UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Abordagem Híbrida FuzzyNetClass: Uma Contribuição à Classificação do Tráfego de Streaming de Vídeo Integrando Lógica Fuzzy Valorada Intervalarmente e Aprendizagem de Máquina

Eduardo Maroñas Monks

Pelotas, 2023

Eduardo Maroñas Monks

Abordagem Híbrida FuzzyNetClass: Uma Contribuição à Classificação do Tráfego de Streaming de Vídeo Integrando Lógica Fuzzy Valorada Intervalarmente e Aprendizagem de Máquina

Orientador: Prof. Dr. Adenauer Corrêa Yamin
Coorientador: Prof. Dr. Renata Hax Sander Reiser

Pelotas, 2023

Insira AQUI a ficha catalográfica
(solicite em http://sisbi.ufpel.edu.br/?p=reqFicha)

Eduardo Maroñas Monks

Abordagem Híbrida FuzzyNetClass: Uma Contribuição à Classificação do Tráfego de Streaming de Vídeo Integrando Lógica Fuzzy Valorada Intervalarmente e Aprendizagem de Máquina

Data da Defesa: 27 de fevereiro de 2023

Banca Examinadora:

Prof. Dr. Anderson Paiva Cruz
Doutor em Computação pela Universidade Federal do Rio Grande do Norte.

Prof. Dr. André Rauber Dubois
Doutor em Computação pela Heriot-Watt University, HWU, Escócia.

Prof. Dr. João Ladislau Barbará Lopes
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Adenauer Corrêa Yamin (orientador)
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Profa. Dra. Renata Hax Sander Reiser (coorientadora)
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Dedico este trabalho a todos os professores que tive na vida, principalmente, aos meus dois primeiros professores que foram os meus pais Pedro e Maria.

AGRADECIMENTOS

Ao Prof. Dr. Adenauer Corrêa Yamin, meu orientador, desde o meu estágio do ensino técnico em 1995, passando pela graduação como orientador e que tenho o prazer de compartilhar conhecimento ao longo de quase 30 anos. Muito obrigado, mais uma vez por toda a ajuda e orientação.

À Profª. Drª. Renata Hax Sander Reiser por toda a dedicação e suporte ao longo da orientação nesta Tese. Muito obrigado por toda a ajuda!

Agradeço aos colegas do grupo de pesquisa LUPS do PPGC da UFPel, em especial à Bruno Moura Paz de Moura pela parceria e pela ajuda na construção dessa Tese, muito obrigado!

A minha família, minha esposa Vanessa e meu filho Greg, pela paciência e compreensão para com os momentos onde não pude participar de forma íntegra na nossa vida social. Valeu o sacrifício e muito obrigado pela força e incentivo, amo vocês!

Aos meus pais, irmãos, familiares e amigos o meu muito obrigado pelas palavras de incentivo e apoio para seguir em frente. Muito obrigado!

Agradeço a UFPEL e ao curso do PPGC em Computação pela oportunidade de realizar este trabalho em uma instituição pública e de qualidade.

People think of education as something they can finish.
— ISAAC ASIMOV

RESUMO

MONKS, Eduardo Maroñas. **Abordagem Híbrida FuzzyNetClass: Uma Contribuição à Classificação do Tráfego de Streaming de Vídeo Integrando Lógica Fuzzy Valorada Intervalarmente e Aprendizagem de Máquina** . Orientador: Adenauer Corrêa Yamin. 2023. 184 f. , Universidade Federal de Pelotas, Pelotas, 2023.

Dentre os desafios para a classificação do tráfego de rede, destaca-se a manutenção da privacidade dos usuários, a qual tem por base o uso de técnicas de criptografia e de ofuscação do tráfego. Tal desafio, associado a dinamicidade e similaridade entre os protocolos utilizados pelos diferentes serviços, bem como às típicas oscilações das condições operacionais das redes, as quais impactam na análise dos dados capturados, vem inviabilizando o uso de métodos clássicos para classificação. Considerando isto, o objetivo geral desta Tese é a concepção de uma abordagem híbrida para classificação do tráfego de rede, denominada FuzzyNetClass, direcionada ao perfil atual de uso das redes de computadores e deste modo considerando as incertezas geradas pelas flutuações nos recursos de rede das infraestruturas compartilhadas, que são de natureza não determinística. Mais especificamente, este trabalho visa contribuir para a classificação do tráfego relacionado ao *streaming* de vídeo, explorando a integração de sistemas de inferência baseados em lógica fuzzy valorada intervalarmente e algoritmos de aprendizagem de máquina. Nesta perspectiva, a abordagem FuzzyNetClass estende os trabalhos relacionados ao explorar algoritmos de aprendizagem que contribuem para a classificação realizada pelo sistema de controle fuzzy proposto, entretanto, preservando para os especialistas envolvidos aspectos relacionados à sua interpretabilidade. Dentre as contribuições da abordagem FuzzyNetClass destacam-se: (i) classificação baseada no conhecimento de especialistas e na exploração da lógica fuzzy multivalorada; (ii) classificação híbrida, a qual tem potencial de prover resultados de maior confiabilidade e também mais realísticos; e, (iii) a concepção de mecanismos de otimização que advém da integração entre técnicas de aprendizagem de máquina e a inferência baseada em lógica multivalorada. Para avaliação das contribuições da abordagem FuzzyNetClass foram discutidos três estudos de casos, os quais consideraram *Datasets* concebidos a partir de capturas reais de tráfego de rede. Os resultados obtidos se mostraram promissores e apontam para a continuidade dos esforços de estudo e pesquisa no tema.

Palavras-chave: Classificação do tráfego de Rede. Lógica Fuzzy Valorada Intervalarmente. Aprendizagem de Máquina. Streaming de Vídeo.

ABSTRACT

MONKS, Eduardo Maroñas. **FuzzyNetClass Hybrid Approach: A Contribution to Video Streaming Traffic Classification Integrating interval-valued fuzzy logic and Machine Learning**. Advisor: Adenauer Corrêa Yamin. 2023. 184 f. , Federal University of Pelotas, Pelotas, 2023.

Among the challenges for classifying network traffic, the maintenance of user privacy stands out based on cryptography and traffic obfuscation techniques. This challenge, associated with the dynamics and similarity between the protocols used by the different services, as well as the typical oscillations in the operational conditions of the networks, which impact the analysis of the captured data, has made the use of classical methods for classification unfeasible. Considering this, the general goal of this Thesis is the conception of a hybrid approach for network traffic classification, called FuzzyNetClass, directed to the current profile of use of computer networks and thus considering the uncertainties generated by fluctuations in the network resources of the infrastructures shared, which are non-deterministic. More specifically, this work aims to contribute to the traffic classification related to video streaming, exploring the integration of inference systems based on interval-valued fuzzy logic and machine learning algorithms. In this perspective, the FuzzyNetClass approach extends the related work by exploring learning algorithms that contribute to the classification of the proposed fuzzy control system, preserving aspects related to its interpretability for the involved specialists. Among the FuzzyNetClass approach contributions, the following stand out: (i) classification based on the specialist's knowledge and the exploration of multivalued fuzzy logic; (ii) hybrid classification, which has the potential to provide more reliable and more realistic results; and, (iii) the design of optimization mechanisms that come from the integration between machine learning techniques and inference based on multivalued logic. Three case studies were discussed to evaluate the FuzzyNetClass approach's contributions, considering datasets conceived from real network traffic captures. The results were promising and pointed to continuing study and research efforts.

Keywords: Network Traffic Classification. Interval-Valued Fuzzy Logic. Machine Learning. Video Streaming.

LISTA DE FIGURAS

Figura 1 Classificação do Tráfego Baseado em Endereços de Portas e Protocolos 28
Figura 2 Classificação do Tráfego Baseado em Endereços de Portas 29
Figura 3 Exemplo de Proporção de Pacotes por Protocolo em uma Captura de Tráfego. 30
Figura 4 Exemplo de Informações Disponíveis em uma Tupla de um Fluxo, com Informações Adicionais 33
Figura 5 Visão Geral da Geração e Transmissão de *Streaming* de Vídeo com Protocolo DASH 40

Figura 6 Funções de Pertinência das Operações Padrões de Conjuntos Fuzzy 45
Figura 7 Representação Gráfica dos α-níveis: $[A]^{\alpha}$ e $[A]^{0} \neq \mathbb{R}$ 46
Figura 8 Arquitetura de um Sistema de Inferência Fuzzy 53
Figura 9 Visão Geral do Controlador de Mamdani 59

Figura 10 Representação dos Conjuntos Fuzzy Valorados Intervalarmente . . 64
Figura 11 Exemplos de Conjuntos Fuzzy do Tipo-2 Intervalares 65
Figura 12 Função de Pertinência Secundária Intervalar em x=4 65
Figura 13 Conjunto Fuzzy Valorado Intervalarmente 66
Figura 14 Comparação de Conjuntos Fuzzy do Tipo 1 e 2 Intervalares 67

Figura 15 Taxionomia para Classificação do Tráfego de Rede 77

Figura 16 FuzzyNetClass: Visão Geral da Arquitetura 97

Figura 17 Visão Geral da Metodologia Desenvolvida para Seleção de Atributos 112
Figura 18 Extração de Atributos em Sendo Processada na Ferramenta CicFlowMeter 113
Figura 19 Procedimentos para a Avaliação de Atributos 114

Figura 20 Funções de Pertinência dos Atributos de Entrada (PLM, PLS, BIAT) e Atributo de Saída (Classificação de Vídeo) 122
Figura 21 Processo de Fuzzificação dos Atributos de Entrada: PLM, PLS e BIAT123
Figura 22 Valores Médios dos Limites Inferiores e Superiores para as Variáveis considerando nos Datasets 17102021 e 24102021 Redutores C e CoS126
Figura 23 Dispersão dos Valores Pontuais para os *Datasets* 17102021 e 24102021 com o Uso dos Redutores C e CoS 127
Figura 24 Acurácia para *Streaming* de Vídeo por Algoritmo, Gerado na Ferramenta KEEL e na Abordagem FuzzyNetClass 128

Figura 25 Quantidade de Regras por Algoritmo Gerado na Ferramenta KEEL e de Forma Manual 129
Figura 26 Interface da Ferramenta KEEL para Experimento com Algoritmo FURIA 134
Figura 27 Acurácia por Algoritmo em Abordagem Fuzzy 134
Figura 28 Acurácia por Algoritmo em Aprendizagem de Máquina 135
Figura 29 F1 Score, Algoritmos Fuzzy para VoD (a) e Live (c), Algoritmos em Aprendizagem de Máquina para VoD (b) e Live (d) 136
Figura 30 Quantidade de Regras Geradas por Algoritmo Geral, para VoD e para Live 137
Figura 31 Modelo das Variáveis de Entrada e Saída 143
Figura 32 Exemplo de Regra Gerada pelo Algoritmo FURIA 145
Figura 33 Valores Médios dos Intervalos Inferiores e Superiores para os Termos Linguísticos 146
Figura 34 Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* A 148
Figura 35 Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* B 148
Figura 36 Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* C 149
Figura 37 Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* D 149

LISTA DE TABELAS

Tabela 1 Exemplificação de Normas Fuzzy Triangulares 50
Tabela 2 Exemplificação de Conormas Fuzzy Triangulares 51

Tabela 3 Comparação entre os Artigos que Abordam Classificação do Tráfego Criptografado 87
Tabela 4 Trabalhos Relacionados 88

Tabela 5 Exemplos de Fluxos com Problemas no *Dataset* UNB ISCX Network Traffic (VPN-nonVPN) 107
Tabela 6 Atributos Extraídos dos Fluxos pela Ferramenta CicFlowMeter . . . 110
Tabela 7 Descrições dos *Datasets* Nomeados A, B, C e D 112
Tabela 8 Valores Utilizados para Normalização em cada Atributo 114
Tabela 9 Resultados do Avaliadores na Ferramenta WEKA 115
Tabela 10 Atributos Aplicados nos Algoritmos de Classificação 116
Tabela 11 Atributos Selecionados 117

Tabela 12 Composição dos Fluxos por *Dataset* 120
Tabela 13 Base de Regras da FuzzyNetClass 124
Tabela 14 Classificação de *Streaming* de Vídeo - Resultados Resumidos . . . 125
Tabela 15 Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores Fuzzy para os Fluxos Contidos no *Dataset* 17102021 129
Tabela 16 Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores em Aprendizagem de Máquina para os Fluxos Contidos no *Dataset* 17102021. 130
Tabela 17 Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores em Aprendizagem de Máquina para os Fluxos Contidos no *Dataset* 24102021 130
Tabela 18 Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores Fuzzy para os Fluxos Contidos no *Dataset* 24102021.130
Tabela 19 Resultados da Métrica de Entropia 131
Tabela 20 Resultados da Métrica de Entropia na Saída 132
Tabela 21 Algoritmos Selecionados para o Estudo de Caso 133
Tabela 22 Tempo de Execução por Algoritmo(s) 137
Tabela 23 Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* A 138
Tabela 24 Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* B 138

Tabela 25 Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* C 139
Tabela 26 Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* D 139
Tabela 27 Configuração das Variáveis 143
Tabela 28 Número de Regras Geradas pelo Algoritmo FURIA para Cada Termo Linguístico de Saída 144
Tabela 29 Tabela Acurácia Resumida - Estudo de Caso 3 145
Tabela 30 Resultados da Métrica de Entropia para as Variáveis de Entrada . . 150
Tabela 31 Resultados da Métrica de Entropia na Saída 151

LISTA DE ABREVIATURAS E SIGLAS

ABR	*Adaptative Bit Rate*
ALFG	Abordagem de Lógica Fuzzy Geral
API	*Application Programming Interface*
CDN	*Content Delivery Network*
CfsSubsetEval	*Correlation-based Feature Subset Selection*
ChiRW	*Chi Approach with Rule Weights*
CNN	*Convolutional Neural Network*
CSV	*Comma Separated Values*
CTR	Classificação do tráfego de Rede
DASH	*Dynamic Adaptive Streaming over HTTP*
DNS	*Domain Naming System*
DOS	*Denial of Service*
DPI	*Deep Packet Inspection*
ELF	Extensões Lógica Fuzzy
FARC-HD	*Fuzzy Association Rule-based Classification method for High-Dimensional problems*
FNC	FuzzyNetClass
FPGA	*Field Programmable Gate Array*
FTP	*File Transfer Protocol*
FURIA	*Fuzzy Unordered Rule Induction Algorithm*
GANN	*Genetic Algorithm-Neural Network*
GPU	*Graphics Processing Unit*
HAS	*HTTP Adaptive Streaming*
HTTPS	*Hyper Text Transfer Protocol Secure*
HTTP	*Hyper Text Transfer Protocol*
IANA	*Internet Assigned Numbers Authority*

IA	Inteligência Artificial
IPSEC	*Internet Protocol Security*
IPv4	*Internet Protocol* v4
IPv6	*Internet Protocol* v6
IoT	*Internet of Things*
IVTURS	*Interval-Valued fuzzy reasoning method with TUning and Rule Selection*
KDD	*Knowledge Data Extraction*
KEEL	*Knowledge Extraction based on Evolutionary Learning*
KNN	*K-Nearest Neighbors*
MPD	*Media Presentation Description*
NFSNet	*National Science Foundation Network*
OSI	*Open System Interconnection*
P2P	*Peer to Peer*
PDF	*Portable Document Format*
RAL	Revisão Assistemática da Literatura
SMTP	*Simple Mail Transfer Protocol*
SSL	*Secure Socket Layer*
SVM	*Support Vector Machine*
T1FL	Lógica Fuzzy Tipo-1
T2FL	Lógica Fuzzy Tipo-2
TCP	*Transmission Control Protocol*
TCR	Tráfego Criptografado
TLS	*Transport Layer Security*
TR	Tráfego de Rede
TTR	Tipos do tráfego de Rede
UDP	*User Datagram Protocol*
UFNET	*University of Florida Network*
VBR	*Variable Bit Rate*
VoD	*Video on Demand*
VOIP	*Voice over IP*
VPN	*Virtual Private Network*

SUMÁRIO

1 INTRODUÇÃO 19
1.1 Principais Desafios para Classificação do Tráfego de Rede 22
1.2 Problema de Pesquisa 23
1.3 Objetivos 24
1.4 Estrutura da Tese 26

2 CLASSIFICAÇÃO DO TRÁFEGO DE REDES 27
2.1 Perspectiva Histórica 27
2.2 Principais Componentes 31
2.3 Estratégias Clássicas para Classificação 34
2.4 Características do Tráfego de *Streaming* de Vídeo 36
2.5 Considerações do Capítulo 40

3 LÓGICA FUZZY 42
3.1 Conceituação de Conjuntos Fuzzy 42
3.1.1 Operações Padrões entre Conjuntos Fuzzy 44
3.1.2 Definição de α-Nível de Conjuntos Fuzzy 45
3.2 Relações sobre Conjuntos Fuzzy 46
3.3 Conectivos da Lógica Fuzzy 48
3.3.1 Negação Fuzzy e Operadores Duais 48
3.3.2 Funções de Agregação Fuzzy 49
3.3.3 Implicações Fuzzy 51
3.4 Sistemas Baseados em Regras Fuzzy 52
3.4.1 Visão Geral 52
3.4.2 Componentes da Arquitetura 53
3.5 Controladores Fuzzy 57
3.5.1 Controlador Fuzzy de Mamdani-Assilian 58
3.5.2 Controlador Fuzzy de Takagi-Sugeno 59
3.6 Considerações do Capítulo 60

4 LÓGICA FUZZY VALORADA INTERVALARMENTE 61
4.1 Contextualização Histórica de Conjuntos Fuzzy Tipo-2 61
4.2 Conceituação de Conjuntos Fuzzy Tipo-2 62
4.2.1 Conceitos Relevantes 63
4.2.2 Conjuntos Fuzzy Valorados Intervalarmente 63
4.2.3 Relações entre Conjuntos Fuzzy e Tipo-2 Intervalar 66
4.3 Lógica Fuzzy Valorada Intervalarmente 67
4.3.1 Ordens Parciais em IVFS 67

4.3.2 Negações Fuzzy Valoradas Intervalarmente . . . 68
4.3.3 Agregações Fuzzy Valoradas Intervalarmente . . . 69
4.3.4 Conectivos Fuzzy e Ordens Admissíveis em IVFS . . . 69
4.3.5 Entropia Intervalar . . . 72
4.4 Considerações do Capítulo . . . 74

5 ESTADO DA ARTE EM CLASSIFICAÇÃO DO TRÁFEGO DE REDES . . . 75
5.1 Classificação do Tráfego de Redes Explorando Aprendizagem de Máquina . . . 76
5.2 Discussão sobre Trabalhos Selecionados Explorando Aprendizagem de Máquina . . . 86
5.3 Classificação do Tráfego Explorando Lógica Fuzzy . . . 87
5.4 Discussão sobre Trabalhos Selecionados Explorando Lógica Fuzzy . 94
5.5 Considerações do Capítulo . . . 95

6 FUZZYNETCLASS: MODELAGEM ARQUITETURAL . . . 96
6.1 Inserção de Dados . . . 96
6.2 Classificação do Tráfego de Redes . . . 98
6.2.1 Visão das Estratégias de Classificação da Abordagem FuzzyNetClass . 98
6.2.2 Etapas da Classificação . . . 99
6.3 Extração de Dados . . . 103
6.4 Considerações do Capítulo . . . 104

7 FUZZYNETCLASS: MODELAGEM DAS ETAPAS OPERACIONAIS PARA INSERÇÃO DE DADOS . . . 105
7.1 Geração dos *Datasets* Empregados . . . 105
7.2 Montagem dos Fluxos de Pacotes e Extração de Atributos . . . 108
7.3 Processo de Seleção de Atributos . . . 111
7.3.1 Captura de Tráfego . . . 111
7.3.2 Extração de Atributos . . . 112
7.3.3 Avaliação de Atributos . . . 114
7.3.4 Validação de Atributos . . . 117
7.4 Considerações do Capítulo . . . 117

8 FUZZYNETCLASS: ESTUDOS DE CASO . . . 118
8.1 Estudo Caso 1: Classificação com Abordagem Fuzzy . . . 118
8.1.1 Descrição do Estudo de Caso . . . 119
8.1.2 Discussão dos Resultados . . . 125
8.2 Estudo de Caso 2: Impacto dos Atributos na Classificação . . . 133
8.2.1 Descrição do Estudo de Caso . . . 133
8.2.2 Discussão dos Resultados . . . 133
8.3 Estudo de Caso 3: Classificação com Abordagem Híbrida . . . 140
8.3.1 Descrição do Estudo de Caso . . . 140
8.3.2 Discussão dos Resultados . . . 145
8.4 Considerações do Capítulo . . . 151

9 CONSIDERAÇÕES FINAIS . . . 154
9.1 Principais Conclusões . . . 154
9.2 Publicações Realizadas . . . 156
9.3 Continuidade da Pesquisa . . . 158

REFERÊNCIAS 159

APÊNDICE A FERRAMENTAS PARA CAPTURA E ANÁLISE DO TRÁFEGO DE REDE 178
A.1 Ferramenta Tcpdump 178
A.2 Ferramenta Wireshark 178
A.3 Ferramenta CicFlowMeter 179

APÊNDICE B CLASSIFICADORES 180
B.1 Classificadores baseados em Lógica Fuzzy 180
B.2 Classificadores Baseados em Aprendizagem de Máquina 181

APÊNDICE C AMBIENTES DE *SOFT COMPUTING* 183
C.1 Ferramenta Juzzy 183
C.2 Plataforma WEKA 183
C.3 Ferramenta KEEL 184

1 INTRODUÇÃO

"Abra que eu quero voar,
O mais alto que eu puder,
Um dia eu vou sair,
Vou morar no ar".

