internações'], dados['Internações Simulação']).round(2)
```

Fonte: Autor (2023)

3.3 MODELO FUZZY TIPO-2

O modelo *fuzzy* tipo-2 foi desenvolvido um modelo com incerteza na média adicionando uma mancha de incerteza nas funções de pertinência definidas para o modelo *fuzzy* tipo-1. Para o PM₁₀ foi adicionada uma mancha de 0,5 para a função B e 0,1 para a função A; no caso do NO₂ a mancha de incerteza é de 0,25 para B e 0,1 para A; para a temperatura mínima adicionou-se uma incerteza de 0,02 para as duas funções de pertinência; para a velocidade do vento a função B recebeu uma mancha de 0,2 e A de 0,1; na variável de saída (número de internações), as funções MB e B receberam uma mancha de 0,02, a mancha para A é de 0,08 e para o restante das funções de pertinência atribui-se uma mancha de 0,04.

Para desenvolvimento do modelo *fuzzy* tipo-2 foi importada da biblioteca *PYIT2FLS* as funções *IT2FS*, *tri_mf* e *trapezoid_mf* para definir as funções de pertinência. Também foi importado da biblioteca os operadores de mínimo e máximo (*min_t_norm* e *max_s_norm*, respectivamente), a função *IT2FS_plot* para plotar as funções de pertinência, a função *crisp* para transformar um valor *fuzzy* em um valor *crisp* e a classe *IT2Mamdani* para aplicação do método de inferência Mamdani.

Assim como feito no modelo *fuzzy* tipo-1, também foram utilizadas as bibliotecas *numpy*, *pandas* e *sklearn.metrics*. No Quadro 17 apresenta-se como foi feita a importação das bibliotecas utilizadas.

Quadro 17 – Bibliotecas importadas para o modelo *fuzzy* tipo-2

```

from pyit2fls import IT2Mamdani, IT2FS, tri_mf, trapezoid_mf, min_t_norm,
max_s_norm, crisp
import numpy as np
import pandas as pd
from sklearn.metrics import mean_squared_error

```

Fonte: Autor (2023)

Da mesma forma que foi feito para o modelo *fuzzy* tipo-1, também foi utilizada a função *arange* da biblioteca *numpy* para definir os universos de discurso de cada variável. Em seguida são criadas as funções de pertinência utilizando o comando *IT2FS* com os parâmetros citados no Quadro 3. A função *IT2FS_plot* é utilizada para visualizar os conjuntos *fuzzy* criados. O Quadro 18 apresenta a definição dos antecedentes e consequentes do modelo.

Quadro 18 - Definição dos antecedentes e consequentes do modelo *fuzzy* tipo-2

```

pm10      = np.arange(5,45,0.1)
no2       = np.arange(4,37,0.1)
temp      = np.arange(8,26,0.1)
vv        = np.arange(0.5,2.3,0.1)
internacoes = np.arange(0,13,0.1)

pm10_b = IT2FS(pm10, trapezoid_mf, [0, 1, 15, 35, 1], tri_mf, [0, 5, 15, 0.5])
pm10_a = IT2FS(pm10, trapezoid_mf, [3, 23, 45, 46, 1], trapezoid_mf, [7, 27, 45, 47, 0.9])
IT2FS_plot(pm10_b, pm10_a)

no2_b = IT2FS(no2, trapezoid_mf, [0, 1, 9.1, 24.1, 1], tri_mf, [0, 4, 15.9, 0.75])
no2_a = IT2FS(no2, trapezoid_mf, [2.35, 18.35, 37, 38, 1], trapezoid_mf, [5.65, 21.65, 37, 38, 0.9])
IT2FS_plot(no2_b, no2_a)

temp_b = IT2FS(temp, trapezoid_mf, [0, 1, 17.2, 19.2, 1], trapezoid_mf, [0, 1, 16.8, 18.8, 0.98])

```

Quadro 18 - Definição dos antecedentes e consequentes do modelo *fuzzy* tipo-2

```

temp_a = IT2FS(temp, trapezoid_mf, [16.8, 18.8, 26, 27, 1], trapezoid_mf, [17.2, 19.2, 26,
27, 0.98])
IT2FS_plot(temp_b, temp_a)

vv_b = IT2FS(vv, trapezoid_mf, [0.4, 0.5, 0.6, 1.5, 1], trapezoid_mf, [0.4, 0.5, 0.42, 1.2,
0.8])
vv_a = IT2FS(vv, trapezoid_mf, [0.51, 1.41, 2.3, 2.4, 1], trapezoid_mf, [0.69, 1.59, 2.3, 2.4,
0.9])
IT2FS_plot(vv_b, vv_a)