Vou Morar No Ar
Casa das Máquinas

A classificação do tráfego em redes de computadores constitui um processo hoje entendido como central para diversas áreas relacionadas às redes de computadores, dentre as quais destacam-se às de segurança, qualidade de serviço, contabilização e diferenciação de serviços (SALMAN et al., 2020).

O aumento exponencial do número de dispositivos em rede na atualidade, principalmente com o advento da IoT (*Internet of Things*) (QIU et al., 2018), introduz a previsão de mais de 29 bilhões de dispositivos conectados em 2023 (STATISTA, 2022a). Neste cenário, a perspectiva de crescimento do volume de dados a serem trafegados e a diversidade de aplicações e protocolos envolvidos, acrescidos da disseminação do emprego de criptografia em todos os dados trafegados, serão fatores que aumentarão a complexidade para classificação do tráfego de rede, o que introduz diferentes desafios de estudo e pesquisa nesta área (D'ALCONZO et al., 2019a).

Dentre os fatores que vem propiciando este crescimento do tráfego global de dados destaca-se o avanço nas tecnologias de comunicação, seja de natureza terrestre, cabos de diferentes padrões e/ou fibras óticas, bem como as tecnologias sem fio para suporte aos dispositivos móveis. Neste particular, importante destacar que estes são predominantes no acesso dos usuários à Internet, existindo cerca de 15 bilhões de dispositivos móveis registrados em 2022 (STATISTA, 2022b).

Os dispositivos móveis oferecem aplicativos que permitem o acesso à ferramentas de comunicação, redes sociais, jogos, dentre outras. Estas aplicações de comunicação, tais como Whatsapp e Telegram, propiciam aos usuários interações com qualidade, além de permitir a reprodução de vídeos com definição elevada. Desta forma,

o consumo de largura de banda tem crescido durante os últimos anos e a tendência é uma elevação contínua do consumo para os próximos anos.

Por sua vez, o cenário da IoT, com sua crescente quantidade de dispositivos em rede, tem o potencial de ser um dos maiores consumidores de largura de banda. Como exemplo de dispositivos na IoT que possuem elevado consumo de largura de banda destacam-se as câmeras de vídeo. Estas câmeras possuem diversas aplicações, dentre estas um dos empregos mais comuns acontecem na área de segurança de patrimônios físicos ou mesmo de pessoas. Neste cenário de uso, potencialmente irá ocorrer uma significativa transferência de dados, devido a quantidade de câmeras que vem sendo adotadas em cenários associados às propostas emergentes de *smart homes* e *smart cities* (KUNST et al., 2018).

As aplicações que fazem uso de vídeos sob demanda, como no caso do Netflix, Youtube e Amazon, possuem alta resolução e geram fluxos com significativo consumo de largura de banda. Este tipo de aplicação é a responsável pela maior parte do tráfego na Internet atualmente, seja pelo volume de dados associado, como pelo significativo número de usuários que fazem uso deste tipo de serviço.

As aplicações que fazem uso de *Live* necessitam de recursos de rede específicos. Nestas aplicações as condições da rede são essenciais para o bom funcionamento devido a comunicação ser síncrona. Em caso do tipo *streaming* de vídeo os recursos de largura de banda, as perdas de pacotes, o atraso e a variação de atraso na rede podem restringir a qualidade e o uso do serviço. Por exemplo, o *Live streaming* também é usado em chamadas de vídeo/áudio em aplicativos de comunicação, sistemas de webconferência/vídeoconferência e transmissões ao vivo de eventos.

Desta forma, a pesquisa de novas abordagens para incremento na classificação do tráfego de rede, em particular quando aplicada em cenários de *streaming* de vídeo, torna-se fundamental para auxiliar os administradores no planejamento da infraestrutura de rede, uma vez que a qualidade deste tipo do tráfego é bastante sensível às condições operacionais da rede.

Oportuno registrar que em função da natureza das infraestruturas das redes de computadores, as informações capturadas das mesmas estão sujeitas a incertezas e imprecisões no seu comportamento. Isto é gerado sobretudo pela gerência não determinística de acesso aos canais de transmissão, bem como pelo uso compartilhado destes canais, o que contribui para uma maior variabilidade no comportamento das informações de gerência capturadas.

Neste contexto, destaca-se a Lógica Fuzzy baseada nos conjuntos fuzzy valorados intervalarmente (IVFS ou IT2FS) (MORENO et al., 2020). Cada intervalo representa o conhecimento do especialista, que apesar de não saber o valor exato das informações pertinentes as variáveis que definem os conjuntos fuzzy, possui informação quanto aos limites inferior e superior, para o grau de pertinência intervalar associada a cada

elemento de um conjunto fuzzy. Esta interpretação é considerada nesta Tese, como metodologia com potencial de trabalhar com informações imprecisas, modelando incertezas de aplicações reais na área de *streaming* de vídeo, a partir dos graus de pertinência em IVFS.

Por sua vez, as abordagens baseadas na lógica multivalorada dos conjuntos fuzzy tipo 2 (T2FS) (ZADEH, 1973), estendem a abordagem dos conjuntos fuzzy tipo 1 (T1FS) (ZADEH, 1965), no sentido de contribuir para soluções de problemas complexos, provendo modelagem para raciocínio, dedução e cálculo com informações imperfeitas (ZADEH, 1994), como aquelas decorrentes do monitoramento de aplicações com fluxos de rede com maior duração, como as que manipulam *streaming* de vídeo, sob redes de computadores de larga abrangência, as quais de forma inerente apresentam comportamento não determinístico.

Nessa interpretação baseada em IVFS, consideram-se duas funções, mapeando as operações sobre subintervalos fechados do intervalo unitário $[0, 1]$ e, portanto, duplica-se a complexidade com relação a abordagem com T1FS. Entretanto, considerando os avanços tecnológicos, incrementando o poder de processamento, considera-se pertinente o desenvolvimento de abordagens com suporte em IVFS, mais especificamente contribuições em problemas onde exista uma grande imprecisão da informação para modelagem das funções de pertinência, como é no caso da classificação do tráfego de *streaming* video em redes computacionais.

As funções de pertinência definindo IVFS não são tão específicas quanto sua contraparte em T1FS, mas essa menor especificidade torna a modelagem dos controladores fuzzy mais realística em diferentes cenários de aplicações. O principal ganho é permitir a expressão da incerteza via a função de pertinência valorada intervalarmente. Esta incerteza, medida pelo diâmetro dos dados intervalares, está considerada desde a modelagem, sendo preservada quando do processamento dos conjuntos fuzzy tem-se resultados mais confiáveis.

A teoria matemática baseada no conceito de IFVS e sua correspondente abordagem lógica, tem sido amplamente estudada como suporte ao desenvolvimento de aplicações em T2FS. Destacam-se algumas áreas de aplicação e desenvolvimento tecnológico baseados em IVFS, como: Raciocínio Aproximado (ZENG; FENG, 2014), Sistemas Especialistas (CHEN; BARMAN, 2019), Sistemas de Classificação Baseados em Regras (SANZ et al., 2011), Sistemas de Controle (MO et al., 2022) Processamento de Imagem (PEKALA et al., 2021) Intelligent Systems (PANDA; KOSKO, 2020), Tomada de Decisão (CHEN; YU, 2022), Medicina (SANZ et al., 2011; ONTIVEROS; MELIN; CASTILLO, 2020), Otimização (CARREON-ORTIZ; VALDEZ; CASTILLO, 2022).

E assim, também a pesquisa em lógica fuzzy multivalorada vem a contribuir para incremento na classificação do tráfego de rede, em particular quando aplicada na modelagem de informações incompletas para suporte ao gerenciamento de cenários

para classificação do tráfego de rede em *streaming* de vídeo.

Por sua vez, considerando o potencial número de atributos de um tráfego de rede típico, cujo total pode atingir várias dezenas, isto tem impactos significativos tanto na interpretabilidade, como no processamento baseado em lógica fuzzy, o que tem potencial de comprometer o seu emprego prático.

Nesta perspectiva, esta Tese tem por premissa otimizar a classificação do tráfego de rede, explorando aprendizagem de máquina para a seleção dos atributos a serem empregados, bem como no ajuste de funções de pertinência utilizadas. O emprego das técnicas de aprendizagem de máquina possibilita a exploração de ferramentas computacionais com contribuições sinérgicas de diversas áreas do conhecimento, sendo as principais a Ciência da Computação, a Matemática e a Estatística (HAO; HO, 2019). A aprendizagem de máquina emprega o princípio da indução, no qual se obtêm conclusões generalizáveis a partir de um conjunto particular de exemplos (REZAEI; LIU, 2019).

Desse modo, a Aprendizagem de Máquina tem apresentado resultados promissores em diversos setores de utilização, com maior potencial de contribuição nos problemas que envolvem volumes de dados de maior porte, como é o caso da classificação do tráfego de rede (D'ALCONZO et al., 2019b).

1.1 Principais Desafios para Classificação do Tráfego de Rede

Considerando este cenário apresentado, dentre os desafios atuais para a classificação do tráfego de rede, destacam-se os sumarizados a seguir:

(i) a adoção generalizada de técnicas de criptografia e de ofuscação do tráfego, a qual tem por finalidade a manutenção da privacidade dos usuários;

(ii) a complexidade, a diversidade e o volume dos dados trafegados, e a decorrente necessidade de recursos computacionais que possam realizar o processamento e apresentação de respostas em tempo útil;

(iii) a busca por resultados mais confiáveis e realísticos, e que também considerem a interpretabilidade dos resultados para suporte e/ou gerenciamento de aplicações que tenham uma maior dependência funcional dos recursos da infraestrutura das redes computacionais.

Estes três desafios vem comprometendo o emprego de métodos clássicos para classificação do tráfego em redes de computadores.

Na consolidação de uma estratégia que contemple contribuições nos três desafios acima destacados, considera-se, nesta Tese, a abordagem lógica multivalorada baseada em IVFS, modelando a informação imprecisa, inerente ao tráfego de dados em redes computacionais.

A concepção e desenvolvimento de uma abordagem para classificação do tráfego de redes de computadores, viabilizando duas relevantes possibilidades de extração do conhecimento e estruturação do raciocínio aproximado:

(i) abordagem fuzzy, mais simplificada que considera a modelagem apenas via especialista e ferramentas tradicionais para análise de dados, e/ou

(ii) abordagem híbrida, mais ampla, integrando à abordagem fuzzy às metodologias de Aprendizagem de Máquina, para modelar a informação imprecisa e determinar os graus de pertinência definido os IVFS, bem com estender a potencialidade do raciocínio aproximado dos correspondentes sistemas de inferência.

Assim, ambas possibilidades buscam contribuir para uma classificação confiável e mais realística do tráfego de *streaming* de vídeo em redes de computadores.

1.2 Problema de Pesquisa

O percentual de utilização do tráfego criptografado obteve um aumento significativo, sendo atualmente o padrão para todos os protocolos mais populares (LIU et al., 2017). Este aumento no uso de criptografia impôs novos desafios e dificuldades para os métodos clássicos de classificação do tráfego. Dentre os desafios impostos pelo uso da criptografia, oportuno destacar que os métodos clássicos por se basearem na análise de informações de cabeçalho, comportamento ou padrões, assinaturas, dos protocolos tem seu emprego inviabilizado.

Por sua vez, com cerca de 70% de todo tráfego de rede na Internet é de *streaming* de vídeo (SANDVINE, 2020). O tráfego de rede em formato de *streaming* de vídeo pode ser categorizado em vídeo sob demanda (VoD -*Video on Demand*) e vídeos de atividades ao vivo (Live - *Live Streaming*). Como exemplos de vídeos do tipo *VoD* estão os vídeos disponibilizados em plataformas tais como: Youtube, Instagram. Netflix, Facebook e Dailymotion. Como exemplos de vídeos do tipo Live estão os vídeos transmitidos em tempo de execução tais como as lives no Youtube, vídeo chamadas de ferramentas de comunicação, tais como Whatsapp e Telegram, e webconferências tais como Google Meet, Zoom, Microsoft Teams, Cisco Webex.

A revisão de literatura realizada nesta Tese, apontou que podem ser encontradas propostas para classificação do tráfego de redes com acurácias elevadas, entretanto voltadas para cenários específicos. Deste modo, esta constatação de que são raras as propostas factíveis de serem aplicados em redes de computadores com diferentes condições operacionais, tem tem mantido presente o interesse científico neste tema de pesquisa. Considerando esta motivação, decorre o Problema de Pesquisa central da Tese:

Como deve ser concebida uma abordagem para a classificação do tráfego de rede, particularmente *streaming* de vídeo, considerando o tratamento das incertezas?

No contexto desta Tese, este Problema de Pesquisa irá considerar seis perspectivas, inerentes as redes atuais de computadores:

- o uso disseminado de criptografia nos dados trafegados;
- a similaridade entre os protocolos que são usados para diferentes tipos de serviços em rede;
- a disponibilidade de *Datasets* confiáveis e atualizados para desenvolvimento da pesquisa;
- a inexistência de uma taxionomia relativa aos atributos que caracterizam os fluxos de rede;
- a imprecisão dos dados capturados, a qual é inerente às típicas flutuações nas condições operacionais das redes de computadores; e ainda
- o comportamento dinâmico dos protocolos usados em *streaming* de vídeo

Para atendimento deste Problema de Pesquisa, a próxima seção introduz os objetivos que nortearam os esforços de estudo e pesquisa desta Tese.

1.3 Objetivos

O objetivo geral desta Tese, considerando o Problema de Pesquisa definido, é a concepção de uma abordagem para classificação do tráfego de rede, denominada FuzzyNetClass. Esta abordagem está concebida em uma perspectiva híbrida, considerando fundamentações da lógica fuzzy valorada intervalarmente e metodologias de aprendizagem de máquina.

Mais especificamente, este trabalho visa contribuir para a classificação do tráfego associado a *streaming* de vídeo, explorando a integração da abordagem fuzzy valorada intervalarmente na consolidação de um sistema de controle fuzzy e algoritmos de classificação de dados, para seleção de atributos e geração automatizada da base regras fuzzy.

Objetivos Específicos

Para atingir o objetivo geral desta Tese, foram consideradas etapas de estudo e pesquisa tendo como premissa uma metodologia que facultasse um crescimento gra-

dativo na especificidade do trabalho desenvolvido. Para tanto foram elencados os seguintes objetivos específicos:

- revisar, na perspectiva da área de redes de computadores, os principais conceitos relacionados à classificação do tráfego;
- revisar a fundamentação em lógica fuzzy e lógica fuzzy valorada intervalarmente, considerando conceitos básicos de relações e conectivos fuzzy, incluindo estudos de sistemas de inferência na perspectiva de estruturação dos controladores fuzzy;
- identificar o estado da arte em classificação do tráfego de rede, considerando abordagens que explorem sistemas fuzzy e/ou técnicas de aprendizagem de máquina, empregando uma Revisão Assistemática da Literatura;
- conceber uma abordagem para classificação do tráfego de rede, considerando o problema de pesquisa elencado;
- definir uma modelagem arquitetural para abordagem concebida para classificação do tráfego de rede;
- explorar seleção de atributos com o objetivo de otimizar a classificação de *streaming* de vídeo, utilizando de forma sinérgica, tanto aprendizagem de máquina, como o seu impacto na acurácia das regras fuzzy;
- explorar as relações de pertinência baseadas em conjuntos fuzzy valorados intervalarmente visando o tratamento das incertezas e imprecisões no procedimento de classificação do tráfego de rede em *streaming* de vídeo;
- selecionar e revisar tecnologias, considerando a prototipação da abordagem FuzzyNetClass, priorizando recursos de softwares reconhecidos pela comunidade científica e de código aberto, contribuindo assim para a reprodutibilidade da abordagem proposta;
- desenvolver cenários de uso que explorem os diferentes aspectos da proposta concebida, considerando como métrica de validação a entropia intervalar;
- divulgar, ante a comunidade científica, os resultados atingidos pela pesquisa por meio de publicações em conferências e/ou periódicos especializados da área.

Sumarizando, esta Tese considera a concepção de uma abordagem híbrida integrando lógica fuzzy valorada intervalarmente e aprendizagem de máquina para realizar a classificação do trafégo de *streaming* de vídeo. A aprendizagem de máquina, irá contribuir tanto na seleção de atributos para classificação, como na otimização

da geração de regras fuzzy para inferência. Esta sinergia visa, sobretudo, prover a possibilidade de emprego da abordagem FuzzyNetClass em diferentes configurações das infraestruturas de rede, ou com distintos perfis para os tipos de tráfego de rede praticados.

1.4 Estrutura da Tese

A estrutura do texto desta Tese está organizada em nove capítulos. No Capítulo 2, são apresentados os conceitos e componentes para a classificação do tráfego de rede, mostrando a evolução cronológica das técnicas de classificação, com os componentes e as estratégias utilizadas, além das características do tráfego de rede em *streaming* de vídeo. No Capítulo 3, são apresentados os fundamentos de lógica fuzzy aplicados nesta Tese, abordando aspectos da lógica tipo fuzzy 1. No Capítulo 4, são apresentados os conceitos fundamentais de lógica fuzzy valorada intervalarmente, com o foco aos conceitos aplicados nesta Tese. No Capítulo 5, são discutidos os trabalhos relacionados para classificação de tráfego de rede com abordagens em lógica fuzzy e em aprendizagem de máquina. No Capítulo 6, é apresentada a modelagem arquitetural da FuzzyNetClass, onde a arquitetura geral da abordagem é explicada. No Capítulo 7, são apresentadas as modelagens das etapas operacionais da FuzzyNetClass, a partir da concepção dos *Datasets* e etapas de execução do modelo. No Capítulo 8, são apresentados três estudos de caso com a abordagem FuzzyNetClass, analisando os procedimentos e os resultados. No Capítulo 9, encontram-se as considerações finais desta Tese, as publicações realizadas até o momento e a sugestão de trabalhos futuros.

2 CLASSIFICAÇÃO DO TRÁFEGO DE REDES

"Complicação, tão fácil de entender".

Um Certo Alguém
Lulu Santos

Este Capítulo apresenta conceitos relacionados à classificação do tráfego de redes, os quais foram entendidos como necessários para contextualização do foco de pesquisa desta Tese. São discutidas as abordagens clássicas para classificação do tráfego de rede de forma cronológica, os principais componentes aplicados na classificação e as características de *streaming* de vídeo.

2.1 Perspectiva Histórica

A necessidade de classificação do tráfego de redes se manifestou já no início das operações da ARPANET. Em um artigo publicado em 1974 (KLEINROCK; NAYLOR, 1974) já foram realizadas análises do comportamento das redes existentes. Neste ano a estimava existissem ao todo 50 *hosts* empregando basicamente o protocolo HOST-to-HOST para as trocas de informação (WALDEN, 1975). Para as comunicações eram empregados enlaces por cabo de 50 Kbit/s e enlaces de satélite com largura de banda de 7,2 Kbit/s. Os autores realizaram medições do tráfego, avaliaram os padrões de tamanho dos pacotes, os atrasos da rede, a vazão e diversas relações entre estas variáveis, usando somente a estatística como ferramenta para análise e classificação do que estava trafegando.

Por sua vez, no artigo seminal para a área de redes de computadores (CLAFFY; POLYZOS; BRAUN, 1993) os autores apresentaram a caracterização de um tráfego já baseado em TCP/IP, para um enlace T1 (1,544 Mbit/s) usado como *backbone* da NFSNET (*National Science Foundation Network*), rede que serviu como base para a topologia a ser utilizada para concepção da Internet pública. Dois diferenciais surgem, portanto, um enlace de maior velocidade (T1) o que implica na escala dos dados envolvidos, e considerar um tráfego baseado em TCP/IP.

Assim, neste trabalho, foram analisadas capturas realizadas ao longo de 4 anos, no período compreendido entre 1988 e 1992, compreendendo o tráfego de rede associado ao emprego da família de protocolos TCP/IP. Foram construídas estatísticas baseadas somente na composição dos pacotes, não levando em consideração o comportamento associado aos fluxos dos pacotes (*flows*).

Um aspecto a ser destacado neste importante trabalho que influenciou diversas pesquisas subsequentes na área, seria a identificação dos protocolos a partir das capturas realizadas durante o período de 4 anos. A identificação realizada está resumida na Figura 1. O tráfego denominado como *OtherProtocols*, quantifica os casos onde não foram identificados protocolos TCP ou UDP. Por sua vez, a categoria *OtherPorts* registra o tráfego onde não foram utilizadas as portas padrão de comunicação dos protocolos da família TCP/IP. Observa-se que a diversidade de protocolos foi bastante pequena, entretanto, a estratégia de realizar a classificação do tráfego somente por portas de comunicação já apresentava potencial de baixa acurácia.

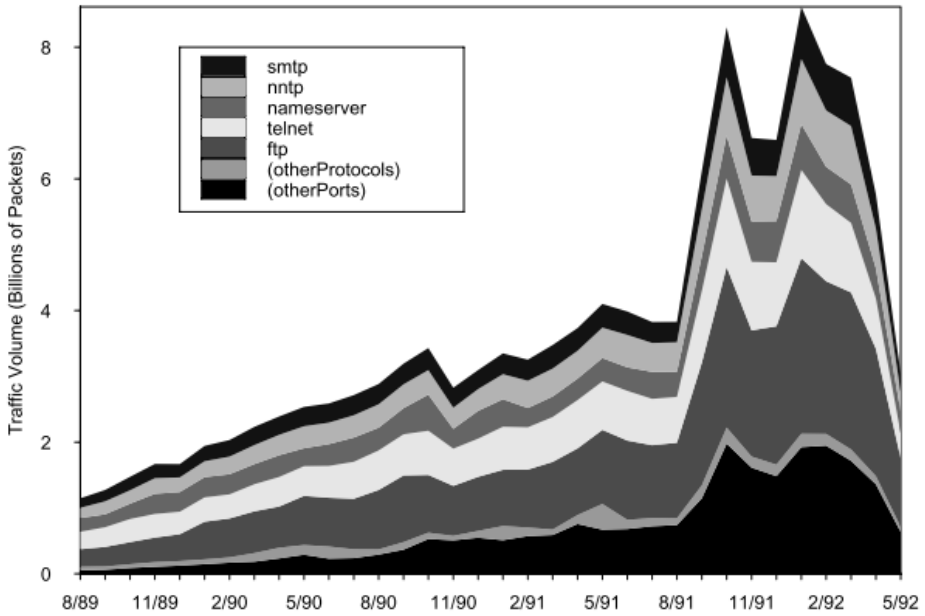


Figura 1 – Classificação do Tráfego Baseado em Endereços de Portas e Protocolos
Fonte: Claffy; Polyzos; Braun (1993)

Dando continuidade aos esforços de pesquisa neste tema, no ano de 1992, foi realizado um trabalho discutido em (VENKATRAMAN et al., 1992), quando é introduzida a avaliação das potenciais interações entre os pacotes, tendo por base o que os autores denominam de "trens de pacotes". Para tanto, os autores realizaram a captura de 29 dias do tráfego no enlace de *backbone* da Universidade da Flórida, na rede chamada de UFNET (*University of Florida Net*).

Também, neste trabalho foram utilizados métodos estatísticos para classificação do tráfego de rede. A nomenclatura "trens de pacotes"(*packet train*), consiste de um ou-

tro nome para a expressão "fluxo de pacotes", a qual é utilizada nas estratégias atuais para a classificação do tráfego de redes. Na Figura 2, pode-se observar uma tabela, organizada na época pelos autores, com a distribuição dos protocolos analisados no estudo e em destaque a grande quantidade de pacotes que não possuíam identificação baseada nas portas de comunicação (LICHTENBERG; CURLESS, 1992).

Internet well known port	# of pkts	# of bytes	% of pkts	% of bytes	Avg pkt size
telnet	1,928,548	157,702,784	25.21	14.10	81.77
login	149,545	12,065,951	1.95	1.07	80.68
nntp	1,262,128	336,395,387	16.50	30.09	266.53
smtp	228,421	33,646,489	2.98	3.00	147.30
ntp	623,628	57,296,109	8.15	5.12	91.87
ftp	22,826	1,643,684	0.29	0.14	72.00
ftp-data	134,727	51,499,024	1.76	4.60	382.24
snmp	72,125	14,944,904	0.94	1.33	207.20
who	4,952	741,000	0.06	0.06	149.63
echo	207,142	20,202,930	2.70	1.80	97.53
dst-unreach	42,552	2,978,774	0.55	0.26	70.00
domain	125,527	12,812,411	1.64	1.14	102.07
route	95,421	31,579,478	1.24	2.82	330.94
irc-6667	731,328	75,384,297	9.56	6.74	103.07
sunrpc	5,376	459,926	0.07	0.04	85.55
finger	3,667	339,005	0.04	0.03	92.44
timed	1,771	208,978	0.02	0.01	118.00
timestamp	598	37,076	0.01	0.00	62.00
uucp	10,828	2,497,601	0.14	0.22	230.66
zephyr	81,011	18,673,245	1.06	1.67	230.50
shell	7,272	3,067,923	0.09	0.27	421.88
syslog	2,196	288,544	0.02	0.02	131.39
talk	2,640	272,108	0.03	0.02	103.07
others	1,902,758	283,125,714	24.88	25.32	148.79
total	7,646,987	1,117,863,342	100.00	100.00	146.18

Figura 2 – Classificação do Tráfego Baseado em Endereços de Portas
Fonte: Venkatraman et al. (1992)

Por sua vez, na Tese (CLAFFY, 1994) o esforço de pesquisa relacionado da classificação do tráfego, teve como principal contribuição considerar enlaces de maior capacidade, os quais constituíram a base do que foi denominado de "Internet Comercial".

Foram apresentados os resultados da classificação do tráfego em enlaces de *backbone* da rede NFSNET, com links de largura de banda em 45 Mbit/s. Este trabalho já utilizou o conceito de fluxo de pacotes para classificação do tráfego, explorando um método baseado na associação direta entre portas de comunicação padrão, com o tipo de protocolo. Na Figura 3, observa-se que mesmo com uma diversidade menor de protocolos, situação típica na época da realização do trabalho, em 1994, foi registrado um percentual representativo de fluxos desconhecidos, destacados pelo autor com um sublinhado na Figura 3.

No trabalho (KARAGIANNIS et al., 2004) são discutidas estratégias para realizar

classificação do tráfego de redes considerando o crescimento do uso de protocolos P2P (*Peer to Peer*). No final dos anos de 1990, o uso de aplicações baseadas em protocolos P2P tornaram a classificação do tráfego de rede bastante mais complexa. Isto foi causado pelo uso de portas aleatórias de comunicação e o não uso de endereços de servidores centralizados de forma fixa e conhecida.

protocol	SD-NSF		SDSC		UCSD		UC-NSF		SD-viz	
	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM
icmp	1.8	1.1	0.7	0.2	2.1	0.8	1.6	0.7	3.0	0.3
tcp	86.4	91.2	15.3	29.8	76.2	84.0	86.4	90.7	2.0	6.6
egp	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
udp	11.4	7.7	37.7	62.4	21.3	15.2	12.0	8.6	94.9	93.2
otherprot	0.3	0.1	46.2	7.6	0.4	0.1	0.0	0.0	0.1	0.0
all	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
applications on top of tcp or udp										
telnet	19.6	17.9	0.6	3.2	17.7	28.9	10.1	19.3	0.0	0.7
x11	0.4	1.4	11.5	2.8	6.6	5.4	0.3	1.1	0.0	0.8
rlogin	1.7	2.5	0.1	2.3	14.1	13.6	0.5	1.5	0.0	2.4
nntp	9.8	10.0	0.5	0.2	5.1	12.4	4.1	3.9	0.1	0.0
dns	2.2	2.6	0.5	0.2	3.1	2.6	7.3	2.5	0.7	0.4
smtp	3.5	6.8	0.0	0.1	2.9	2.6	4.4	4.1	0.2	0.2
ftpdata	32.3	25.2	1.6	8.5	5.8	6.4	37.9	29.7	0.2	0.7
nfs	0.0	0.0	1.5	52.1	3.3	7.2	0.1	0.1	54.2	77.2
andrew	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	0.0
otherap	28.4	32.3	36.9	22.9	38.9	19.9	30.3	37.2	41.5	17.4

Figura 3 – Exemplo de Proporção de Pacotes por Protocolo em uma Captura de Tráfego.
Fonte: Claffy (1994)

As aplicações P2P possibilitavam o compartilhamento de arquivos de música, inicialmente, e depois de qualquer tipo de arquivo. Devido a isto, o tráfego gerado por estas aplicações se tornou relevante para provedores de acesso, para a segurança da informação, para os detentores de direitos autorais dos arquivos compartilhados e pelo grande volume do tráfego de rede gerado. Desta forma, novas estratégias de classificação do tráfego tornaram-se necessárias para este tipo de aplicação. As linhas de pesquisa basearam-se em análise da carga útil dos pacotes denominada DPI (*Deep Packet Inspection*) e da análise das características dos fluxos de rede para uso com técnicas de IA (Inteligência Artificial).

Como decorrência do crescente aumento da complexidade do perfil do tráfego de rede, o *survey* (NGUYEN; ARMITAGE, 2008) se propõe a identificar os principais trabalhos em classificação do tráfego que fazem uso de estratégias baseadas em IA. A expressão Inteligência Artificial, no período, era amplamente adotada em esforços de classificação e passou a ser chamada de aprendizagem de máquina.

Neste sentido, foi possível constatar durante a revisão de literatura feita nesta Tese que o uso de estratégias com emprego de técnicas de aprendizagem de máquina, foi estimulado pela disseminação de aplicações que se valiam de portas de comunicação

aleatórias ou mesmo portas conhecidas com protocolos diferentes, por exemplo, uso da porta 80 para protocolos de aplicações de P2P.

Um trabalho identificado como representativo neste particular seria o (CALLADO et al., 2009), onde são apresentadas iniciativas sobre classificação do tráfego de rede com diferentes tipos de estratégias, contemplando os principais desafios de pesquisa na área. Dentre outras contribuições, neste trabalho são sistematizados os principais objetivos da classificação do tráfego de rede, que são:

- Identificação do uso de aplicações e tendências: a identificação correta de aplicações de usuários e as tendências de uso de aplicações populares podem fornecer informações valiosas aos operadores de rede, para a engenharia do tráfego e para provedores na oferta de serviços de rede de acordo com a demanda.

- Detecção de Anomalias: o diagnóstico de anomalias é um fator crítico para operadores de rede e usuários finais em relação a segurança de dados e disponibilidade de serviços. As anomalias no tráfego de rede podem causar mudanças significantes no volume do tráfego e tem como origem ataques tais como DoS (*Denial of Service*) e *Worms*.

- Contabilização: para os provedores de serviços de Internet, o conhecimento sobre as aplicações dos clientes é fundamental para contabilização, cobrança e oferta de novos produtos. Por exemplo, os provedores podem ter interesse em identificar usuários que fazem uso de serviços de VoIP (*Voice over IP*) ou *streaming* de vídeo.

Esta perspectiva histórica associada ao esforço de classificação do tráfego em redes de computadores, introduz desafios que se mantiveram presentes em trabalhos relevantes da atualidade, como pode ser visto no Capítulo 5.

Os desafios historicamente sistematizados pela comunidade científica da área ainda persistem, bem como ganham complexidade em função da escalabilidade e diversidade das modernas estruturas de rede. Este aspecto, constitui um indicador concreto do potencial para estudos e pesquisas, que a área de classificação do tráfego de rede apresenta.

2.2 Principais Componentes

Neste seção são apresentados os principais componentes do tráfego de rede na perspectiva da sua classificação. As comunicações de dados em redes de computadores que utilizam a família de protocolos TCP/IP são baseadas em um modelo de 5 camadas. Esta organização compreende os protocolos e os serviços que são prestados entre as camadas (TANENBAUM; WETHERALL, 2021).

Desta forma, as camadas inferiores são responsáveis por fornecer acesso e compartilhamento ao meio físico, entre redes distintas não há necessidade de existir o mesmo padrão comunicação. Assim, os protocolos usados nestas camadas inferiores são independentes e as informações dos protocolos das camadas inferiores, normalmente, não são aproveitados para classificação do tráfego.

Por sua vez, os protocolos das camadas superiores utilizam os protocolos da família TCP/IP para comunicação, mesmo entre redes distintas. Esta propriedade é fundamental para a classificação do tráfego de rede (KUROSE; ROSS, 2021).

Modernamente, os protocolos IPv4, IPv6, TCP e UDP são a base para a troca de dados das aplicações em rede, e os componentes usualmente considerados na classificação do tráfego de rede são os protocolos, os pacotes e os fluxos de pacotes.

Protocolos de Rede

Segundo (KUROSE; ROSS, 2021), um protocolo de rede "*define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento*".

Os protocolos de rede são organizados em camadas, onde cada uma das camadas possui funcionalidades e serviços que tornam os protocolos independentes. Quando relacionados em conjunto, os protocolos de várias camadas formam uma pilha de protocolos (KUROSE; ROSS, 2021). A pilha de protocolos dominante é a pilha de protocolos TCP/IP, a qual possui 5 camadas, baseadas no modelo de referência OSI (*Open System Interconnection*) de 7 camadas. Os protocolos que atuam em cada uma das camadas são a base para a classificação do tráfego de rede por meio das informações de controle, cabeçalhos, inseridas nos pacotes.

O fato de coexistirem nas redes de computadores protocolos abertos, com implementações de protocolos de aplicação proprietários, tais como o do Skype, Zoom, Microsoft Teams, dentre outros, a classificação do tráfego de rede torna-se ainda mais complexa. Para estes protocolos de aplicação proprietários, usualmente os desenvolvedores não disponibilizam acesso a documentação sobre o formato e a função das diferentes mensagens utilizadas.

Pacotes de Rede

Um pacote de rede é formado por um cabeçalho de controle (*header*) e uma área de dados (*payload*). Os cabeçalhos dos pacotes são gerados por camada de protocolo e possuem relevância na camada onde foi gerado (TANENBAUM; WETHERALL, 2021). Por exemplo, um pacote que utilize o protocolo UDP como transporte deverá

possuir informações sobre os endereços das portas de comunicação para possibilitar que o sistema operacional no *host* remoto faça a entrega dos dados ao processo correto.

O tamanho do cabeçalho, normalmente, possui um tamanho fixo em bytes e quando existem tamanhos variáveis deve ser indicado em um campo de controle no próprio cabeçalho. No caso da área de dados do pacote, normalmente, o tamanho é variável porque dependerá do tipo de uso da aplicação que está gerando os pacotes e do padrão de rede utilizado nas camadas inferiores. Por exemplo, uma aplicação de conversação, *chat*, terá pacotes com tamanhos variáveis de acordo com o que um usuário estiver enviando ou recebendo de dados. Em uma aplicação de transferência de arquivos, comumente, os pacotes ocuparão toda a área de dados disponível até o término da transferência.

As informações dos cabeçalhos de controle dos pacotes, bem como as características das áreas de dados dos pacotes dão suporte ao emprego de diferentes metodologias para classificação do tráfego de rede, viabilizando a identificação das diferentes aplicações que geram tráfego.