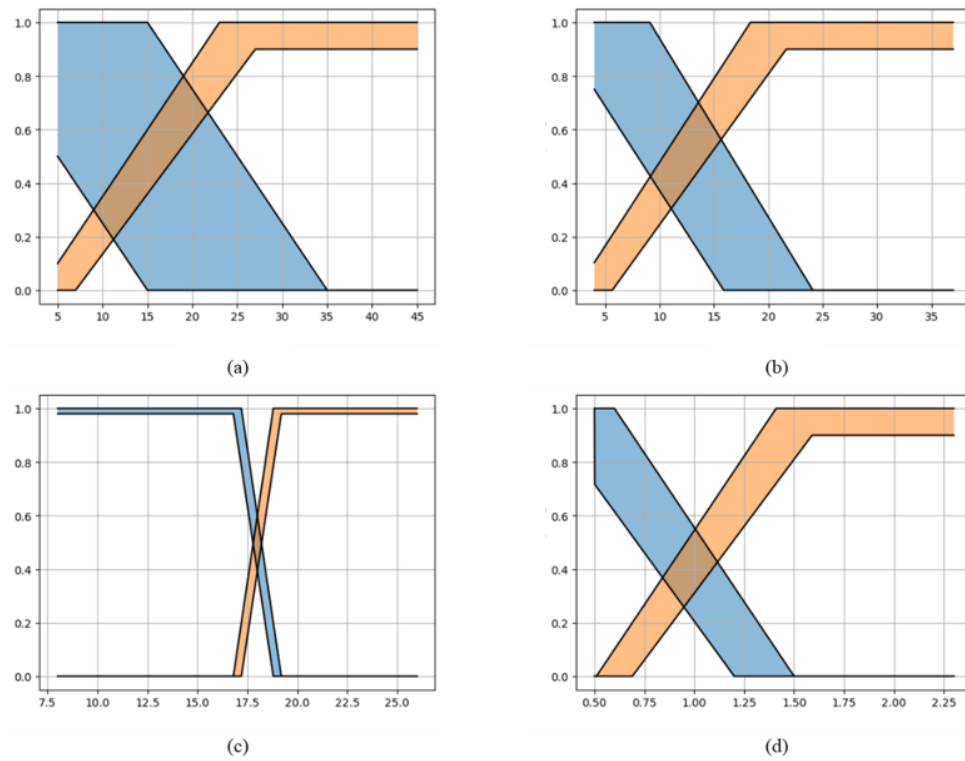
internacoes_mb = IT2FS(internacoes, trapezoid_mf, [-0.9, 0, 0.1, 0.9, 1], tri_mf, [-0.9, 0,
0.7, 0.98])
internacoes_b = IT2FS(internacoes, trapezoid_mf, [0.17, 0.87, 1.13, 1.83, 1], tri_mf, [0.33,
1, 1.57, 0.98])
internacoes_bm = IT2FS(internacoes, trapezoid_mf, [1.24, 1.74, 2.26, 2.76, 1], tri_mf,
[1.76, 2, 2.24, 0.96])
internacoes_m = IT2FS(internacoes, trapezoid_mf, [2, 2.7, 3.3, 4, 1], tri_mf, [2.6, 3, 3.4,
0.96])
internacoes_am = IT2FS(internacoes, trapezoid_mf, [2.9, 3.7, 4.3, 5, 1], tri_mf, [3.5, 4, 4.5,
0.96])
internacoes_a = IT2FS(internacoes, trapezoid_mf, [3.7, 4.7, 5.3, 6.3, 1], tri_mf, [4.3, 5, 5.7,
0.96])
internacoes_ma = IT2FS(internacoes, trapezoid_mf, [4.7, 5.7, 13, 14, 1], trapezoid_mf,
[5.3, 6.3, 13, 14, 0.92])
IT2FS_plot(internacoes_mb, internacoes_b, internacoes_bm, internacoes_m,
internacoes_am, internacoes_a, internacoes_ma)

```

Fonte: Autor (2023)

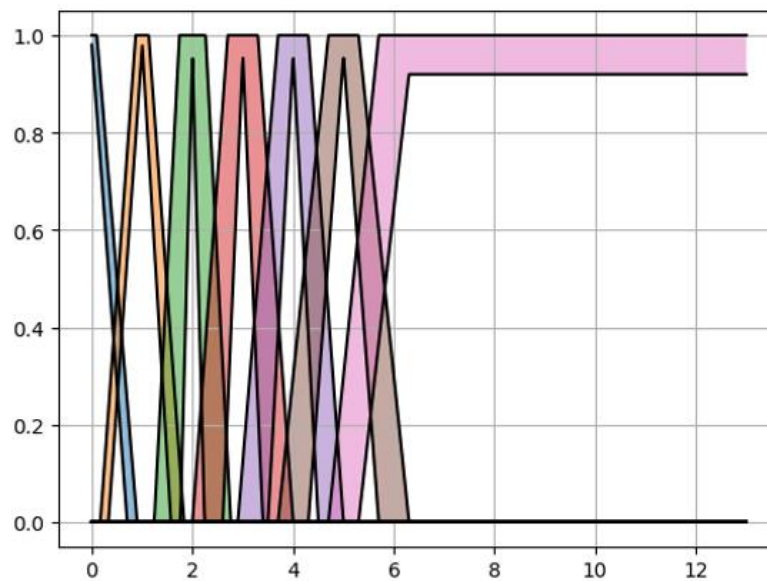
As Figuras 21 e 22 ilustram como ficaram as funções de pertinência de entrada e saída.

Figura 21 - Funções de pertinência de entrada do modelo *fuzzy* tipo-2: (a) PM₁₀; (b) NO₂; (c) temperatura mínima; (d) velocidade do vento



Fonte: Autor (2023)

Figura 22 – Funções de pertinência de saída do modelo *fuzzy* tipo-2



Fonte: Autor (2023)

Após definidas as funções de pertinência é feita a criação do sistema *fuzzy* tipo-2. Para criação de um modelo que utiliza o método de inferência Mamdani é criado um objeto da classe IT2Mamdani e são passados quatro parâmetros para o construtor desta classe: *t_norm*, que define qual operador da família t-norma será utilizado para intersecção, no caso é utilizado o mínimo (*min_t_norm*); *s_norm*, que especifica qual operador da família t-conorma será aplicado para fazer a união de conjuntos, no caso é utilizado o operador máximo (*max_s_norm*); *method*, que é um *string* que define o método de defuzzificação, para o trabalho em questão foi utilizado o método centroide; e *algorithm*, que determina o algoritmo que será usado para redução de tipo, o escolhido para o trabalho foi o algoritmo KM. O Quadro 19 mostra os parâmetros atribuídos passados para a função IT2Mamdani

Quadro 19 – Criação de um sistema *fuzzy* tipo-2 com método de inferência Mamdani

```
it2fls = IT2Mamdani(t_norm=min_t_norm, s_norm=max_s_norm,
                    method="Centroid", algorithm="KM")
```

Fonte: Autor (2023)

Em seguida, são adicionadas ao modelo as variáveis de entrada e de saída. As funções *add_input_variable* e *add_output_variabel* da classe Mamdani recebem uma *string* que corresponde a variável *fuzzy* que será adicionada como entrada ou saída. No Quadro 20 observa-se como foram adicionadas no modelo as variáveis de entrada e saída.

Quadro 20 – Adicionando as variáveis de entrada e saída no modelo *fuzzy* tipo-2