Fluxos de Rede

As informações dos cabeçalhos de controle dos pacotes determinam a relação entre os mesmos, viabilizando a construção de um fluxo de pacotes (*flow*). O fluxo é composto por 5 campos, formados pelo endereço IP de origem, endereço IP de destino, protocolo da camada de transporte, endereço da porta de origem e endereço da porta de destino (KUROSE; ROSS, 2021).

O fluxo é considerado o mesmo quando os campos que o formam permanecem os mesmos. O fluxo pode ser analisado de forma bidirecional ou unidirecional, o que implica no sentido da comunicação de pacotes. A partir desta premissa pode-se coletar e analisar outras informações relevantes para classificar o tráfego de rede, tais como a quantidade de bytes gerada, o tempo de duração e a diferença de tempo entre cada pacote.

Na Figura 4 é disponibilizado o exemplo de uma tupla de 5 campos que caracteriza um fluxo coletado com o protocolo Netflow (CLAISE et al., 2004). Nesta Figura 4 também são listados campos com informações adicionais de data, hora, duração do fluxo, número de pacotes e quantidade de bytes.

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Packets	Bytes	Flows
2019-07-21 18:28:52.888	0.011	TCP	10.30.0.1:9080	->	192.168.66.3:52325	5	456	1

Figura 4 – Exemplo de Informações Disponíveis em uma Tupla de um Fluxo, com Informações Adicionais

2.3 Estratégias Clássicas para Classificação

A revisão da literatura aponta que as diferentes estratégias para classificação do tráfego sofreram uma evolução a partir da complexidade de protocolos e serviços. O primeiro método utilizado foi por meio da identificação de informações contidas nos cabeçalhos de forma direta, por exemplo relacionando a porta de comunicação ao tipo de aplicação ou protocolo.

Outro método utilizado foi a análise do comportamento de *hosts*, considerando os destinos dos pacotes e fluxos gerados. Com o avanço das aplicações e protocolos em rede, principalmente P2P, o uso de análise da carga útil dos pacotes começou a ser utilizada para relacionar padrões com assinaturas de protocolos preexistentes (TANENBAUM; WETHERALL, 2021).

Classificação Baseada em Portas

A classificação baseada em portas de comunicação faz uso da identificação do endereçamento de portas definida pela IANA (*Internet Assigned Numbers Authority*) (IANA, 2019) para os protocolos da camada de transporte TCP e UDP (KIM et al., 2008). Este método é rápido e simples para classificação do tráfego em protocolos tradicionais tais como HTTP, DNS, SMTP e FTP. Entretanto, com o advento de novas aplicações que fazem uso de portas dinâmicas e aleatórias, em conjunto com a não obrigatoriedade para registro de portas na IANA, este método tornou-se pouco eficiente.

Classificação Baseada em Comportamento de Hosts

A classificação do tráfego de rede baseada no comportamento dos *hosts* tem como característica principal a observação e modelagem de padrões nos fluxos de rede entre determinados *hosts* (NARI; GHORBANI, 2013). Nesta estratégia são usadas heurísticas sem o uso de conteúdo dos pacotes. Por exemplo, são analisados os dados sobre os destinos conectados, o número de portas diferentes em uma conexão e os protocolos de transporte utilizados.

A acurácia desta estratégia está intimamente relacionada ao local de observação do tráfego, por exemplo, se os dados foram coletados a partir dos *hosts* clientes, dos *hosts* servidores ou de algum roteador intermediário. Esta estratégia é de difícil aplicação em ambientes com um número elevado de *hosts* a serem observados, mas pode ser usada em conjunto com outras estratégias (CAO; DRABECK; HE, 2017).

Classificação Baseada na Carga Útil dos Pacotes

A classificação baseada na carga útil dos pacotes, também conhecido como DPI (*Deep Packet Inspection*), analisa o conteúdo dos pacotes em busca de padrões que estejam em uma base de assinaturas correspondentes a protocolos ou a aplicações (BUJLOW; CARELA-ESPAÑOL; BARLET-ROS, 2015). Uma das principais dificuldades desta estratégia é a necessidade de análise da carga útil de cada pacote, o que gera um alto custo computacional, e, por consequência, potencialmente um elevado tempo para identificação.

Outro fator limitante desta estratégia está associado ao uso de criptografia nos protocolos, o que torna a identificação baseada em padrões ineficiente e, em muitos casos, não utilizável. Por outro lado, devido a necessidade da análise dos dados existentes nos pacotes, surge uma possibilidade de infringimento da privacidade dos usuários, o que também poderá restringir a aplicação desta estratégia.

Classificação Baseada nas Características do Tráfego

A classificação baseada nas características do tráfego utiliza ferramentas estatísticas e a descoberta de padrões por meio de técnicas de aprendizagem de máquina. Os estudos nesta frente ganharam dimensão com o trabalho (NGUYEN; ARMITAGE, 2008) e continuam se desdobrando em diferentes trabalhos da atualidade (TAHAEI et al., 2020), (WANG et al., 2019), (D'ALCONZO et al., 2019b), (AHMAD et al., 2020).

A proposta desta estratégia é inferir características que possam distinguir unicamente protocolos e aplicações usando somente dados estatísticos oriundos de pacotes e fluxos. Por exemplo, a distribuição de tamanho dos pacotes, o tempo de inatividade entre as trocas de dados, o intervalo de tempo entre a chegada dos pacotes e o tamanho dos pacotes são algumas das características utilizadas para a classificação do tráfego.

Esta estratégia pode atingir ótimos níveis de acurácia, não introduzindo a necessidade de infringir a privacidade dos usuários. Outra vantagem desta estratégia é que mesmo que seja usada criptografia, ainda será possível realizar a classificação. Neste sentido, quando as aplicações geradoras de tráfego fazem uso de criptografia na camada de rede tais como o protocolo IPSEC) ou tunelamento (VPN - *Virtual Private Network*), empregados de forma combinada ou não, esta estratégia de classificação ainda poderá ser utilizada, desde que sejam realizados ajustes nas suas parametrizações operacionais.

2.4 Características do Tráfego de *Streaming* de Vídeo

Nos últimos anos, o desenvolvimento de redes móveis e tecnologias de *streaming* permitiram que os clientes assistissem vídeos em seus dispositivos móveis a qualquer momento, com vídeo representando 67% do tráfego global em 2016 e deverá seguir a tendência de crescimento nos próximos anos (SANDVINE, 2020).

A Internet não foi originalmente projetada para a entrega sustentada de aplicativos modernos com uso intensivo de largura de banda tais como como *streaming* de vídeo em alta qualidade. A diferença fundamental entre o tráfego de dados tradicional, um *download* de arquivo por exemplo, e o tráfego de vídeo são as restrições em tempo real no tráfego de vídeo. Não só a largura de banda necessária para o uso de vídeos em *streaming*, mas questões relacionadas ao tipo de serviço, transmissões ao vivo, vídeo-chamadas ou webconferências, tornam ainda mais complexa a tarefa de disponibilizar as transmissões sem garantias dos recursos de rede na Internet pública.

O *streaming* de vídeo em rede é caracterizado pelo envio de blocos denominados *chunks* (SANI; MAUTHE; EDWARDS, 2017). Os *chunks* são segmentos de dados enviados de acordo com as condições da rede e os recursos disponíveis no cliente e no servidor. Devido à possibilidade de alteração da qualidade do vídeo durante a transmissão e ao uso de protocolos como HTTPS, HTTP/2 e, QUIC, os quais são criptografados po padrão, a aplicação de métodos clássicos de identificação e classificação do tráfego torna-se menos eficaz (BENTALEB et al., 2018).

Os quadros de vídeo devem ser reproduzidos entre 24 e 30 quadros por segundo para criar a ilusão de movimento. Com o uso de compressão de vídeo os algoritmos realizam compressão intra e inter *frame*, que possuem dependências temporais, resultando em Intra (I), Quadros bidirecionais (B) e previstos (P). I quadros são maiores porque eles usam apenas compressão intraquadro, enquanto B e quadros P são menores porque usam quadros I anteriores para de redução de tamanho. A variabilidade em bits codificados por segundo leva ao vídeo de taxa de bits variável (VBR (*Variable Bit-Rate*)), conforme usados nos codificadores H.264 (WIEGAND et al., 2003) e MPEG-4 (PEREIRA et al., 2002).

O vídeo codificado em VBR é então transmitido para a rede. Em casos como a Internet, onde não há garantias dos recursos da rede, a transmissão será realizada com base no melhor esforço, sem garantias de entrega. Se a largura de banda não for suficiente para suportar a taxa de bits de vídeo, então o decodificador no lado do cliente começará a consumir os dados de vídeo a uma taxa maior do que a taxa dos novos dados estão sendo recebidos da rede. O decodificador acabará ficando sem pacotes de vídeo para decodificar, o que resulta em um congelamento de tela, onde o vídeo apresentará variações na qualidade e travamentos ou podem acontecer eventos de *rebuffering*.

Para evitar estas consequências das variações, sem ter que introduzir garantias caras e complexas em mecanismos de controle de largura de banda, as seguintes soluções foram usadas para tentar adaptar a taxa de bits do vídeo aos recursos disponíveis de rede:

- Usando um *buffer* de reprodução: variações na taxa de transferência de rede em curto prazo podem ser superadas usando um *buffer* de reprodução. O *player* de vídeo pode decodificar os dados pré-buscados armazenados no *buffer* de reprodução e manter o fluxo de reprodução sem falhas.

- Soluções baseadas em transcodificação: essas soluções mudam um ou mais parâmetros da compactação de dados de vídeo para variar a taxa de bits resultante. Exemplos incluem a variação da resolução de vídeo, da taxa de compactação ou da quantidade de quadros. No entanto, este processo é computacionalmente intensivo e requer suporte de hardware complexo.

- Soluções de codificação escaláveis: Essas soluções são implementadas pelo processamento dos dados de vídeo codificados. Para isso, o vídeo codificado pode ser adaptado em tempo real usando os recursos de escalabilidade do codificador.

  Algumas técnicas incluem adaptar a resolução da imagem ou a taxa de quadros, explorando a escalabilidade espacial ou temporal no de dados codificados. No entanto, são necessários servidores especializados para implementar essas soluções.

- Soluções de comutação de fluxo: Esta técnica é a mais simples para implementar e usa a estratégia de CDNs (*Content Delivery Network*), servidores de *cache* mais próximos as redes dos clientes. Os dados de vídeo brutos são pré-processados para produzir vários fluxos codificados, cada um em uma taxa de bits diferente, resultando em várias versões do mesmo conteúdo. Um algoritmo adaptativo do lado do cliente é então usado para selecionar a taxa mais apropriada dadas as condições da rede durante a transmissão.

  Estas soluções não requerem servidores especializados e usam um menor poder de processamento. No entanto, mais armazenamento e granularidade mais fina nas taxas de bits dos codificadores são necessárias para permitir que o cliente otimize sua de seleção.

Considerando a viabilidade das implementações, a opção adotada pelos serviços de *streaming* foram as abordagens com o uso de *buffers* de reprodução e soluções de comutação de fluxo.

Streaming de Vídeo com o Protocolo HTTP

No início dos anos 2000, a comunidade científica e a indústria começaram a avaliar o uso do protocolo TCP para suporte a serviços de rede tolerantes a atrasos. Até então o uso do protocolo UDP era o padrão para *streaming* de vídeo e áudio. Um *buffer* de reprodução da camada de aplicação foi introduzido para compensar as flutuações de taxas de transmissões do TCP. Outra abordagem foi aproveitar o protocolo HTTP sobre o TCP que se provou ser muito conveniente, gerando vários benefícios devido a compatibilidade com a biblioteca de software existente (BENTALEB et al., 2018).

Uma das primeiras tentativas para a implementação de entrega de vídeo por HTTP/TCP foi chamada de *Progressive Download*, onde o cliente simplesmente baixava o arquivo de vídeo, com qualidade de vídeo constante, tão rápido quanto TCP permitisse. O *player* de vídeo no lado do cliente iniciava a reprodução do vídeo antes que o *download* fosse concluído.

Uma grande desvantagem desta técnica está no fato que diferentes clientes, com diferentes capacidades e em diferentes conexões de rede recebem a mesma qualidade de vídeo, o que pode causar interrupções de reprodução indesejadas sem haver adaptações relacionadas as condições dos recursos de rede.

A partir disto, houve o desenvolvimento da abordagem denominada HAS (*HTTP Adaptive Streaming*) ou DASH (*Dynamic Adaptive Streaming over HTTP*). Nesta abordagem, um cliente de vídeo poderia solicitar de forma adaptativa diferentes taxas de bits de vídeo para que correspondesse a largura de banda que a rede pudesse suportar. Sendo assim, se a rede apresentasse melhores condições de recursos o cliente poderia reproduzir um vídeo com melhor qualidade de resolução. As principais diferenças entre o DASH e os protocolos anteriores para *streaming* de vídeo são:

- Ao contrário dos esquemas anteriores baseados em UDP, o DASH é construído sobre o protocolo TCP na camada de transporte.
- O cliente aciona o algoritmo para adaptar a qualidade do vídeo. O cliente normalmente solicita taxas de bits de vídeo com base nas condições de rede observadas, regulando assim a taxa de transmissão do servidor.
- O DASH solicita e recebe dados do vídeo em pedaço, (*chunks*), que são trechos de vídeo com vários segundos, ao invés de um fluxo contínuo de pacotes de vídeo.

Streaming de Vídeo Dinamicamente Adaptativa com o Protocolo HTTP

Esta seção apresenta uma visão geral da arquitetura do DASH e suas aplicações, os benefícios do uso do protocolo HTTP e os princípios gerais que orientam os algoritmos de adaptação de taxa de transmissão de vídeo (TOGOU; MUNTEAN, 2022).

Em sistemas DASH o *streaming* de vídeo é codificado em várias versões de diferentes taxas de bits que correspondem a resolução da imagem. Cada vídeo codificado é então fragmentado em pequenos segmentos ou pedaços de vídeo (*chunks*), cada um contendo alguns segundos de vídeo. Os pedaços de uma taxa de bits são sincronizados na linha do tempo do vídeo para com os pedaços de outras taxas de bits para que o cliente pode alternar suavemente as taxas de bits, se necessário.

Informações de conteúdo, como perfis de vídeo, metadados, tipo MIME, *codecs*, intervalos de bytes, endereço IP do servidor e URLs de *download* são descritos nos arquivos de descrição de apresentação de mídia denominados MPD (*Media Presentation Description*) associados.

O MPD descreve um conteúdo de vídeo em uma duração específica como um Período. Em um Período, existem várias versões de conteúdo, cada um conhecido como Representação. Em uma Representação, existem vários segmentos ou partes de vídeo. As URLs (*Uniform Resource Locator*) apontando para os fragmentos de vídeo em um MPD podem ser descritos explicitamente ou serem construídos por meio de um modelo no qual o cliente deriva uma URL válida para cada pedaço em uma determinada Representação.

Partes de vídeo são formatados em 3GP (STOCKHAMMER, 2011) e em cada Representação, há um único segmento de inicialização que contém os dados de configuração e as referências dos demais segmentos de mídia. Os pedaços de vídeo e MPDs são então servidos aos clientes usando servidores HTTP padrão.

Ao contrário das estratégias tradicionais de *streaming*, o DASH não controla a taxa de transmissão de vídeo diretamente. Desta forma, depende do algoritmo TCP para regular a taxa de transmissão de vídeo, que é determinada pelo *feedback* de congestionamento do caminho de rede entre cliente e servidor.

Quando uma sessão de *streaming* é iniciada, o cliente solicita o arquivo MPD do servidor HTTP e então começa a solicitar pedaços de vídeo, normalmente em ordem de sequência, o mais rápido possível para preencher o *buffer* de reprodução. Uma vez que o *buffer* de reprodução está cheio, o cliente entra em uma fase de estado estável, onde periodicamente baixa novos pedaços de acordo com a escolha baseada em algum algoritmo para ABR (*Adaptative Bit Rate*) (SPITERI; SITARAMAN; SPARACIO, 2019). Os algoritmos ABR são executados no *player* de vídeo para ajustar a qualidade de recebimento dos *chunks* de vídeo de acordo com os recursos de rede.

Na fase estável, o reprodutor de vídeo ficará no estado ON quando estiver realizando o *download* de um *chunk*, e no estado OFF em caso contrário, resultando em um padrão do tráfego alternado denominado ON-OFF (POYMANOVA; TATARNIKOVA, 2018). O tempo entre o início de dois períodos ON é denominado tempo de ciclo, normalmente de acordo com o tamanho do bloco, variando de 2 a 4 segundos em média (RAO et al., 2011)). O cliente normalmente mantém alguns pedaços no *buffer* para

manter a reprodução de forma constante.

Na Figura 5 está representada uma visão geral da geração e transmissão de *streaming* de vídeo com o uso de *chunks*, onde cada cor diferente representa uma versão com qualidade diferente do mesmo vídeo. O cliente, a partir das condições da rede ou por meio de alguma configuração selecionável, optará pela melhor versão para reprodução ao usuário.

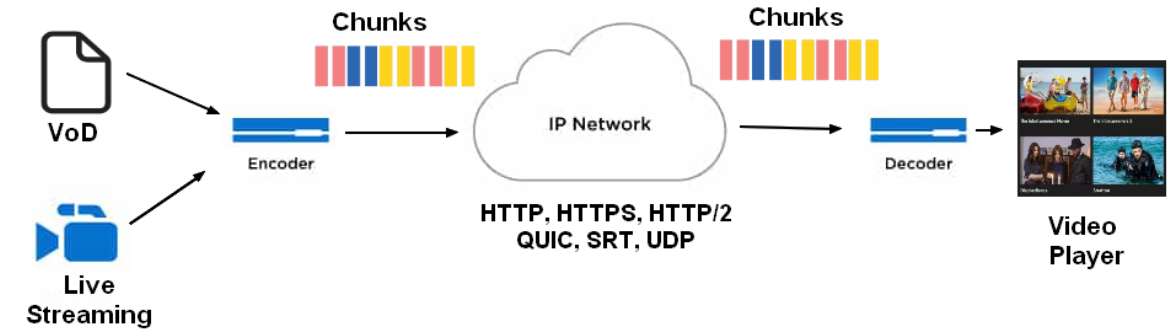


Figura 5 – Visão Geral da Geração e Transmissão de *Streaming* de Vídeo com Protocolo DASH

Live Streaming

A principal diferença entre *streaming* sob demanda e ao vivo é o tempo de geração e transmissão de conteúdo. No *streaming* sob demanda, todo o conteúdo do servidor foi gerado antecipadamente antes de transmitir para os clientes, enquanto o conteúdo de mídia é gerado dinamicamente no caso de transmissão ao vivo (DAO et al., 2022).

Devido as restrições de latência para a transmissão de eventos ao vivo, os codificadores, os protocolos e os reprodutores de vídeo devem ser ajustados para disponibilizar qualidade ao conteúdo observado pelo usuário. O método de ON-OFF é ajustado para tempos menores do que em vídeo sob demanda, o que torna as variações dos recursos de rede mais perceptíveis na apresentação do vídeo.

Outro fator importante na diferenciação de *streaming* ao vivo do *streaming* sob demanda é a impossibilidade de utilizar distribuidores de conteúdo (CDN) ou *caches* para diminuir o impacto na infraestrutura de provedores de serviços de Internet.

2.5 Considerações do Capítulo

Neste Capítulo foram apresentados aspectos históricos da área de classificação do tráfego de rede. Esta área foi consequência de demandas associadas as necessidades de gerenciamento das redes de computadores, já identificadas nas primeiras e modestas infraestruturas de rede, mas que perduram até hoje, bem como da impor-

tância da classificação dos diferentes tipos de dados trafegados para as demandas atuais daqueles que precisam administrar redes de computadores de maior porte.

Praticamente, desde o início das redes de computadores já ocorreram publicações de trabalhos relacionadas à classificação do tráfego de rede, devido a sua importância, mesmo em um período em que as infraestruturas de rede eram bastante reduzidas, tanto em número de equipamentos, como em fluxos de dados trafegados.

Mesmo em condições de baixa escalabilidade, os resultados dos trabalhos iniciais já apresentavam dificuldades em classificar de forma correta o tráfego das redes existentes na época. Com o avanço da área tecnológica surgiram novas ferramentas para auxiliar na análise e classificação do tráfego de redes, entretanto, a complexidades das infraestruturas computacionais, bem como o seu uso, surgiram de forma não proporcional aos novos desafios.

Também neste capítulo foram introduzidos aspectos relacionados aos principais componentes do tráfego de rede, bem como discutidas as vulnerabilidades das estratégias clássicas para classificação de tráfego. Por fim, são revisadas as diferentes particularidades associadas ao tráfego de *streaming* de vídeo.

3 LÓGICA FUZZY

"Dessa coisa que mete medo,
Pela sua grandeza,
Não sou o único culpado,
Disto eu tenho a certeza".

Queixa
Caetano Veloso

Neste Capítulo estão elencados os fundamentos referentes a LF (Lógica Fuzzy) entendidos como necessários para o desenvolvimento do trabalho de pesquisa contemplado nesta Tese.

As primeiras abordagens estendendo a noção valores verdade da lógica clássica foram apontadas por Jan Lukasiewicz (FONT; HÁJEK, 2002) em 1920, um lógico polonês que introduziu a lógica dos conceitos vagos a partir dos conjuntos com três graus de pertinência (0, 0.5, 1).

Segundo Von altrock (1996), a primeira publicação sobre LF foi em 1965, quando recebeu esta denominação. Seu autor foi Lotfi Asker Zadeh (ZADEH, 1965), professor na Universidade da Califórnia, Berkeley, EUA, desenvolvendo os fundamentos da LF ao combinar os conceitos da lógica clássica e os conjuntos de Lukasiewicz, definindo as funções de pertinência como extensão das funções características (ZADEH, 1965, 1975, 1994).

A principal diferença entre a proposição definida pelos conjuntos clássicos e a proposição definida sobre conjuntos fuzzy introduzida por Zadeh, está na valoração do grau de pertinência, cujos valores são números reais entre 0 e 1. Na abordagem clássica um elemento pertence ou não a um determinado conjunto, ou é verdadeiro, ou é falso, ou ainda, pode ser 0 ou 1.

3.1 Conceituação de Conjuntos Fuzzy

Na teoria dos conjuntos fuzzy, o elemento pode pertencer, não pertencer, ou ainda, pertencer parcialmente a um determinado conjunto. Assim, a cada elemento de um

conjunto fuzzy é atribuído um grau de pertinência, tendo como valoração um número real maior que 0 e menor que 1.

Na teoria clássica dos conjuntos, um elemento pertence ou não pertence a um determinado conjunto. A pertinência ou não pertinência do elemento, pode ser interpretada como uma função característica dada pela definição:

Definição 1 ***Função Característica:*** *Sejam χ um conjunto universo não vazio $(\chi \neq \emptyset)$ e A um subconjunto de χ. Define-se a função característica $f_A(x) : \chi \to \{0, 1\}$*

$$f_A(x) = \begin{cases} 1, & \textit{se } \mathrm{x} \in A; \\ 0, & \textit{se } \mathrm{x} \notin A. \end{cases}$$

Dessa forma, f_A é uma função cujo domínio é χ e a imagem está contida no conjunto $\{0, 1\}$, onde $f_A(\mathrm{x}) = 1$ indica que o elemento x está em A, e $f_A(\mathrm{x}) = 0$ indica que x não é elemento de A. Logo, a função característica descreve completamente o conjunto A, definindo quais elementos do universo χ são também elementos de A.

No entanto, sistemas que modelam incertezas nem sempre possuem fronteiras de pertinência bem definidas, por exemplo, as aplicações descritas em (RAMEZANI; LU; HUSSAIN, 2013; SEDDIKI et al., 2014; TOOSI; BUYYA, 2015; RAMEZANI; NADERPOUR; LU, 2016; THEIN et al., 2018; POURGHAFFARI; BARARI; SEDIGHIAN KASHI, 2019).

No contexto da Teoria dos Conjuntos Fuzzy, pela função de pertinência, todo elemento de um universo pertence a todos os conjuntos fuzzy sobre esse universo e com possíveis distintos graus de pertinência.

Definição 2 ***Função de Pertinência:*** *(ZADEH, 1965; ROSS, 2010a) Seja um conjunto universo $\chi \neq \emptyset$. Um conjunto fuzzy A em χ é caracterizado pela função de pertinência $\mu_A : \chi \to [0, 1]$ onde, para cada $\mathrm{x} \in \chi$, $\mu_A(\mathrm{x})$ indica o grau de pertinência de cada elemento x no conjunto fuzzy A.*

De acordo com a Definição 2, um conjunto fuzzy A é dado por:

$$A = \{(\mathrm{x}, \mu_A(\mathrm{x}))|\mathrm{x} \in \chi\}$$

onde $0 \leq \mu_A(\mathrm{x}) \leq 1$, sendo que frequentemente $\mu_A(\mathrm{x})$ e indicada por $A(\mathrm{x})$. Pode-se então descrever um conjunto fuzzy A em um universo χ como um conjunto de pares ordenados. E ainda, esse conjunto de pares ordenados pode ser visto como um conjunto de n-tuplas na abordagem lógica multi-valorada.

O valor $\mu_A(\mathrm{x}) \in [0, 1]$ denota o grau com que o elemento $\mathrm{x} \in \chi$ pertence ao conjunto fuzzy A; sendo que $\mu_A(\mathrm{x}) = 0$ e $\mu_A(\mathrm{x}) = 1$ denotam, respectivamente, a não pertinência e a pertinência completa de x ao conjunto fuzzy A.

Neste contexto, um conjunto clássico é um caso particular de um dado conjunto fuzzy, cuja função de pertinência μ_A coincide com sua função característica χ_A. Ou seja, a definição de conjunto fuzzy foi concebida a partir da extensão da função característica $\chi_A : \chi \to \{0, 1\}$, cujo contradomínio $\{0, 1\} \subseteq [0, 1]$.

3.1.1 Operações Padrões entre Conjuntos Fuzzy

Na teoria dos conjuntos fuzzy, operações com conjuntos fuzzy geram novos conjuntos fuzzy. As operações devem exibir propriedades que correspondam à intuição, cumprir a semântica da operação pretendida e também serem flexíveis o suficiente para atender aos requisitos da aplicação.

Além disso, quando as operações entre conjuntos fuzzy são aplicadas a conjuntos crisp, elas devem retornar os mesmos resultados encontrados ao operar com todos os elementos destes conjuntos (PEDRYCZ; GOMIDE et al., 1998; PEDRYCZ, 2021; WAGNER; HAGRAS, 2011).

Na abordagem clássica, sejam A e B conjuntos em χ representados pelas funções características $\chi_A, \chi_B : \chi \to \{0, 1\}$, respectivamente. Os conjuntos definindo a união e intersecção entre A e B pode ser, respectivamente, dados pelas expressões:

$$A \cup B = \{(\mathrm{x}, f_{A\cup B}(\mathrm{x}))|\mathrm{x} \in \chi\}, \text{ onde } f_{A\cup B} : \chi \to \{0, 1\}, \chi_{A\cup B}(\mathrm{x}) = \max\{f_A(\mathrm{x}), \chi_B(\mathrm{x})\}$$
$$A \cap B = \{(\mathrm{x}, f_{A\cap B}(\mathrm{x}))|\mathrm{x} \in \chi\}, \text{ onde } f_{A\cap B} : \chi \to \{0, 1\}, f_{A\cup B}(\mathrm{x}) = \min\{f_A(\mathrm{x}), \chi_B(\mathrm{x})\}.$$

E, o conjunto clássico definindo o complemento de A é dado pela expressão:

$$A^{'} = \{(\mathrm{x}, f_{A'}(\mathrm{x}))|\mathrm{x} \in \chi\}, \text{ onde } f_{A'} : \chi \to \{0, 1\}, f_{A'}(\mathrm{x}) = 1 - f_A(\mathrm{x}).$$

Na abordagem fuzzy, os conjuntos são caracterizados pelas funções de pertinência, as quais se apresentam como extensões de funções características.

Sejam A e B conjuntos fuzzy em χ representados pelas funções de pertinência $\mu_A, \mu_B : \chi \to [0, 1]$, respectivamente. Os conjuntos fuzzy definindo a união e intersecção entre A e B podem ser, respectivamente, dados pelas expressões:

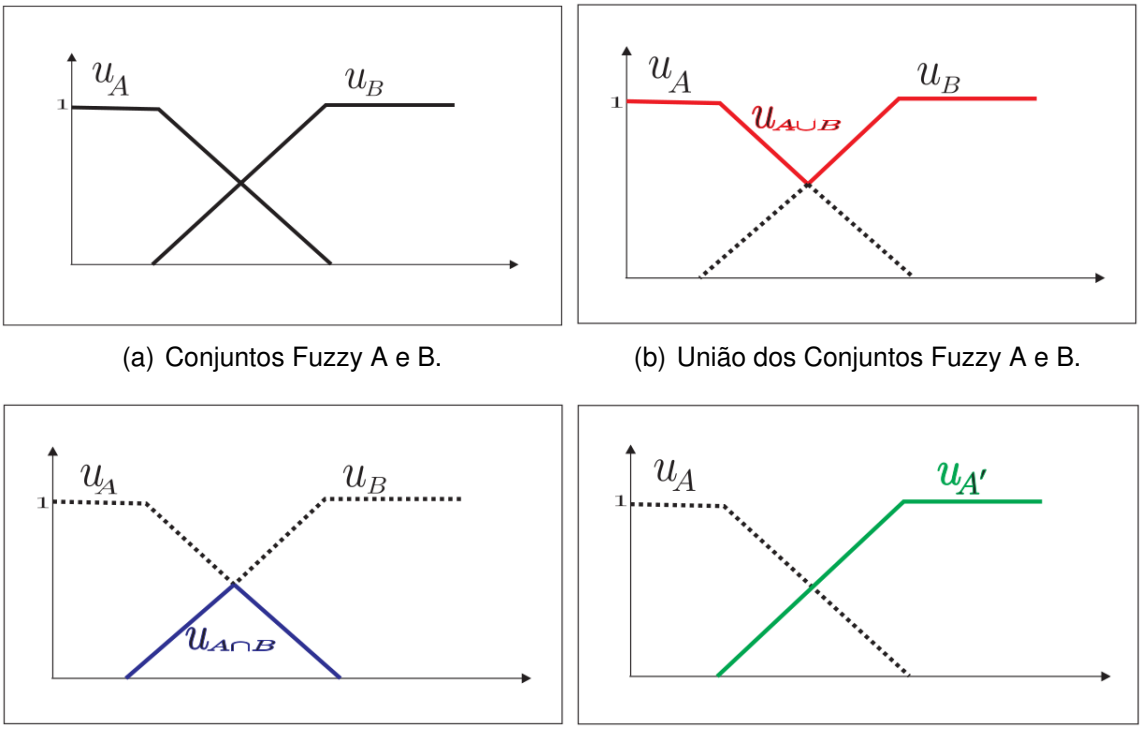
$$A \cup B = \{(\mathrm{x}, \mu_{A\cup B}(\mathrm{x}))|\mathrm{x} \in \chi\}, \text{ onde } \mu_{A\cup B} : \chi \to [0, 1], \mu_{A\cup B}(\mathrm{x}) = \max\{\mu_A(\mathrm{x}), \mu_B(\mathrm{x})\}$$
$$A \cap B = \{(\mathrm{x}, \mu_{A\cap B}(\mathrm{x}))|\mathrm{x} \in \chi\}, \text{ onde } \mu_{A\cap B} : \chi \to [0, 1], \mu_{A\cup B}(\mathrm{x}) = \min\{\mu_A(\mathrm{x}), \mu_B(\mathrm{x})\}.$$

E, o conjunto fuzzy expressando o complemento fuzzy de A em U, é definido por:

$$A^{'} = \{(\mathrm{x}, \mu_{A'}(\mathrm{x}))|\mathrm{x} \in \chi\}, \text{ onde } \mu_{A'} : \chi \to [0, 1], \mu_{A'}(\mathrm{x}) = 1 - \mu_A(\mathrm{x}).$$

Sejam A e B conjuntos fuzzy graficamente representados na Figura 6(a). Ilustrando, funções de pertinência trapezoidais para a união e a intersecção entre A e

B estão respectivamente representadas nas Figuras 6(b) e 6(c). E ainda, veja na Figura 6(d), o conjunto fuzzy complementar de A, no caso considerando a negação fuzzy padrão.



(a) Conjuntos Fuzzy A e B.

(b) União dos Conjuntos Fuzzy A e B.

(c) Interseção dos Conjuntos Fuzzy A e B.

(d) Conjunto Fuzzy A e seu Complementar A$'$.

Figura 6 – Funções de Pertinência das Operações Padrões de Conjuntos Fuzzy
Adaptada de: Wagner; Hagras (2011)

Na sequência, reporta-se o conceito de subconjunto fuzzy, interpretando a noção de relação de inclusão fuzzy.

Sejam A e B dois conjuntos fuzzy. Dizemos que A é um subconjunto fuzzy de B se $\mu_A(\mathrm{x}) \leq \mu_B(\mathrm{x})$, para todo $\mathrm{x} \in \chi$. Ou seja, neste contexto, todo elemento do universo tem grau de pertinência no conjunto A menor que no conjunto B.

Salienta-se que, a função de pertinência do conjunto vazio $\emptyset$ é dada por $\mu_\emptyset(\mathrm{x}) = 0$, $\forall \mathrm{x} \in \chi$. E, para o conjunto universo χ, tem-se que $\mu_\chi(\mathrm{x}) = 1$, $\forall \mathrm{x} \in \chi$.

3.1.2 Definição de α-Nível de Conjuntos Fuzzy

Sejam A um conjunto fuzzy e um escalar $\alpha \in (0, 1]$. Define-se o α-nível de A como o conjunto $[A]^\alpha = \{\mathrm{x} \in U : \mu_A(\mathrm{x}) \geq \alpha\}$. Na Figura 7, tem-se graficamente representado $[A]^\alpha$, para $0 < \alpha \leq 1$, considerando, numa abordagem simplificada, como universo o conjunto de números reais $\mathbb{R}$.

O suporte de um conjunto fuzzy A, indicado por $supp(A)$, é o conjunto *crisp* de todos os elementos de χ que têm grau de pertinência diferente de zero em A, ou seja,

$$supp(A) = \{\mathrm{x} \in \chi : \mu_A(\mathrm{x}) > 0\}.$$

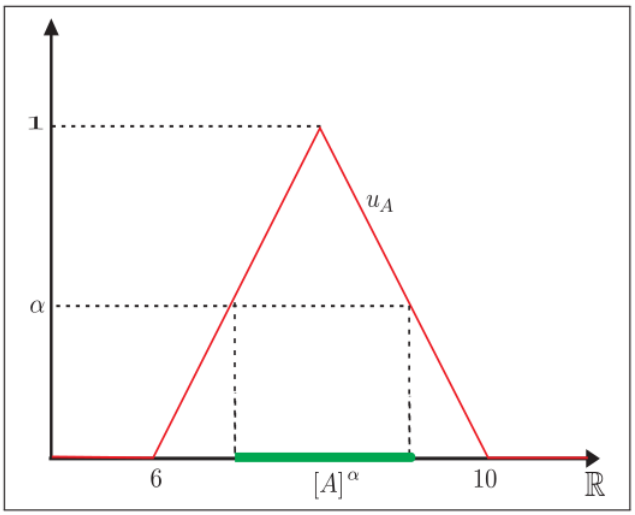


Figura 7 – Representação Gráfica dos α-níveis: $[A]^{\alpha}$ e $[A]^{0} \neq \mathbb{R}$

O suporte de um T1FS A identifica todos os elementos do universo U que possuem alguma associação com A. E ainda, um T1FS cujo suporte contém apenas um único ponto $\mathrm{x} \in U$ e tal que $\mu_A(x) = 1$ é denominado T1FS unitário.

O nível zero de um conjunto fuzzy A constitui o fecho do suporte de A, indicado por $[A]^0 = \overline{suppA}$, ou seja, o menor sub-intervalo fechado de [0,1] contendo $\overline{suppA}$.

Dizemos que um T1FS A é convexo se, e somente se, a condição dada por: $A(\lambda \mathrm{x}_1 + (1-\lambda)\mathrm{x}_2) \leq \min[A(\mathrm{x}_1), A(\mathrm{x}_2)]$, é satisfeita para todo par $(\mathrm{x1}, \mathrm{x2}) \in \chi \times \chi$.

A altura de um T1FS A está definida como o supremo dos valores de sua função de pertinência, ou seja, $\sup_{\mathrm{x} \in U} A(\mathrm{x})$. E, tem-se que o T1FS A é normal se a sua altura é igual a 1, ou seja, $\sup_{\mathrm{x} \in \chi} A(\mathrm{x}) = 1$.

Se um T1FS A é normal, tal fato significa que existe pelo menos um ponto no universo χ esta totalmente compatível com o conceito subjacente modelado por A. Diante dessa interpretação, neste trabalho construímos um modelo fuzzy cujos conjuntos fuzzy estão normalizados.

E finalizando esta sessão, reportam-se que um T1FS A sobre U é chamado de número fuzzy T1 sempre que A é normal, convexo e possui suporte limitado.

3.2 Relações sobre Conjuntos Fuzzy

O conceito de relação em matemática é formalizado como uma associação entre elementos xd dois conjuntos não vazios. Intuitivamente, pode-se dizer que a relação será fuzzy quando se opta pela teoria dos conjuntos fuzzy para conceitualizar a extensão da relação em estudo, no sentido que esta inclui as relações definidas na teoria clássica dos conjuntos (BARROS; BASSANEZI, 2006).

Uma relação clássica indica se há ou não alguma associação entre dois conjuntos de objetos, enquanto que a relação fuzzy além de indicar se existe ou não tal associa-

ção, indica também o grau de pertinência dos pares ordenados do produto cartesiano a esta relação fuzzy, em outras palavras uma relação fuzzy é um conjunto fuzzy tendo como universo um produto cartesiano (BARROS; BASSANEZI, 2006).