```
it2fls.add_input_variable("pm10")
it2fls.add_input_variable("no2")
it2fls.add_input_variable("temp")
it2fls.add_input_variable("vv")
it2fls.add_output_variable("internacoes")
```

Fonte: Autor (2023)

Para criação da base de regras é utilizada a função *add_rule*, os parâmetros passados para esta função são uma lista com os antecedentes e uma lista com os consequentes, as listas são delimitadas por colchetes. As regras aplicadas para o modelo *fuzzy* tipo-2 são as mesmas definidas no Quadro 12. O Quadro 21 apresenta o código aplicado para desenvolvimento da base de regras do modelo.

Quadro 21 – Criando base de regras do modelo *fuzzy* tipo-2

```

it2fls.add_rule([("pm10", pm10_b), ("no2", no2_b), ("temp", temp_b), ("vv", vv_b)],
[("internacoes", internacoes_b)])
it2fls.add_rule([("pm10", pm10_b), ("no2", no2_b), ("temp", temp_b), ("vv", vv_a)],
[("internacoes", internacoes_b)])
it2fls.add_rule([("pm10", pm10_b), ("no2", no2_a), ("temp", temp_b), ("vv", vv_b)],
[("internacoes", internacoes_mb)])
it2fls.add_rule([("pm10", pm10_b), ("no2", no2_b), ("temp", temp_a), ("vv", vv_a)],
[("internacoes", internacoes_mb)])
it2fls.add_rule([("pm10", pm10_b), ("no2", no2_a), ("temp", temp_b), ("vv", vv_b)],
[("internacoes", internacoes_m)])
it2fls.add_rule([("pm10", pm10_b), ("no2", no2_a), ("temp", temp_b), ("vv", vv_a)],
[("internacoes", internacoes_bm)])
it2fls.add_rule([("pm10", pm10_b), ("no2", no2_a), ("temp", temp_a), ("vv", vv_b)],
[("internacoes", internacoes_bm)])
it2fls.add_rule([("pm10", pm10_b), ("no2", no2_a), ("temp", temp_a), ("vv", vv_a)],
[("internacoes", internacoes_mb)])

it2fls.add_rule([("pm10", pm10_a), ("no2", no2_b), ("temp", temp_b), ("vv", vv_b)],
[("internacoes", internacoes_am)])
it2fls.add_rule([("pm10", pm10_a), ("no2", no2_b), ("temp", temp_b), ("vv", vv_a)],
[("internacoes", internacoes_m)])
it2fls.add_rule([("pm10", pm10_a), ("no2", no2_b), ("temp", temp_a), ("vv", vv_b)],
[("internacoes", internacoes_m)])
it2fls.add_rule([("pm10", pm10_a), ("no2", no2_b), ("temp", temp_a), ("vv", vv_a)],
[("internacoes", internacoes_bm)])
it2fls.add_rule([("pm10", pm10_a), ("no2", no2_a), ("temp", temp_b), ("vv", vv_b)],
[("internacoes", internacoes_ma)])
it2fls.add_rule([("pm10", pm10_a), ("no2", no2_a), ("temp", temp_b), ("vv", vv_a)],
[("internacoes", internacoes_a)])
it2fls.add_rule([("pm10", pm10_a), ("no2", no2_a), ("temp", temp_a), ("vv", vv_b)],
[("internacoes", internacoes_am)])

```

Quadro 21 – Criando base de regras do modelo *fuzzy* tipo-2

```
it2fls.add_rule([("pm10", pm10_a), ("no2", no2_a), ("temp", temp_a), ("vv", vv_a)],
[("internacoes", internacoes_m)])
```

Fonte: Autor (2023)

A validação do modelo foi feita de forma análoga ao modelo *fuzzy* tipo-1. Para importar a base de dados que contém os valores reais foi utilizado o código apresentado no Quadro 14. No Quadro 22 tem-se as linhas de programa aplicadas para simular o modelo criado utilizando os vetores de entrada presentes na base de dados.

A função *evaluate* utilizada para simular o modelo recebe como parâmetro uma variável do tipo dicionário (variável no formato chave: valor) cuja as chaves são os nomes das variáveis de entrada definidas para o modelo e o valor é o valor de entrada a ser avaliado. Utilizando a função *evaluate* com um modelo que utiliza centroide método de defuzzificação, serão retornados dois valores: *it2out*, que é o conjunto de saída em *fuzzy* tipo-2; e *tr*, que é o intervalo de tipo reduzido. Para calcular o valor real de saída é utilizada a função *crisp* e este valor é então armazenado no *array* *saida_simulacao*.

Quadro 22 – Simulação do modelo *fuzzy* tipo-2

```
saida_simulacao = []

for i in range(len(pm10_valores)):
    it2out, tr = it2fls.evaluate({"pm10":pm10_valores[i], "no2":no2_valores[i],
    "temp":temp_valores[i], "vv":vv_valores[i]})
    saida_simulacao.append(int(crisp(tr["internacoes"])))
```

Fonte: Autor (2023)

Para calcular verificar o desempenho do modelo utilizando o erro quadrático médio, foi utilizado o código apresentado no Quadro 16.

4 RESULTADOS

Dentro dos dados analisados totalizou-se 3674 internações de crianças na cidade de São José dos Campos com as doenças respiratórias correspondente aos CID mencionados anteriormente. O intervalo das variáveis (máximos e mínimos), assim como a média e o desvio padrão são mostrados na Tabela 2.

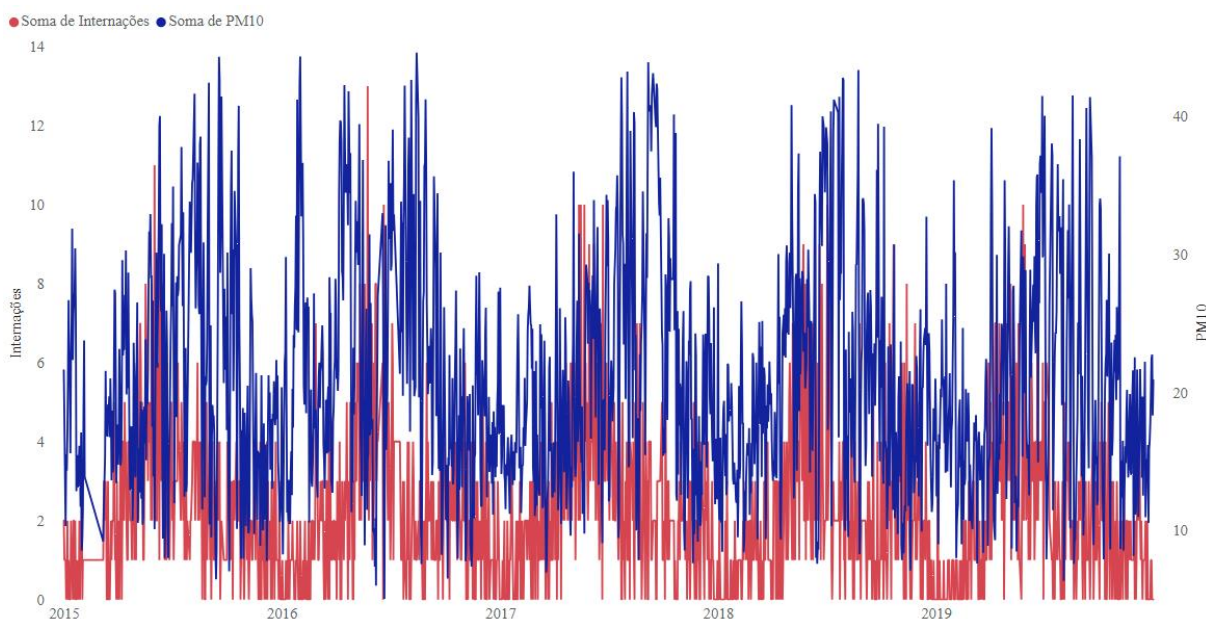
Tabela 2 - Valores médios, mínimos e máximos com os respectivos desvios padrão, das variáveis PM₁₀, NO₂, temperatura mínima, velocidade do vento e número de internações. São José dos Campos, 2015-2019.

Variável	Média	Desvio Padrão	Mínimo	Máximo
PM ₁₀ (µg/m ³)	20,74	8,13	5,04	44,58
NO ₂ (µg/m ³)	18,59	6,67	4,04	36,98
Temperatura mínima (°C)	17,48	3,21	8,15	25,7
Velocidade do vento (m/s ²)	1,37	0,32	0,55	2,24
Número de internações	2,21	1,95	0	13

Fonte: Autor (2023)

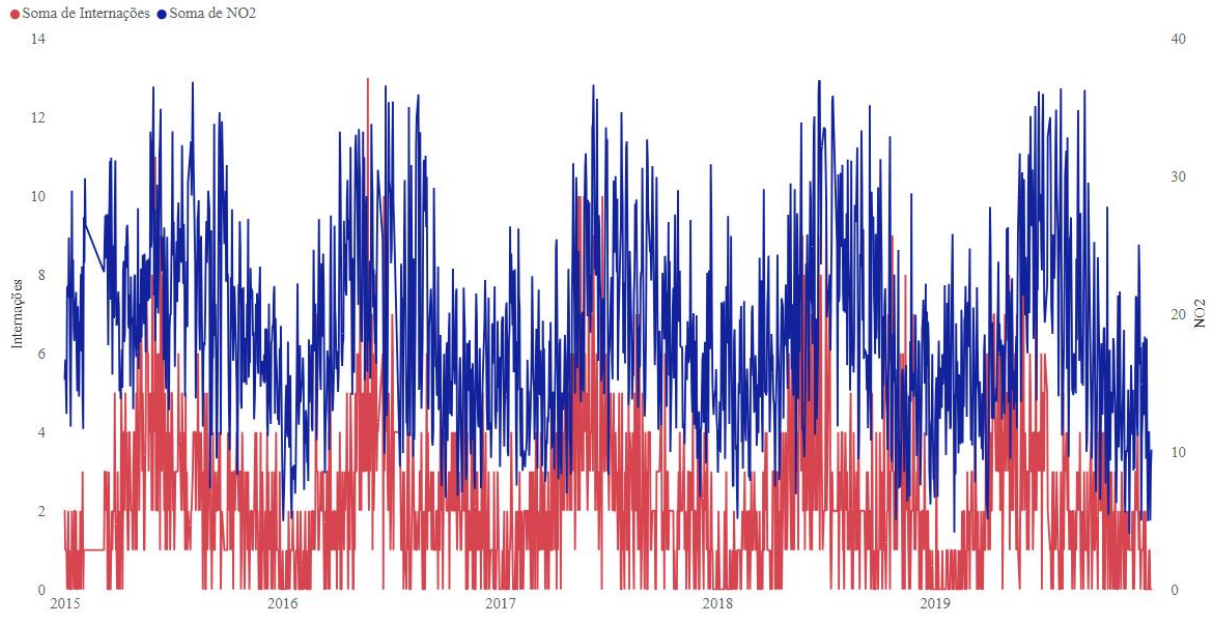
As Figura 23 até a 26 mostram a distribuição dos dados obtidos, cada variável de entrada foi plotada no mesmo gráfico que a variável de saída para visualizar possíveis correlações.

Figura 23 – Distribuição dos valores das variáveis PM₁₀ e número de internações por dia



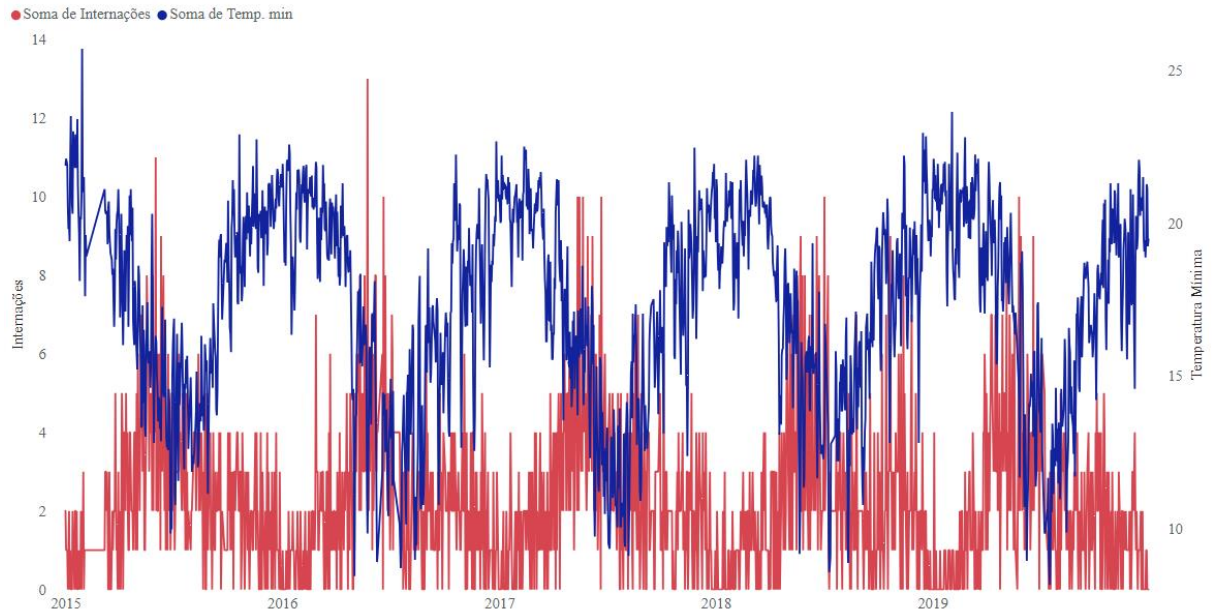
Fonte: Autor (2023)

Figura 24 – Distribuição dos valores das variáveis NO₂ e número de internações por dia



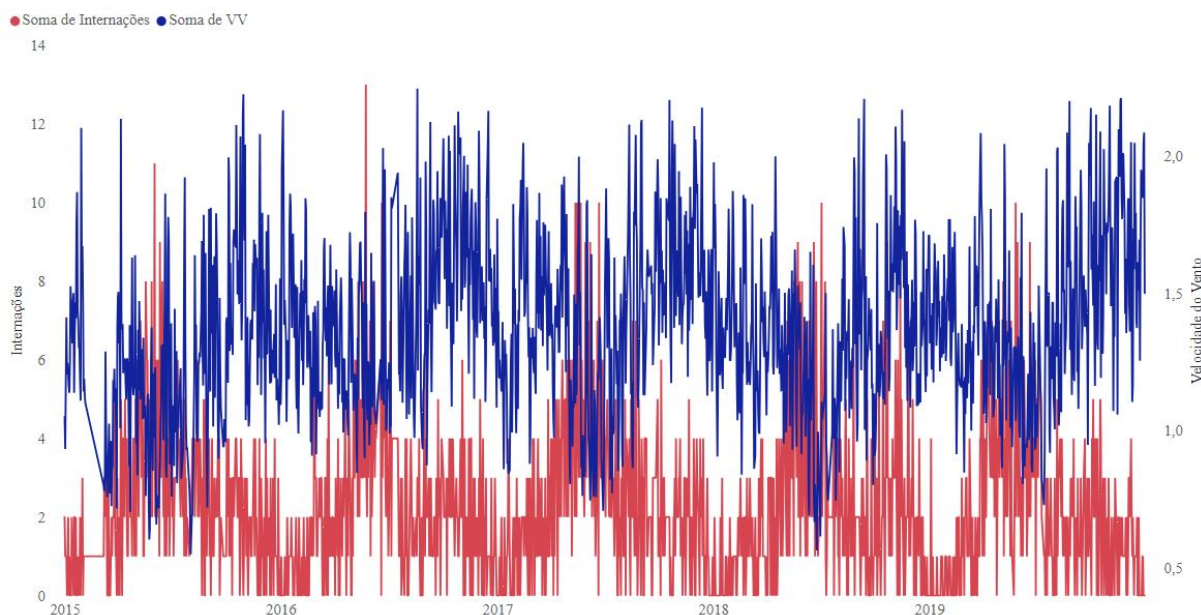
Fonte: Autor (2023)

Figura 25 – Distribuição dos valores das variáveis temperatura mínima e número de internações por dia



Fonte: Autor (2023)

Figura 26 – Distribuição dos valores das variáveis velocidade do vento e número de internações por dia



Fonte: Autor (2023)

No período de junho a setembro tem-se as maiores concentrações de PM_{10} e NO_2 em decorrência da baixa umidade e da inversão térmica ocasionada pelo inverno. Estabelecendo uma comparação entre os gráficos apresentados nota-se que o número de internações por doenças respiratórias é diretamente proporcional à concentração dos poluentes estudados, o contrário ocorre para a temperatura mínima e a velocidade do vento.

A primeira versão do modelo fuzzy tipo-2 foi desenvolvida adicionando uma mancha de incerteza de 0,1 para todas as funções de pertinência. Para comparar o desempenho dos sistemas fuzzy tipo-1 e tipo-2 foi calculada a correlação entre os valores reais e os gerados pelo modelo, o erro quadrático médio (MSE) e a raiz quadrada do erro médio (RMSE) de cada modelo.