Em (BENTKOWSKA; KRÓL, 2015), demonstra-se o contexto das relações fuzzy no que se refere à preservação de suas propriedades no processo de agregação. Outros trabalhos complementam este estudo formal das propriedades algébricas das relações fuzzy, incluindo classes e aplicações (BARRENECHEA et al., 2014; BENTKOWSKA et al., 2015).

Na teoria de conjuntos clássica uma relação n-dimensional R é qualquer subconjunto clássico do produto cartesiano $\chi_1 \times \chi_2 \times \cdots \times \chi_n$ representada por sua função característica $\chi_R : \chi_1 \times \chi_2 \times \cdots \times \chi_n \to \{0, 1\}$, sendo

$$\chi_R(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n) = \begin{cases} 1, & \text{se } (\mathrm{x}_1, \mathrm{x}_2, \ldots \mathrm{x}_n) \in R \\ 0, & \text{caso contrário.} \end{cases}$$

No caso bi-dimensional, $R \subseteq \chi_1 \times \chi_2$ é denominada de relação binária sobre $\chi_1 \times \chi_2$. Se $\chi_1 = \chi_2 = \cdots = \chi_n = \chi$, diz-se que R é uma relação n-ária sobre χ^n.

O conceito matemático de relação fuzzy é formalizado a partir do produto cartesiano usual entre conjuntos clássicos. Considera-se uma função de pertinência de uma relação fuzzy como extensão da função característica de uma relação clássica.

Definição 3 *(BARROS; BASSANEZI, 2006) Uma relação fuzzy está associada a um subconjunto fuzzy em $\chi_1 \times \chi_2 \times \cdots \times \chi_n$, cuja uma função de pertinência $\mu_R : \chi_1 \times \chi_2 \times \cdots \times \chi_n \to [0, 1]$ onde $\mu_R(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n) \in [0, 1]$ indica o grau com que os elementos que compõem uma n-upla $(x_1, x_2, \ldots, x_n)$ estão relacionados por R.*

Do ponto de vista de inferência, com o objetivo de tomar alguma decisão, uma relação fuzzy tem grande importância, principalmente na teoria dos controladores fuzzy (BARROS; BASSANEZI, 2006).

Uma relação fuzzy entre conjuntos A e B é frequentemente definida de forma semelhante à operação de interseção entre A e B aplicada a pares ordenados $(x, y) \in A \times B$. A grande diferença está nos conjuntos universos envolvidos: enquanto na interseção os subconjuntos fuzzy são de mesmo universo; no produto cartesiano, eles podem ser distintos (ROSS, 2010b), de acordo com a Definição 1.

Exemplificação 1 *(BARROS; BASSANEZI, 2006) Sejam os conjuntos fuzzy $A_1, \ldots, A_n$ respectivamente definidos nos universos $\chi_1, \ldots, \chi_n$ não vazios. O produto cartesiano $A_1 \times \cdots \times A_n$ é definido pela função de pertinência, dada por:*

$$\mu_{A_1 \times \cdots \times A_n}(\mathrm{x}_1, \ldots, \mathrm{x}_n) = \min(\mu_{A_1}(\mathrm{x}_1), \ldots, \mu_{A_n}(\mathrm{x}_n)), \tag{1}$$

sendo que $\min : [0, 1]^n \to [0, 1]$ corresponde ao operador de mínimo n-dimensional.

Observa-se que se $A_1, \ldots, A_n$ forem conjuntos clássicos, então o produto cartesiano clássico $A_1 \times \cdots \times A_n$ pode ser obtido pela Equação 1, substituindo as funções de pertinência pelas respectivas funções características dos conjuntos $A_1, \ldots, A_n$.

Sejam $\chi_m = \{\mathrm{x}_1, \ldots, \mathrm{x}_m\}$, $\chi_n = \{\mathrm{y}_1, \ldots, \mathrm{y}_n\}$ e a relação binária fuzzy $R : \chi_n \times \chi_m \to [0, 1]$, cuja função de pertinência é dada por $\mu_R(\mathrm{x}_i, \mathrm{y}_j) = r_{ij} \in [0, 1]$, para cada $1 \leq i, j \leq n$ pode ser representada por uma estrutura tabular e/ou sua correspondente construção matricial, conforme segue:

$\mathcal{R}$	y_1	y_2	$\cdots$	y_n
x_1	r_{11}	r_{12}	$\cdots$	r_{1n}
x_2	r_{21}	r_{22}	$\cdots$	r_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
x_m	r_{m1}	r_{m2}	$\cdots$	r_{mn}

$$\text{ou} \quad \mathcal{R} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}.$$

3.3 Conectivos da Lógica Fuzzy

Na sequência, são apresentados os principais conectivos fuzzy estudados, bem como são analisadas as propriedades e a construção do sistema de inferência baseado em lógica fuzzy.

3.3.1 Negação Fuzzy e Operadores Duais

Nesta seção, são tratadas as propriedades de negações fuzzy, conectivos definidos a partir de operações de complemento entre valores fuzzy no intervalo unitário $[0, 1]$ (ASIAIN et al., 2017). São apresentados, também, exemplos de negação fuzzy com seus respectivos pontos de equilíbrio. Na última subseção, está a definição de função N-dual.

Uma função $N : [0, 1] \to [0, 1]$ é uma **negação fuzzy** se satisfaz as propriedades:

$N1$: $N(0) = 1$ e $N(1) = 0$; (condições de borda)

$N2$: Se $x \leq y$ então $N(x) \geq N(y), \forall x, y \in U$(antitonicidade).

Se N também verifica a propriedade involutiva, esta é chamada de negação forte (BUSTINCE; BURILLO; SORIA, 2003; KLEMENT; MESIAR; PAP, 2004):

$N3$: $N(N(x)) = x, \forall x \in [0, 1]$.

Uma negação fuzzy é chamada estrita quando satisfaz as seguintes propriedades:

$N4$: N é contínua; e

$N5$: Se $x > y$ então $N(x) > N(y), \forall x, y \in [0, 1]$.

Toda negação estrita tem inversa N^{-1} a qual também é uma negação estrita. Observa-se que, negações fortes também são negações estritas, mas o contrário não é verdadeiro (BUSTINCE; BURILLO; SORIA, 2003). Por exemplo, a negação fuzzy $N(x) = 1 - x^2$ é estrita, mas não forte.

Observação 1 *Seja $N : [0,1] \to [0,1]$ uma negação fuzzy. Se uma negação N é forte, então $N = N^{-1}$. Se e é um ponto de equilíbrio (PE) em N, ou seja, $N(e) = e$, então pela antitonicidade de N, para cada $x \in U$, se $x \leq e$ então $e \leq N(x)$ e se $e \leq x$ então $N(x) \leq e$ (BEDREGAL, 2010a). E ainda, todas as negações fuzzy têm no máximo um PE (KLIR, 1993).*

A seguir, exemplos de negações fuzzy e respectivos PE são reportadas.

Exemplificação 2 *Negações correlacionadas aos seus pontos de equilíbrio:*

1. *Negação padrão forte: $N_S : [0,1] \to [0,1]$ dada por: $N_S(x) = 1 - x$. PE: $x = 0,5$;*
2. *Negação forte: $N : [0,1] \to [0,1]$ dada por: $N(x) = \sqrt{1 - x^2}$. PE: $x = \frac{\sqrt{2}}{2} \approx 0,7$;*
3. *Negação: $N_K : [0,1] \to [0,1]$ dada por: $N_K(x) = 1 - x^2$. PE: $x \approx 0,6$.*

Entretanto, nem todas as negações fuzzy possuem um ponto de equilíbrio. Um exemplo é a negação $N_\perp$ definida a seguir:

$$N_\perp(x) = \begin{cases} 0, & \text{se } x > 0, \\ 1, & \text{se } x = 0. \end{cases}$$

3.3.2 Funções de Agregação Fuzzy

Em (BUSTINCE; BARRENECHEA; MOHEDANO, 2004, Definição.2), uma função de *agregação* binária $M : [0,1]^2 \to [0,1]$ satisfaz as seguintes propriedades:

$A1$: $M(0,0) = 0$ e $M(1,1) = 1$;

$A2$: Se $x \leq z$ então $M(x,y) \leq M(z,y)$, $\forall x, y, z \in [0,1]$;

$A3$: $M(x,y) = M(y,x)$, $\forall x, y \in [0,1]$;

Funções de agregação satisfazendo a propriedade de idempotência

$A4$**:** $M(x,x) = x$, $\forall x \in [0,1]$,

são chamadas de *funções de agregação idempotentes*.

Sejam as funções de agregação idempotentes $\wedge, \vee : U^2 \to U$, respectivamente definidas em (DESCHRIJVER; KERRE, 2005, Definição 4.1) pelas expressões:

$$\wedge(x,y) = \min(x,y); \qquad \vee(x,y) = \max(x,y), \forall x, y \in [0,1].$$

Se M é uma função de agregação idempotente, então temos:

$$\wedge(x,y) \leq M(x,y) \leq \vee(x,y), \forall x,y \in [0,1].$$

3.3.2.1 Normas e Conormas Triangulares

As funções de agregação que qualificam as intersecções fuzzy e uniões fuzzy, são geralmente referidas na literatura como t-normas e t-conormas, respectivamente. No contexto da inclusão $\{0,1\} \subseteq [0,1]$, as normas e conormas triangulares são extensões das funções que representam a conjunção e disjunção na lógica clássica, respectivamente.

Definição 4 *(BELIAKOV; PRADERA; CALVO, 2009) Uma **norma triangular** (t-norma) é uma função $T : U^2 \to U$, satisfazendo as seguintes propriedades, para todo $u, v, x, y, z \in U$:*

$T1$*:* $T(x,y) = T(y,x)$ *(Comutatividade);*

$T2$*:* $T(x, T(y,z)) = T(T(x,y), z)$ *(Associatividade);*

$T3$*:* $T(x,y) \leq T(u,v)$*, se* $x \leq u$ *e* $y \leq v$ *(Monotonicidade);*

$T4$*:* $T(x,1) = x$ *(Elemento neutro).*

De acordo com (CALVO; MAYOR; MESIAR, 2002; KAHRAMAN; ÖZTAYŞI; ONAR, 2016) os exemplos mais referenciados e utilizados de normas triangulares são: a (i) Interseção Padrão; (ii) Produto Algébrico; (iii) Interseção Drástica; (iv) Lukasiewicz e (v) Nilpotente Mínimo.

Exemplificação 3 *De acordo com (DUBOIS; PRADE, 2000) a Tabela 1 apresenta os exemplos mais referenciados e utilizados de t-normas.*

Tabela 1 – Exemplificação de Normas Fuzzy Triangulares

Nome	Expressão Algébrica de t-normas
Intersecção-Padrão:	$T_M(x,y) = min\ \{x,y\}$
Produto Algébrico:	$T_P(x,y) = x.y$
Intersecção Drástica:	$T_D(x,y) = \begin{cases} 0, & \text{se } x < 1 \text{ e } y < 1 \\ \min\{x,y\}, & \text{caso contrário;} \end{cases}$
Łukasiewicz:	$T_L(x,y) = \max\{x+y-1, 0\}$
Nilpotente Mínimo:	$T_m(x,y) = \begin{cases} 0, & \text{se } x+y \leq 1 \\ \min\{x,y\}, & \text{caso contrário.} \end{cases}$

Definição 5 *(KLEMENT; MESIAR; PAP, 2000; KAHRAMAN; ÖZTAYŞI; ONAR, 2016) Uma* ***t-conorma triangular*** *(s-norma) é uma função $S : U^2 \to U$, satisfazendo as seguintes propriedades, para todo $u, v, x, y, z \in U$:*

$S1$: $S(x, y) = S(y, x)$ *(Comutatividade);*

$S2$: $S(x, S(y, z)) = S(S(x, y), z)$ *(Associatividade);*

$S3$: $S(x, y) \leq S(u, v)$ *se* $x \leq u$ *e* $y \leq v$ *(Monotonicidade);*

$S4$: $S(x, 0) = x$ *(Elemento neutro).*

Analogamente ao caso da t-norma, de acordo com (DUBOIS; PRADE, 2000) os principais exemplos de t-conormas, são: a (i) União Padrão; (ii) Soma Probabilística; (iii) União Drástica; (iv) Łukasiewicz e (v) Nilpotente Máximo.

Exemplificação 4 *Analogamente, de acordo com (DUBOIS; PRADE, 2000) a Tabela 2 apresenta os principais exemplos de t-conormas.*

Tabela 2 – Exemplificação de Conormas Fuzzy Triangulares

Nome	Expressão Algébrica de t-conormas
União Padrão:	$S_M(x, y) = max\ \{x, y\}$
Soma Probabilística:	$S_P(x, y) = x + y - xy$
União Drástica:	$S_D(x, y) = S_D(x, y) = \left\{ \begin{array}{ll} 1, & \text{se } 0 < x \text{ e } 0 < y \\ \max\{x, y\}, & \text{caso contrário;} \end{array} \right\}$.
Łukasiewicz:	$S_L(x, y) = \min\{x + y, 1\}$
Nilpotente Máximo:	$S_m(x, y) = \left\{ \begin{array}{ll} 1, & \text{se } x + y \geq 1 \\ \max\{x, y\}, & \text{caso contrário.} \end{array} \right\}$.

3.3.3 Implicações Fuzzy

A literatura apresenta uma diversidade de definições e aplicações no assunto de implicações fuzzy e construções duais (BACZYŃSKI, 2004; BACZYŃSKI; JAYARAM, 2008; BALASUBRAMANIAM, 2007; BUSTINCE; BURILLO; SORIA, 2003). Sendo a implicação fuzzy uma generalização da abordagem clássica, o único consenso em suas definições é que a implicação fuzzy deve ter o mesmo comportamento da implicação clássica quando os valores de entrada da implicação forem uma combinação dos extremos do intervalo $[0, 1]$.

No sentido de Fodor e Rubens (FODOR, 1991), um operador binário $I : U^2 \to U$ é uma **implicação fuzzy** se I satisfaz as seguintes condições:

$I1$: $I(1, 1) = I(0, 1) = I(0, 0) = 1$ e $I(1, 0) = 0$ (condições de contorno);

$I2$: $x \leq z \Rightarrow I(x,y) \geq I(z,y)$ (antitonicidade no primeiro argumento); e

$I3$: $y \leq z \Rightarrow I(x,y) \leq I(x,z)$ (isotonicidade no segundo argumento);

Tais propriedades das implicações fuzzy provêm suporte para modelagem do comportamento das regras e composição de métodos de inferência em sistemas de tomada de decisões baseadas na abordagem fuzzy.

3.4 Sistemas Baseados em Regras Fuzzy

Nesta seção as principais características dos sistemas baseados em regras fuzzy são abordadas, as quais reafirmam que estes modelos garantem o suporte para a construção de sistemas de auxílio na tomada de decisões em ambientes de incertezas, como o caso do processo de classificação do tráfego de rede.

3.4.1 Visão Geral

Com o incremento na complexidade de um sistema, a habilidade de fazer declarações precisas e significativas sobre o seu comportamento diminui, até alcançar um limite além do qual precisão e relevância tornam-se características mutuamente exclusivas.

Conforme Zadeh (ZADEH, 1973), a transcrição acima é definida como o Princípio da Incompatibilidade, e mostra a relevância em utilizar a lógica fuzzy para auxiliar na resolução de problemas que tradicionalmente são difíceis de resolver. A ideia básica de um sistema fuzzy é considerar funções que mapeiam um valor escalar em um número limitado entre 0 e 1, indicando o grau de pertinência desse valor ao conjunto.

Um sistema fuzzy pode estimar funções de entrada e saída, por meio do uso de técnicas heurísticas, onde um ou mais especialistas humanos, entrevistados para ajudar a formular o conjunto de regras fuzzy, pode articular associações de entrada/saída linguísticas.

Assim, sistemas fuzzy podem produzir estimativas de um sistema complexo a partir de variáveis linguísticas familiares da linguagem natural, mas fundamentais em modelos matemáticos. Nesse escopo, a metodologia fuzzy é um método de estimação de entrada e saída que considera modelos matemáticos (SHAW, 1999).

Um sistema de inferência considera os seguintes blocos principais (ROSS, 2010a):

- **BR (Base de regras)**, contendo as regras/proposições fuzzy onde as variáveis antecedentes/consequentes são VL (Variáveis Linguísticas) e os possíveis valores de uma VL são representados por conjuntos fuzzy;
- **Base de dados**, definindo as funções de pertinência dos conjuntos fuzzy nas regras fuzzy;

- **Unidade de decisão lógica**, realizando operações de inferência para obter, a partir da avaliação dos níveis de compatibilidade das entradas, com as condições impostas pela BR, uma ação a ser realizada pelo sistema;
- **Interface de fuzzificação**, utilizando as funções de pertinência pré-estabelecidas mapeia cada variável de entrada do sistema em graus de pertinência de cada conjunto fuzzy que representa a variável em questão;
- **Interface de defuzzificação**, transformando os resultados fuzzy produzidos pela saída da inferência obtidos no módulo da unidade de decisão lógica, com o emprego das funções de pertinência das VL da parte consequente das regras, obtém uma saída não fuzzy; E, nesta etapa, as regiões resultantes são convertidas em valores de saída do sistema.

3.4.2 Componentes da Arquitetura

Esta seção descreve, brevemente, os componentes da arquitetura de sistemas baseados na lógica fuzzy, e suas possíveis extensões, como a T2FL (Lógica Fuzzy Tipo-2) e a IT2FL (Lógia Fuzzy Tipo-2 Intervalar) também denominada IVFL (Lógica Fuzzy Valorada Intervalarmente). A Figura 8 apresenta um esquema gráfico de sistemas abrangendo regras fuzzy tipo-2 que manipulam dados de T2FS. Os sistemas baseados em regras fuzzy tipo-1 podem ter estrutura análoga ao fuzzy tipo-2, exceto pelo módulo de redução de tipo, que atua na saída da inferência sobre IVFS reduzindo um IT2FS para um T1FS.

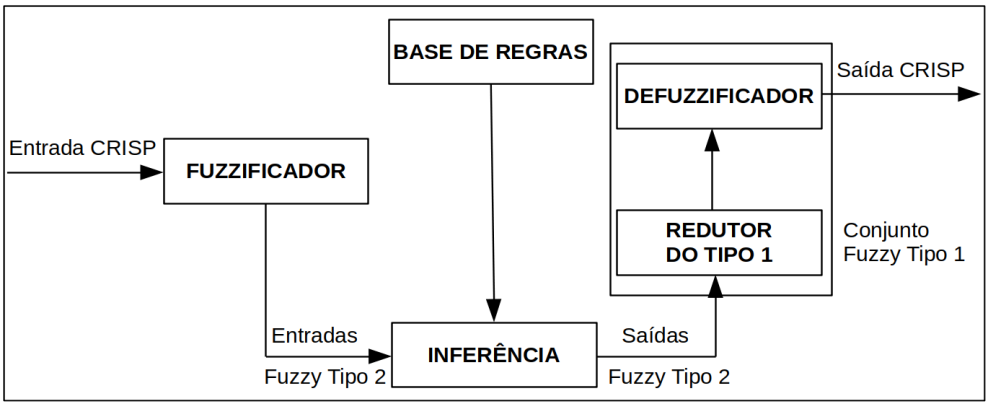


Figura 8 – Arquitetura de um Sistema de Inferência Fuzzy
Adaptada de: Mendel (2007)

Fuzzificação

O processo de fuzzificação é um mapeamento de subconjuntos de números reais, em geral discretizados, para o domínio fuzzy. A fuzzificação também indica que

há atribuição de valores linguísticos, descrições vagas ou qualitativas, definidas por funções de pertinência às variáveis de entrada.