Tabela 3 - Comparação entre os modelos fuzzy tipo-1 e tipo-2 com FOU de 0,1

Modelo	Correlação	MSE	RMSE
Fuzzy tipo-1	0,31	9,33	3,05
Fuzzy tipo-2	0,35	9,56	3,09

Fonte: Autor (2023)

Observou-se então uma melhora na correlação, mas o MSE apresentou um aumento de 2,47%. Para diminuir este valor, foi feita uma análise dos dados que apresentaram maior diferença entre o valor real e o simulado e foram constatados os seguintes pontos:

- Para PM_{10} foi observado um maior erro quando a concentração do poluente é baixa, evidenciando uma maior incerteza entre relação entre PM_{10} e o número de internações

para estes casos, por isso, aumentou-se o valor da mancha de incerteza da função B e foi observado um melhor resultado para um FOU de 0,5.

- A mesma constatação foi feita para o NO₂ e velocidade do vento, variando a função de pertinência B foi notado um melhor resultado para FOU de 0,15 e 0,2 para NO₂ e velocidade do vento, respectivamente.
- Para temperatura mínima notou-se uma correlação mais estreita com o número de interações. Ao fim, chegou-se a uma FOU de 0,02.

A Tabela 4 apresenta as métricas de validação após ajustadas todas as manchas de incerteza.

Tabela 4 - Comparação entre os modelos *fuzzy* tipo-1 e tipo-2

Modelo	Correlação	MSE	RMSE
<i>Fuzzy</i> tipo-1	0,31	9,33	3,05
<i>Fuzzy</i> tipo-2	0,35	7,31	2,70

Fonte: Autor (2023)

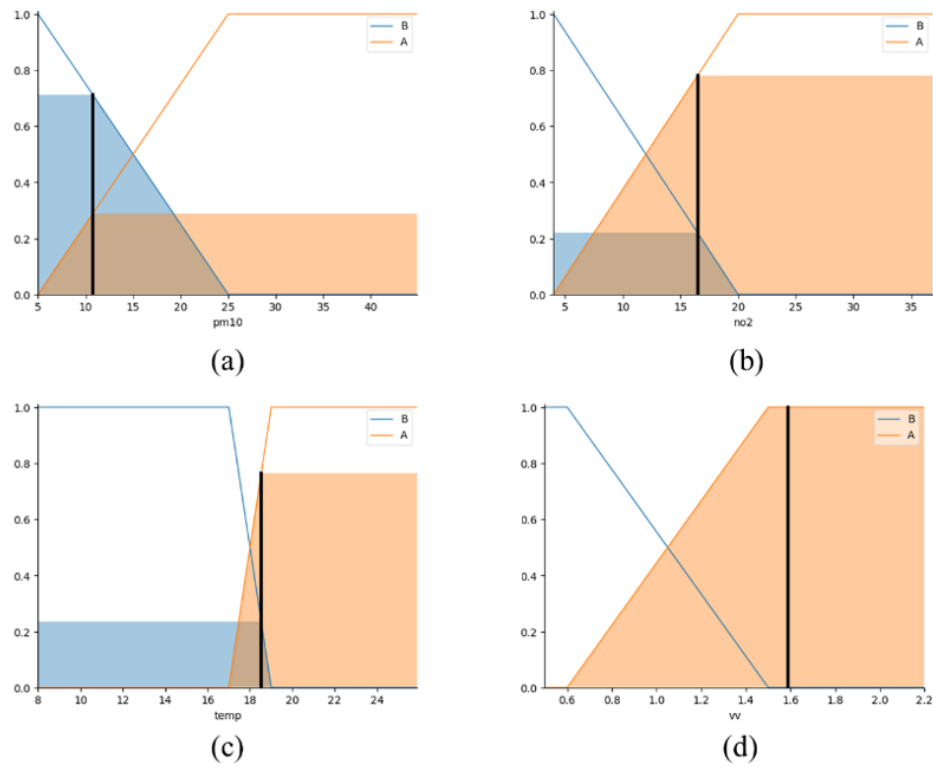
Após ajustadas as manchas de incerteza, o modelo *fuzzy* tipo-2 apresentou melhores resultados, em relação ao *fuzzy* tipo-1, para ambas as métricas de avaliação calculadas.

4.1 SIMULAÇÃO DO MODELO

As Figuras a seguir ilustram uma simulação do modelo *fuzzy* tipo-1 para os seguintes valores de entrada: PM₁₀ = 10,77 µg/m³; NO₂ = 16,48 µg/m³; Temperatura mínima = 18,54 °C; Velocidade do Vento = 1,59 m/s.

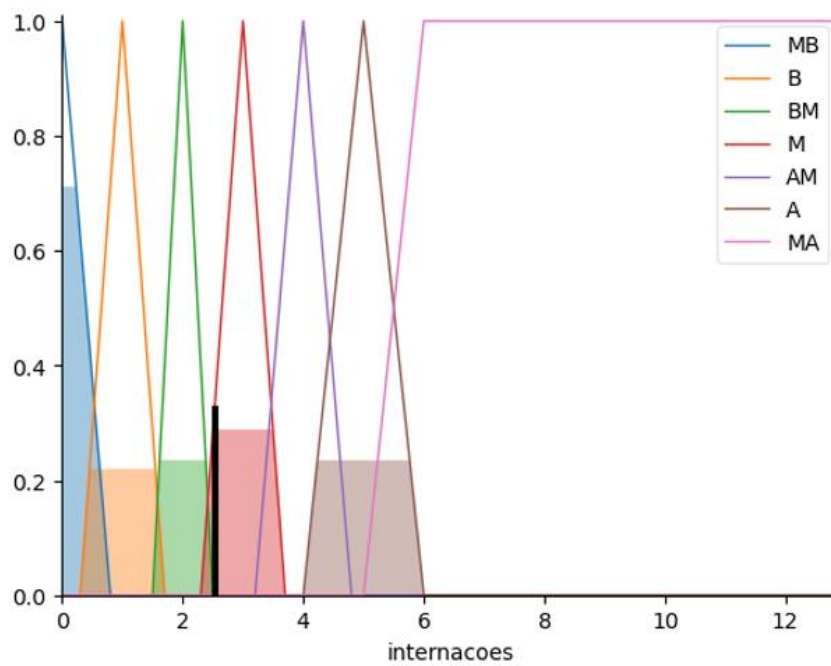
Esses valores de entrada foram retirados de uma linha da base de dados e o número de interações para este dia foi igual a 2. Pela Figura 28 é possível observar quais as funções de pertinência de saída foram ativadas e a linha preta indica a saída *crisp* do modelo (centroide), para esse caso o modelo apresenta uma boa resposta com valor bem próximo ao esperado.

Figura 27 – Exemplo de ativação das funções de pertinência de entrada do modelo *fuzzy* tipo-1: (a) PM₁₀; (b) NO₂; (c) temperatura mínima; (d) velocidade do vento



Fonte: Autor (2023)

Figura 28 – Exemplo de ativação das funções de pertinência de saída do modelo *fuzzy* tipo-1

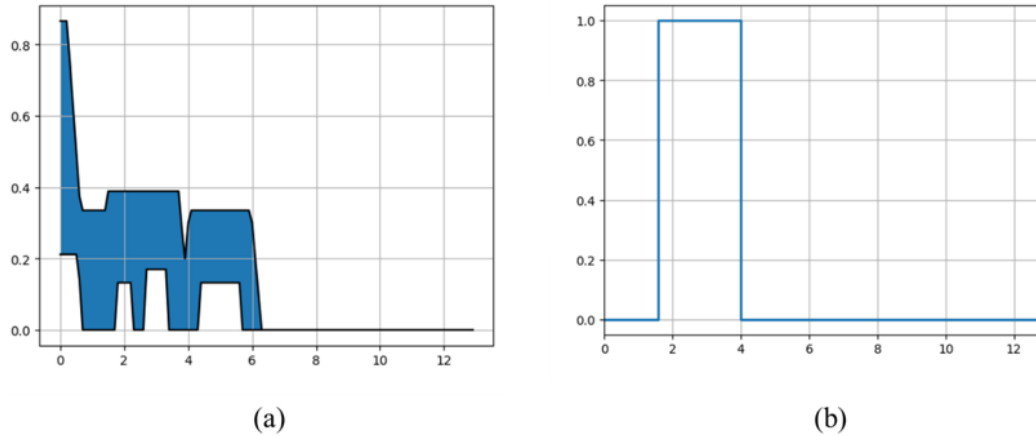


Fonte: Autor (2023)

A Figura 29 apresenta a saída do modelo *fuzzy* tipo-2 quando aplicada as mesmas variáveis de entrada usadas no exemplo de aplicação do modelo *fuzzy* tipo-1.