A fuzzificação também pode indicar uma espécie de *pré-processamento* de categorias ou classes dos sinais de entrada, reduzindo o número de valores a serem processados. Uma menor quantidade de valores processados significa menos complexidade das computações. As funções de pertinência também podem ser descritas por tabulação de valores numéricos, e consultas via tabelas podem acelerar a etapa de fuzzificação.

Resumindo, são utilizadas duas sub etapas:

- (i) a entrada é um valor numérico; e
- (ii) a saída correspondendo a um número do intervalo real [0,1].

E ambas são dependentes dos graus de pertinência na definição dos conjuntos fuzzy. Para cada valor de entrada é aplicada uma função de pertinência, a qual retornará o valor da avaliação lógica da proposição.

Regras e Inferência

As regras podem ser fornecidas por especialistas, em forma de sentenças linguísticas, e se constituem em um aspecto fundamental no desempenho de um sistema de inferência fuzzy.

No caso de controlador fuzzy, o bom desempenho está condicionado às regras que descrevem a estratégia de controle de forma consistente. Extrair regras de especialistas na forma de sentenças condicionais é uma tarefa difícil, por mais conhecedores que eles sejam do problema em questão.

Alternativamente ao uso de especialistas para a definição da base de regras, existem métodos de extração de regras de dados numéricos úteis em problemas de classificação e previsão de séries temporais (KLIR, 2005).

No estágio de inferência, ocorrem as operações com conjuntos fuzzy propriamente ditas: combinação dos antecedentes das regras, implicação e *modus ponens* generalizado. Os conjuntos fuzzy de entrada, relativos aos antecedentes das regras, e o de saída, referente ao consequente, podem ser definidos previamente ou, alternativamente, gerados automaticamente a partir dos dados.

Nesta etapa, tem-se a aplicação dos operadores fuzzy, sendo que a entrada consta de dois ou mais valores, resultantes da fuzzificação. No caso da aplicação do operador de implicação, há uma remodelação nos dados de entrada pela aplicação de uma implicação fuzzy.

Na sequência, tem-se a agregação dos resultados das inferências, onde são justapostas todas as saídas fuzzy em um único conjunto fuzzy.

Para a elaboração dessas regras é importante ter em mente alguns conceitos importantes. São eles:

- Variáveis Linguísticas: elas são o centro da técnica de modelagem de sistemas fuzzy. Com elas é possível nomear os conjuntos, e ainda, qualificá-los utilizando os qualificadores tais como muito (*High*), regular (*Average*) e pouco (*Low*). Dessa forma, a modelagem do sistema se torna mais próxima do mundo real.
- Conexões lógicas: do tipo AND/OR, para criar a relação entre as variáveis.
- Implicações do tipo: Se "x_1 é A_1" E "x_2 é A_2" E "x_3 é A_3" então "y é B"

Os principais sistemas de inferência fuzzy estão baseados em dois operadores. Para tal, sejam A e B conjuntos fuzzy: Existem dois tipos básicos de implicações fuzzy (SHAW, 1999):

1. *Modus Ponens* (Modo Afirmativo), operando frequentemente em controladores fuzzy e sistemas especialistas.

$$
\begin{array}{rll}
 & \text{Premissa 1:} & X = A \\
modus\ ponens: & \text{Premissa 2:} & se\ X = A\ então\ Y = B \\
 & \text{Consequência:} & Y = B
\end{array}
$$

   onde $A \subset X\ e\ B \subset Y$.

2. *Modus Tollens:* (Modo Negativo), operam com base em premissas ou condições, as quais geram uma determinada consequência. Tal tipo está sendo utilizado somente em sistemas especialistas.

$$
\begin{array}{rll}
 & \text{Premissa 1:} & Y = não - B \\
modus\ tollens: & \text{Premissa 2:} & se\ X = A\ então\ Y = B \\
 & \text{Consequência:} & X = não - A
\end{array}
$$

O mapa de regras fuzzy relaciona as entradas fuzzy entre si, gerando as saídas fuzzy correspondentes, formando assim a base de conhecimento do sistema. As entradas do mapa são preenchidas durante a identificação do sistema fuzzy quando um operador humano, ou ainda, um sistema especialista auxilia na identificação da operação e controle do processo.

Na inferência fuzzy, também se indicam como as regras são agregadas e combinadas, provendo a construção da região resultante da aplicação das regras.

Na agregação, ou seja, na composição dos vários conjuntos fuzzy de entrada em uma regra, as *t*-normas (T_M e T_P) são mais comuns, enquanto que na combinação,

ou composição das saídas fuzzy de cada regra, a *t*-conormas S_m e S_P têm sido as mais praticadas (ROSS, 2010a).

Assim, definem-se as estruturas *max-min* ou *max-produto* para controladores fuzzy. Produto (P) e min (M) são ambos operadores de interseção fuzzy, referidos como conectivos *AND* (BELIAKOV; PRADERA; CALVO, 2009).

Defuzzificador

No defuzzificador, o valor da VL de saída inferida pelas regras fuzzy será traduzido num valor discreto, ou seja, geometricamente as regiões resultantes do processo de inferência são convertidas em valores precisos para a variável de saída do sistema. O objetivo é obter um único valor numérico discreto (*crisp*) que melhor represente os valores fuzzy inferidos da VL de saída (ROSS, 2010c).

A defuzzificação é uma transformação inversa que traduz a saída do domínio fuzzy para o domínio discreto. Para selecionar o método apropriado de defuzzificação, pode-se utilizar um enfoque baseado no centroide ou nos valores máximos que ocorrem da função de pertinência resultante.

Existem diversas técnicas de defuzzificação, as principais delas são: centro da área, centro do máximo e média do máximo (SHAW, 1999), resumidas na sequência desta seção.

(i) Centro da Área (CoA - *Center of Area*) O método centro da área, também conhecido como centro de gravidade, calcula o centroide da área composta que representa o termo de saída fuzzy (μ_{OUT}), esse termo é composto pela união de todas as contribuições de regras. O centroide é um ponto que divide a área de μ_{OUT} em duas partes iguais. O cálculo do CoA se dá da seguinte forma:

$$\mathrm{x} = \frac{\sum_{i=1}^{N} \mathrm{x}_i \mu_{OUT}(\mathrm{x}_i)}{\sum_{i=1}^{N} \mu_{OUT}(\mathrm{x}_i)} \tag{2}$$

onde $\mu_{OUT}(\mathrm{x}_i)$ é a área de uma função de pertinência modificada pelo resultado da inferência fuzzy, e x_i é a posição do centroide da função de pertinência individual. Tal equação calcula o centroide composto, para o qual contribuem as funções de pertinência indicadas.

(ii) Centro do Máximo (CoM - *Center of Maximum*)

Neste método, os picos das funções de pertinência representados no universo de discurso da variável de saída são usados, enquanto ignoram-se as áreas das funções de pertinência, as contribuições múltiplas de regras são consideradas por esse método.

Os valores não nulos do vetor de possibilidades de saída são posicionados nos correspondentes picos. Assumindo que tais valores representam pesos, o valor de saída defuzzificado (discretizado) é determinado pelo ponto de apoio onde os pesos ficam equilibrados. Assim, as áreas das funções de pertinência não desempenham papel relevante, apenas os máximos são utilizados. A saída discreta é calculada como uma média ponderada dos máximos, cujos pesos são os resultados da inferência. O cálculo do valor defuzzificado é realizado por meio da seguinte equação:

$$\mathrm{x} = \frac{\sum_{i=1}^{N} \mathrm{x}_i \sum_{k=1}^{n} \mu_{O,k}(\mathrm{x}_i)}{\sum_{i=1}^{N} \sum_{k=1}^{n} \mu_{O,k}(\mathrm{x}_i)} \tag{3}$$

onde $\mu_{O,k}(u_i)$ indicam os pontos em que ocorrem os máximos (alturas) das funções de pertinência de saída.

Essa abordagem representa um melhor compromisso entre possíveis saídas com multiplicidade de disparo de conjuntos fuzzy, ou seja, se três regras forem acionadas, duas impondo uma saída e uma impondo outra saída, por exemplo.

(iii) Média do Máximo (MoM - *Mean of Maximum*) O método Média do Máximo é utilizado em casos onde a função de pertinência tenha mais de um máximo, pois a abordagem CoM não funcionaria bem, devido à necessidade de escolher qual máximo utilizar. E, a média de todos os máximos (MoM), é dada pela expressão:

$$\mathrm{x} = \sum_{m=1}^{M} \frac{\mathrm{x}_m}{M} \tag{4}$$

onde u_m é o m-ésimo elemento no universo do discurso, pressupõe que a função $\mu_{OUT}(u_i)$ tenha um máximo e M é o número total desses elementos. Esse método também é conhecido como solução mais plausível, pelo fato de desconsiderar o formato das funções de pertinência de saída.

3.5 Controladores Fuzzy

Sejam A, B subconjuntos fuzzy dos conjuntos χ_1 e χ_2, respectivamente, e $T : [0,1]^2 \to [0,1]$ uma t-norma contínua à esquerda (KLEMENT; MESIAR; PAP, 2000). Os seguintes são equivalentes:

(i) Existe uma relação fuzzy R em $\chi_1 \times \chi_2$ que resolve a equação relacional $A \circ_T R = B$

(ii) A expressão analítica para a relação fuzzy $R_T(A,B)$ em $\chi_1 \times \chi_2$ dada pela equação relacional $A \circ_T R = B$ está indicada por:

$$\mu_{R_T(A,B)}(x,y) = \mu_A(x) \to_T \mu_B(y). \tag{5}$$

Considerando uma t-norma T contínua à esquerda (KLEMENT; MESIAR; PAP, 2000) e os subconjuntos fuzzy A e B em χ_1 e χ_2, a relação fuzzy $R_T(A, B)$ é uma solução prototípica da equação relacional $A \circ_T R = B$. Na verdade, o valor $\mu_{R_T}(A, B)(x, y)$ pode ser interpretado pela BR :

$$\text{Se } x \text{ é } A \text{ então } y \text{ é } B$$

é frequentemente chamada de regra (linguística) SE ... ENTÃO. Portanto, faz sentido usar uma solução de $A \circ_T R = B$ para descrever a relação entre A e B.

3.5.1 Controlador Fuzzy de Mamdani-Assilian

Em (MAMDANI; ASSILIAN, 1975) foi definido um primeiro tipo de controlador fuzzy que utiliza conjuntos fuzzy tanto no espaço de entrada quanto no de saída e, portanto, pelo menos em um contexto técnico, geralmente necessita de uma defuzzificação para produzir uma função de entrada-saída.

Consideram-se espaços de entrada arbitrários, geralmente multi-dimensionais, e para o espaço de saída são restringidos ao caso mais usual $Y = \mathbb{R}$. Justificam-se estas escolhas, pois espaços de saída multi-dimensionais podem ser tratados considerando vários controladores fuzzy, com múltiplas saídas em paralelo.

Em aplicações e situações práticas, as condições da Definição 6 a seguir, devem ser verificadas.

Definição 6 *Sejam X um espaço de entrada arbitrário, $A_1, A_2, \ldots, A_n$ e $B_1, B_2, \ldots, B_n$ conjuntos fuzzy em χ e $\mathcal{R}$ uma relação definida por pares de funções de pertinência μ_{A_i} e μ_{B_i}, para $1 \leq i \leq n$, e uma t-norma $T : [0, 1]^2 \to [0, 1]$. E, seja a BR gerada por*

$$\textit{Se } \mathrm{x} \textit{ é } A_i \textit{ então } \mathrm{y} \textit{ é } B_i, \forall \mathrm{x}, 0 \leq \mathrm{x} \leq 1.$$

A função $F_M : \chi \to \mathbb{R}$ modelando o controlador de Mamdani é definida por:

$$F_M(\mathrm{x}) = \frac{\int_{\mathbb{R}} \mu_{\mathcal{R}}(\mathrm{x}, \mathrm{y}).\mathrm{y} \ \ d\mathrm{y}}{\int_{\mathbb{R}} \mu_{\mathcal{R}}(\mathrm{x}, \mathrm{y}) d\mathrm{y}} \tag{6}$$

sempre que $\int_{\mathbb{R}} \mu_{\mathcal{R}}(x, y)dy > 0, \forall \mathrm{x} \in \chi$, e a relação $\mathcal{R}$ definida pela função de pertinência $\mu_{\mathcal{R}} : \chi \times \chi \to [0, 1]$, a qual é expressa por:

$$\mu_{\mathcal{R}}(\mathrm{x}, \mathrm{y}) = \max\left(T(\mu_{A_1}(\mathrm{x}), \mu_{B_1}(\mathrm{y})), \ldots, T(\mu_{A_n}(\mathrm{x}), \mu_{B_n}(\mathrm{y}))\right),$$

com $\mu_{A_i}(\mathrm{x}), \mu_{B_i}(\mathrm{x}) \in [0, 1]$, para $\forall i, 1 \leq i \leq n$.

Em (KLEMENT; MESIAR; PAP, 2013, Definição 14), tem-se aplicação do método especial de defuzificação, o centro de gravidade, que está basicamente contido na

Equação 6. Oportuno destacar, que também existem outros métodos de defuzzificação, por exemplo, a média dos máximos (KRUSE; GEBHARDT; KLAWONN, 1994).

As t-normas mais amplamente usadas para o controlador Mamdani são T_M ou T_P:

- (i) no primeiro caso, também referida como $\max - \min$ inferência; e
- (ii) no outro caso, tem-se a $\max$-prod inferência.

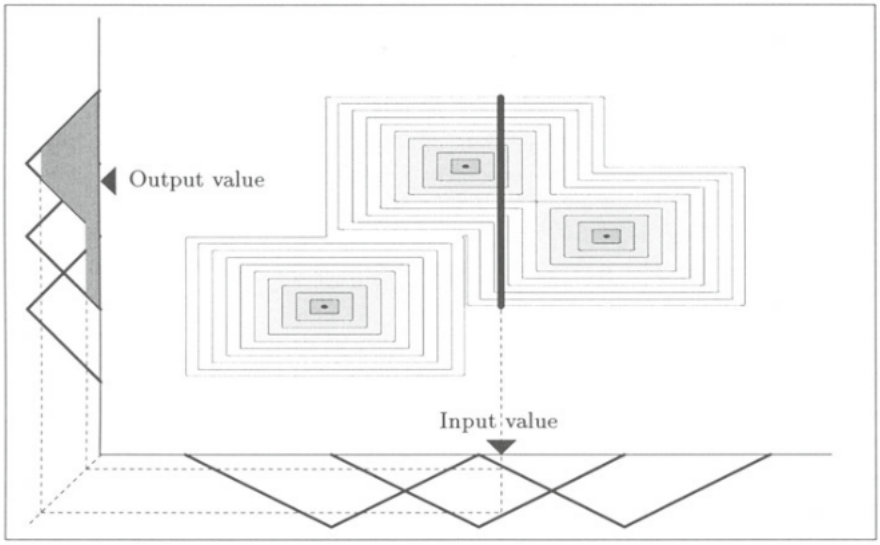


Figura 9 – Visão Geral do Controlador de Mamdani
Adaptada de: Klement; Mesiar; Pap (2013)

No controlador de Mamdani apresentado na Figura 9, os conjuntos fuzzy do espaço entrada/saída são mostrados na parte inferior e esquerda, respectivamente. E, a relação fuzzy induzida pelo controlador está representada pelo gráfico de contorno; o valor de saída é o centro da área em relação à região cinza, apresentada na esquerda desta região.

3.5.2 Controlador Fuzzy de Takagi-Sugeno

Outro controlador fuzzy muito usado em aplicações de tomoda de decisão é o controlador de Takagi-Sugeno (TAKAGI; SUGENO, 1985; SUGENO, 1985) que usa valores crisp no espaço de saída. E, a inferência inclui engloba a etapa de defuzzificação.

Definição 7 *Sejam U um espaço de entrada, $A_1, \ldots, A_n$ subconjuntos fuzzy de U com $\sum_{i=1}^{n} \mu_{Ai}(\mathrm{x}) > 0$ para todo $\mathrm{x} \in U$, e as funções $f_1, \ldots, f_n : U \to \mathbb{R}$. Te-se a BR:*

$$\textit{Se } x \textit{ é } A_i \textit{ então } y \textit{ é } f_i(x), \forall 1 \leq i \leq n.$$

A função de entrada-saída $F_{TS} : X \to \mathbb{R}$ dado por

$$F_{TS}(\mathrm{x}) = \frac{\sum_{i=1}^{n} \mu_{Ai}(\mathrm{x}).f_i(\mathrm{x})}{\sum_{i=1}^{n} \mu_{A_i}(\mathrm{x})}$$

define o controlador Takagi-Sugeno.

Relacionamentos entre os controladores são explorados em (KLEMENT; MESIAR; PAP, 2013). Em especial, se cada função f_i é constante, ou seja, $f_i(x) = u_i$, $\forall i = 1, \ldots, n$, o controlador Takagi-Sugeno é um caso especial do controlador de Mamdani.

3.6 Considerações do Capítulo

Neste Capítulo foram abordados tópicos referentes a teoria dos T1FS, tratando definição, diferenciação de função característica e pertinência, incluindo o estudo das relações e operações padrões entre conjuntos fuzzy de união, de interseção e o complementação. Também foram revisados os conceitos como α-níveis, suporte, convexidade, normalidade de um conjunto fuzzy.

De mesmo modo, foram sistematizados conceitos relativos aos conectivos fuzzy, focando em negações, funções de agregação (normas e conormas triangulares), implicações, e finalmente, os principais tópicos que definem um sistema de inferência fuzzy foram apontados, incluindo as etapas de fuzzificação, inferência e defuzzificação. Estes conceitos serão estendidos, como suporte a fundamentando para extensão intervalar da lógica fuzzy, considerada como a abordagem lógica nesta Tese.

4 LÓGICA FUZZY VALORADA INTERVALARMENTE

"Parece um teorema sem ter
demonstração,
E parece que sempre termina,
Mas não tem fim ".

Teorema
Legião Urbana

Este Capítulo sumariza o estudo feito com relação a teoria de conjuntos fuzzy tipo-2 (T2FS), destacando os aspectos mais relevantes da classe de conjuntos fuzzy valorada intervalarmente (IVFS), os quais são também denominados conjuntos fuzzy tipo-2 intervalares (IT2FS), considerando o foco de estudo e pesquisa desta Tese.