Figura 29 – Exemplo de ativação das funções de pertinência de saída do modelo *fuzzy* tipo-2:

(a) saída *fuzzy* tipo-2; (b) saída com tipo reduzido



Fonte: Autor (2023)

A saída para o modelo *fuzzy* tipo-2, conforme descrito na equação (20), seria a média entre os valores de y_r e y_l . Pela saída com tipo reduzido verifica-se que, para os valores de entrada aplicados, y_r assume um valor um pouco menor que 2 e y_l é igual a 4. Logo, a saída simulada seria também um valor próximo de 2.

5 CONSIDERAÇÕES FINAIS

O trabalho descrito apresenta uma abordagem baseada em lógica *fuzzy* para desenvolver um modelo capaz de correlacionar os efeitos da poluição do ar e saúde infantil. Com esse estudo, foram coletadas informações de número de internações de crianças por doenças respiratórias de diagnósticos de J04, J12 a J18, J20, J21 e J45 segundo o CID – 10ª revisão na cidade de São José dos Campos (SP). Também foram analisados relatórios de concentração dos poluentes PM10 e NO2, assim como a temperatura mínima e velocidade do vento no período de 2015 a 2019.

Foram desenvolvidos modelos utilizando lógica *fuzzy* tipo-1 e tipo2, relacionando as variáveis de entrada: PM₁₀, NO₂, temperatura mínima e velocidade do vento com a variável de saída, número de internações de crianças por doenças respiratórias. O modelo desenvolvido utilizando lógica *fuzzy* tipo-2 apresentou melhor correlação ($r=0,35$), menor erro quadrático médio (MSE = 7,31) e raiz quadrática do erro médio (RMSE = 2,70) .

A medicina é um campo de estudo repleto de incertezas, apesar de ser possível observar uma relação entre os poluentes e fatores atmosféricos utilizados como entrada e o número de internação por doenças respiratórias, ainda há fatores que não são contemplados na base de dados analisada, tais como possíveis erros de diagnóstico. O número de internações obtidos através do DATASUS não contempla internações de plano de saúde privado, assim não se tem o número real de internações. Além disso, também existe uma certa incerteza com relação a veracidade dos diagnósticos.

As centrais de monitoramento das quais foram extraídos os dados das condições atmosféricas e a concentração de poluentes realizam medições em regiões específicas da cidade, dessa forma ao utilizar esses dados considera-se que essas condições são homogêneas, o que não corresponde ao cenário real.

A presença de tais incertezas justifica a escolha de criação de um modelo preditivo utilizando lógica *fuzzy*. O modelo *fuzzy* tipo-1 apresentou um erro quadrático médio de 9,33 indicando um certo afastamento do modelo dos dados reais. Esse erro foi diminuído em 22% quando adicionadas manchas de incerteza nas funções de pertinência. Com esta avaliação é possível notar como a lógica *fuzzy* se enquadra como uma boa alternativa para solução de problemas complexos. Mediante a uma base de dados que apresentam informações ambíguas e passíveis de erro, a adição de graus de incerteza possibilita a criação de modelos que se aproximem cada vez mais da realidade.

Apesar da otimização do modelo após a implementação da mancha de incerteza prevista pelo *fuzzy* tipo-2, sistemas preditivos baseados em lógica *fuzzy* possuem suas limitações. Um dos fatores limitantes é a subjetividade presente na definição de regras e funções de pertinência, onde não existe uma metodologia sistemática para definição destes aspectos. A base de regras é especificada segundo a análise dos profissionais envolvidos no desenvolvimento do sistema e podem variar segundo a interpretação de cada um. Uma alternativa para melhorar a acurácia do modelo é a aplicação de algoritmos de aprendizagem, como *neuro-fuzzy*.

Mesmo com os fatores limitantes, o trabalho desenvolvido se mostra viável, pois apresenta baixo custo de desenvolvimento e pode ser aplicado em qualquer cidade que apresente pontos de coleta de fatores climáticos e poluentes presentes no ar. Os modelos descritos *fuzzy* tipo-1 e tipo-2 descritos neste trabalho foram anteriormente desenvolvidos utilizando o software MATLAB, porém o alto custo desta ferramenta talvez torne a aplicação inviável para algumas localidades. O modelo desenvolvido em Python representa uma alternativa de aplicação sem custo.

O modelo elaborado poderá ser utilizado na gestão hospitalar, pois com a estimativa do número de hospitalização será possível preparar todos os recursos hospitalares e médicos necessários para que os pacientes sejam atendidos com prontidão.

REFERÊNCIAS

BRASIL. Conama Resolução nº 491 de 2018. **Diário oficial da união**. Brasília, DF, 21 nov. 2018. Seção 1, p. 155.

BRASIL. Instituto Brasileiro de Geografia e Estatística. **São José dos Campos**. Rio de Janeiro: IBGE, 2023. Disponível em: <https://cidades.ibge.gov.br/brasil/sp/sao-jose-dos-campos/panorama> . Acesso em: 01 mar. 2023.

BRASIL. Ministério da Saúde. DATASUS, [S.d.]. Disponível em: <http://www2.datasus.gov.br/DATASUS/index.php>. Acesso em: 20 de maio de 2023.

BRASIL. Ministério da Saúde. **TABWIN**. DATASUS, [S.d.]. Disponível em: <https://datasus.saude.gov.br/transferecia-de-arquivos/>. Acesso em: 11 nov. 2023.

CANÇADO J. E. D; *et al.* Repercussões clínicas da exposição à poluição atmosférica. **Jornal brasileiro de pneumologia**, Brasília, v. 32, Supl. 1, S5-S11, 2006.

CARVALHO, W.F. *et al.* **Dispersão de poluentes na cidade de São José dos Campos**. São José dos Campos: INPE, 2004. Disponível em: http://mtc-m16b.sid.inpe.br/col/cptec.inpe.br/walmeida/2004/06.14.10.03/doc/Carvalho_Dispersao%20dos%20poluentes%20na%20cidade%20de%20Sao%20Jose%20dos%20Campos.pdf. Acesso em: 01 jul. 2023

CASTILLO, O.; MELIN, P. **Type-2 Fuzzy logic: A theory and applications**. Alemanha: Springer, 2008. ISBN 978-3642095139.

FÓRUM INTERNACIONAL DE SOCIEDADES RESPIRATÓRIAS. **O impacto global da doença respiratória - Segunda edição**. Mexico: Associação latino-americana do tórax, 2017. Disponível em: https://theunion.org/sites/default/files/2020-08/The_Global_Impact_of_Respiratory_Disease_port.pdf. Acesso em: 01 mar. 2023

SÃO PAULO (Estado). Companhia Ambiental do Estado de São Paulo. **Qualar – Sistema de informações da qualidade do ar**. São Paulo: CETESB, [S.d.]. Disponível em: <https://cetesb.sp.gov.br/ar/qualar/>. Acesso em: 01 jul. 2023.

HAGHRAH, A.A; GHAEMI, S. PyIT2FLS: A new python toolkit for interval type 2 Fuzzy logic systems. **arXiv**. Disponível em: <https://arxiv.org/abs/1909.10051>. Acesso em: 23 dez. 2023.

MENDEL, J. M. Fuzzy logic systems for engineering: A tutorial. **Proceedings of the IEEE**, New York, v. 30, n. 3, p. 355-377, mar. 1995.

MENDEL, J.M.; JOHN, R.I.; LIU, F. Interval type-2 Fuzzy logic systems made simple. **IEEE transactions on fuzzy systems**, New York, v. 14, n. 6, p. 808-821, 2006.

MENDEL, J.M. Type-2 Fuzzy sets and systems: An overview. **IEEE computational intelligence magazine**, New York, v. 2, n. 2, p. 20-29, mai 2007.

ORGANIZAÇÃO MUNDIAL DA SAÚDE (OMS). 9 out of 10 people worldwide breathe polluted air, but more countries are taking action. **WHO news**. Geneva, 2 mai. 2018. Disponível em: <https://www.who.int/news-room/detail/02-05-2018-9-out-of-10-people-worldwide-breathe-polluted-air-but-more-countries-are-taking-action>. Acesso em: 20 jul. 2023.

RASCHKA, S; PATTERSON, J; NOLET, C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. **Information**, v. 11, n. 4, p. 193, 2020. Disponível em: <https://www.mdpi.com/2078-2489/11/4/193> . Acesso em: 03 jun. 2023.

RIZOL, P. M. S. R.; MESQUITA, L.; SAOTOME, O. Lógica Fuzzy tipo-2. **Revista sodebras**, v. 6, p. 27-46, 2011. Disponível em: <https://www.sodebras.com.br/edicoes/N70.pdf>. Acesso em: 01. mar. 2023

ROSS, T. J. **Fuzzy logic with engineering applications**. 3.ed, Reino Unido: John Wiley and Sons Ltd, 2010.

SANDRI, S.; CORREA, C. Lógica Nebulosa. In: ESCOLA DE REDES NEURAIS, 5, 1999, São José dos Campos. **Anais [...]** São José dos Campos: ITA, 1999. p. c073-c090.

WARNER, J; SEXAUER, J. JDWarner/scikit-fuzzy: Scikit-Fuzzy version 0.4.2. Disponível em: <https://zenodo.org/records/3541386>. Acesso em: 23 dez. 2023.

ZADEH, L. A. Fuzzy sets. **Information and control**, v. 8, n. 3, p. 338-353, 1965. Disponível em: <https://www.sciencedirect.com/science/article/pii/S00199586590241X>. Acesso em: 02 mai. 2023