4.1 Contextualização Histórica de Conjuntos Fuzzy Tipo-2

A teoria de conjuntos fuzzy tipo-2 foi introduzida por Lotfi Zadeh em (ZADEH, 1975) como uma extensão dos conjuntos fuzzy tradicionais (T1FS). Seu surgimento está relacionado com a insuficiência da teoria T1FS tradicional em modelar, por apenas um número no intervalo unitário, apenas as incertezas inerentes à definição das funções de pertinência dos antecedentes e consequentes em um sistema de inferência fuzzy.

A consolidação da teoria de conjuntos fuzzy recebeu significativas contribuições, desde sua introdução (SAMBUC, 1975; ZADEH, 1975), evoluindo em abordagens teóricas como nos trabalhos pioneiros (GORZALCZANY, 1989; TURKSEN, 1986; TURKSEN; ZHONG, 1990) e mais recentemente .eCDK04, DeschrijverK05a.

A abordagem lógica baseada em T2FS tem-se mostrado eficiente em circunstâncias onde há necessidade de lidar com imprecisão dos dados em computação, incluindo interpretação da hesitação, ambiguidade e indeterminação de vários especialistas em relação à pertinência de múltiplos atributos, além das incertezas já modeladas via abordagem fuzzytradicional (TAKÁČ, 2014).

Aplicações baseadas na teoria IVFS são cada vez mais considerados na modelagem de problemas, considerando as restrições na interpretação via conjuntos

fuzzy (BUSTINCE et al., 2016) e o suporte provido pela matemática intervalar. Para um melhor entendimento sobre toda esta discussão veja (HERNÁNDEZ; CUBILLO; TORRES-BLANC, 2022; SOLA et al., 2015)

Os estudos reportam, entre outros fatores, que IVFS fornecem uma representação da imprecisão via diâmetro do intervalo, além da modelagem da incerteza da linguagem natural, inerente à construção dos vários cenários de uma aplicação via abordagem fuzzy. Novas metodologias (SANZ; BUSTINCE, 2021) e teorias que fazem uso de ordens admissíveis viabilizaram, recentemente, não apenas a preservação do diâmetro dos dados intervalares mas também promovem a representação e comparação dos resultados intervalares nas aplicações científicas e tecnológicas (TAKÁC et al., 2019; CRUZ ASMUS et al., 2022).

Aliam-se a estas relevantes argumentações, a característica de explicabilidade dos controladores fuzzy, provida pelo sistema de inferência baseada em regras condicionais (CHOI; MUN; AHN, 2012; JURIO et al., 2011; SANZ et al., 2013a).

A visibilidade das regras selecionadas durante a execução dos controladores fuzzy mostram aos usuários, como os resultados extraídos do sistema foram alcançados a partir dos dados de entrada. Tem-se, portanto maior segurança via argumentação lógica, para explicar a partir dos atributos selecionados e das variáveis linguísticas definidas por especialistas, como os resultados das execuções foram gerados (BARRENECHEA et al., 2011, 2014).

Esta Tese considera ainda a integração entre uma abordagem baseada em IVFS (Lógica Fuzzy Multivalorada) com a eficiência consolidada das metodologias baseadas em aprendizagem de máquina (BENTKOWSKA et al., 2015; BURILLO; BUSTINCE, 1996). Neste contexto, o estudo teórico, via IVFL (Lógica Fuzzy Valorada Intervalarmente), está direcionado à fundamentação necessária para desenvolvimento de métodos que contribuam para a classificação do tráfego de rede, mais especificamente na identificação dos fluxos de rede do tipo *streaming* de vídeo.

4.2 Conceituação de Conjuntos Fuzzy Tipo-2

Os conjuntos fuzzy tipo-2 são conjuntos fuzzy cujos graus de pertinência dos elementos do universo U são conjuntos fuzzy tradicionais, e não apenas um único valor pontual (MENDEL, 2007; PEDRYCZ, 2021).

Na teoria dos T2FS, a função de pertinência tipo-2 está totalmente precisa, provendo uma interpretação flexível baseada na “mancha” de incerteza, ou seja, FOU (*Foot Print of Uncertainty*) , a qual é tratada pelo sistema de inferência em T2FS.

Um conjunto fuzzy tipo-2 (T2FS) $\tilde{A}$ sobre U, é caracterizado por uma função de pertinência do tipo-2, indicada por $\mu_{\tilde{A}}(\mathrm{x}, u)$, $\forall \mathrm{x} \in U$, $u \in [0, 1]$.

No contexto desta Tese, o grau de pertinência passa a ser não somente um con-

junto de pares ordenados expressando a função de pertinência que modela um T1FS. Entende-se que, o grau de pertinência de cada elemento $\mathrm{x} \in U$, caracteriza-se por um subintervalo fechado do intervalo unitário (BUSTINCE et al., 2016).

Os conjuntos fuzzy assim gerados são conjuntos fuzzy tipo-2 intervalares (IT2FS), ou ainda, conjuntos fuzzy valorados intervalarmente(IVFS), descritos na Definição 8.

Definição 8 *Um T2FS $\tilde{A}$ é caracterizado pela função de pertinência $\mu_{\tilde{A}}(\mathrm{x}, u)$, ou seja:*

$$\tilde{A} = \{((\mathrm{x}, u), \mu_{\tilde{A}}(\mathrm{x}, u)) : \forall \mathrm{x} \in U, \forall u \in J_{\mathrm{x}} \subset [0, 1]\},$$

sendo que $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

4.2.1 Conceitos Relevantes

Na Definição 8, tem-se que $\mathrm{x} \in \chi$ e $u \in U$ são chamadas de **variável primária** e **variável secundária** do T2FS $\tilde{A}$, respectivamente. E, J_{x} é um T1FS chamado de **pertinência primária** de $\mathrm{x} \in \chi$ em $\tilde{A}$.

E ainda, a restrição $u \in J_{\mathrm{x}} \subseteq [0, 1]$ é consistente com a restrição em T1FS, ou seja, quando as incertezas desaparecem em uma função de pertinência do tipo 2, esta se transforma em uma função de pertinência do tipo 1. Neste caso, a variável u transforma-se em $\mu_A(\mathrm{x})$, onde $0 \leq \mu_A(\mathrm{x}) \leq 1$. A restrição $0 \leq \mu_{\tilde{A}}(\mathrm{x}, u) \leq 1$ é consistente com o fato de que as amplitudes de função de pertinência em T2FS também assumirem valores no intervalo unitário $[0, 1]$.

A representação gráfica da função de pertinência $\mu_{\tilde{A}}(\mathrm{x}, u)$ é uma função tridimensional, cuja imagem está representada no eixo perpendicular ao plano gerado pelos eixos $\overrightarrow{\mathrm{x}}$ e $\overrightarrow{u}$, no qual estão representados os pares (x, u), com entradas associados às variáveis primária e secundária, respectivamente.

Fixado $\mathrm{x}' \in \chi$, um corte vertical no gráfico da função de pertinência do tipo 2, gerando um plano com eixos definidos por $\overrightarrow{u}$ e $\mu_{\tilde{A}}(\mathrm{x}', u)$ é uma função de **pertinência secundária**, indicada por $\mu_{\tilde{A}}(\mathrm{x} = \mathrm{x}', u) = \mu_{\tilde{A}}(\mathrm{x})$ definida por T1FS. E adicionalmente, verifica-se que o domínio da função de pertinência secundária coincide com a pertinência primária J_{x}, para cada $\mathrm{x} \in \chi$. O grau secundário é o valor de pico maior amplitude assumida em cada corte, para $\mu_{\tilde{A}}(\mathrm{x})$.

O T2FS mais amplamente estudado é chamado Conjunto Fuzzy Tipo-2 Intervalar (IVFS ou IT2FS). No contexto deste trabalho, também considera-se os conjuntos fuzzy valorados intervalarmente, e os correspondentes conectivos fuzzy valorados intervalarmente são estudados na sequência.

4.2.2 Conjuntos Fuzzy Valorados Intervalarmente

Pela Definição 8, um IVFS está definido ao tomar $\mu_{\tilde{A}}(\mathrm{x}, u) = 1$, $\forall \mathrm{x} \in \chi$ e $u \in J_{\mathrm{x}}$, sendo $J_{\mathrm{x}} \subseteq [0, 1]$, $\forall \mathrm{x} \in \chi$ (SOLA et al., 2015).

Neste caso, o suporte da variável secundária u, $J_{\mathrm{x}} = \{u \in [0,1] \colon \mu_{\tilde{A}}(\mathrm{x},u) > 0\}$, com $\mathrm{x} \in \chi$, pode ser dado pela expressão $J_{\mathrm{x}} = \left[\underline{\mu}_{\tilde{A}}(\mathrm{x},u), \overline{\mu}_{\tilde{A}}(\mathrm{x},u)\right]$ sendo que:

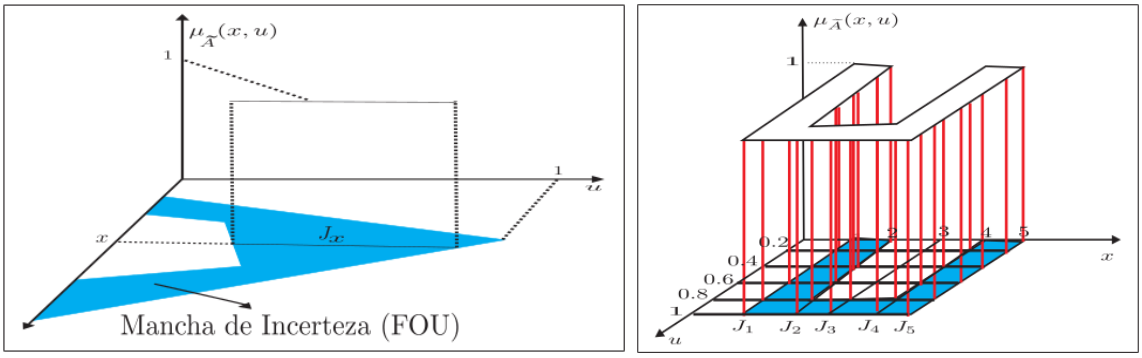
$$\underline{\mu}_{\tilde{A}}(\mathrm{x},u)) = \inf\{u \colon u \in [0,1], \mu_{\tilde{A}}(\mathrm{x},u) > 0\}, \tag{7}$$

$$\overline{\mu}_{\tilde{A}}(\mathrm{x},u)) = \sup\{u \colon u \in [0,1], \mu_{\tilde{A}}(\mathrm{x},u) > 0\}. \tag{8}$$

Para cada IVFS, tem-se que a FOU é definida por:

$$FOU(\tilde{A}) = \{(\mathrm{x},u) \colon \mathrm{x} \in \chi, u \in [\underline{\mu}_{\tilde{A}}(\mathrm{x},u)), \overline{\mu}_{\tilde{A}}(\mathrm{x},u))]\}.$$

As Figuras 10(a), e 10(b) mostram a representação gráfica tridimensional da função de pertinência de um IVFS e suas respectivas áreas coloridas em azul representam FOU, a região de incerteza (MENDEL, 2007).



(a) (b)

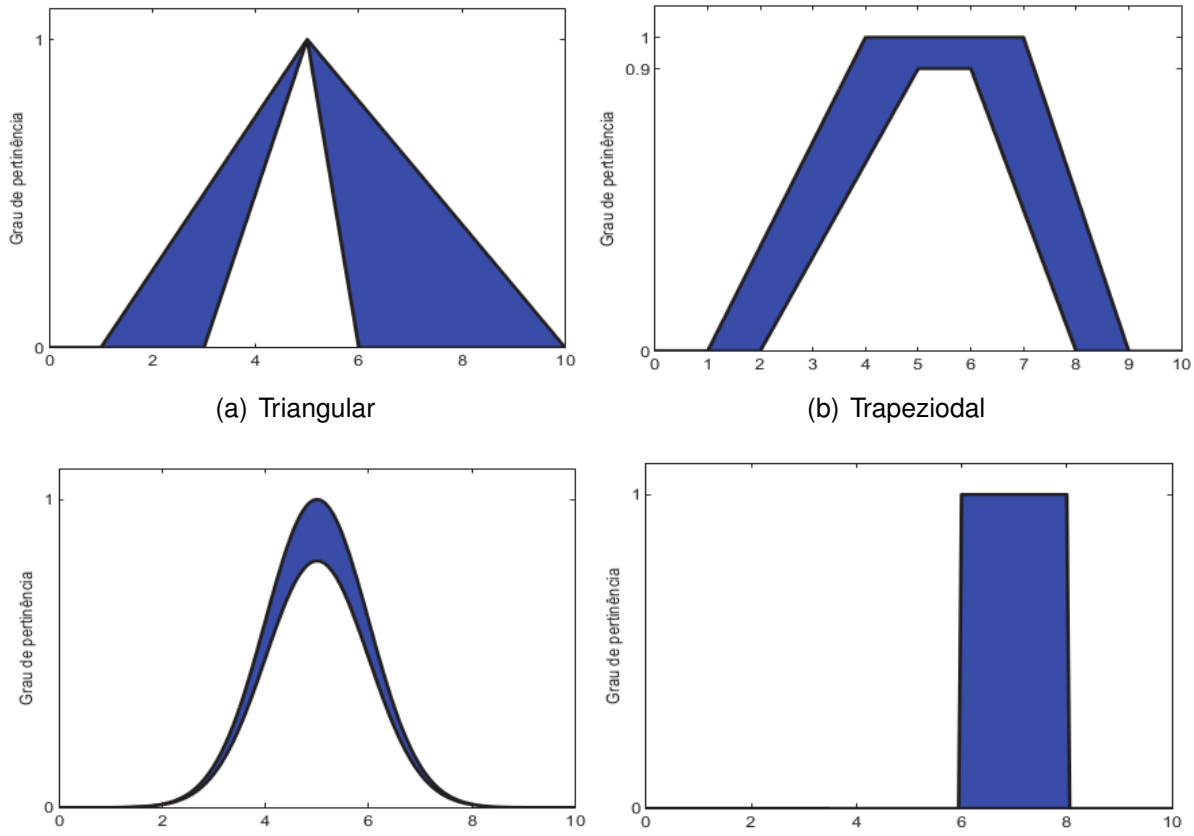
Figura 10 – Representação dos Conjuntos Fuzzy Valorados Intervalarmente
Adaptada de: Mendel (2007)

Como o grau secundário dos conjuntos fuzzy valorados intervalarmente sempre é igual a 1 em J_x, e igual a 0 para o restante dos pontos do plano que define J_{x}, tem-se que a terceira dimensão acaba não mostrando nenhuma informação adicional ao intervalo J_x. As Figuras 11(a), 11(b), 11(c) e 11(d), apresentam conjuntos fuzzy valorados intervalarmente, triangular, trapezoidal, gaussiano e “singlenton”, respectivamente.

Definição 9 *O corte vertical de $\mu_{\tilde{A}}(x,u)$ é definido como sendo o plano bidimensional em um dado $x = x'$, cujos eixos são u e $\mu_{\tilde{A}}(x',u)$.*

A função de pertinência secundária é o corte vertical de $\mu_{\tilde{A}}(x,u)$ em determinado valor de $x = x'$. Como mostrado na Figura 12.

Definição 10 *A pertinência primária J_{x} de x, é definida como o domínio da função de pertinência secundária para o valor de x, com $J_{\mathrm{x}} = [\underline{J}_{\mathrm{x}}, \overline{J}_{\mathrm{x}}] \subseteq [0,1]$, $\forall \mathrm{x} \in U$ (MENDEL, 2007).*



(a) Triangular (b) Trapeziodal

(c) Gaussiano (d) Singleton

Figura 11 – Exemplos de Conjuntos Fuzzy do Tipo-2 Intervalares
Adaptada de: Castro; Castillo; Melin (2007)

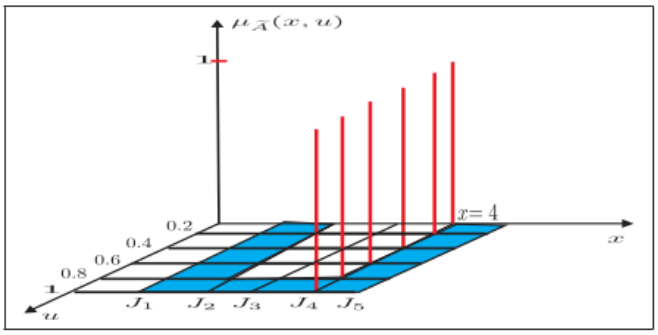


Figura 12 – Função de Pertinência Secundária Intervalar em x=4
Adaptada de: Mendel (2007)

Definição 11 *A "mancha" de incerteza (FOU) é definida como a união de todas as pertinências primárias, isto é,*

$$FOU(\tilde{A}) = \bigcup_{x \in \chi} (x, J_x) \tag{9}$$

consistindo em um conjunto fuzzy do tipo-2 definido por funções de pertinência do tipo-1, a superior e a inferior (TAKÁČ, 2013), sendo a incerteza representa pela região limitada por estas (MENDEL, 2007; CASTILLO; MELIN, 2012a,b; CASTILLO; MELIN;

PEDRYCZ, 2011).

Tem-se, a seguir, novas expressões na abordagem intervalar para as funções descritas pelas Eq.7 e Eq.8 :

Em (MENDEL, 2007), a função de pertinência inferior é representada na forma $\underline{\mu}_{\tilde{A}}(\mathrm{x}), \forall(\mathrm{x}) \in \chi$, dados por:

$$\underline{\mu}_{\tilde{A}}(\mathrm{x}) = \bigcup_{\mathrm{x}\in\chi}(x, \underline{J}_{\mathrm{x}}) \tag{10}$$

onde $\bigcup$ representa o operador de união.

A Figura 13 exemplifica uma FOU com suas funções de pertinência superior e inferior para abordagem intervalar.

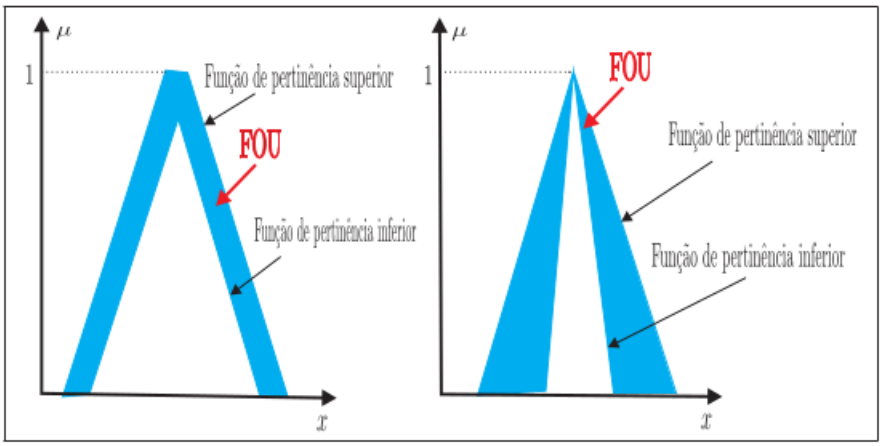


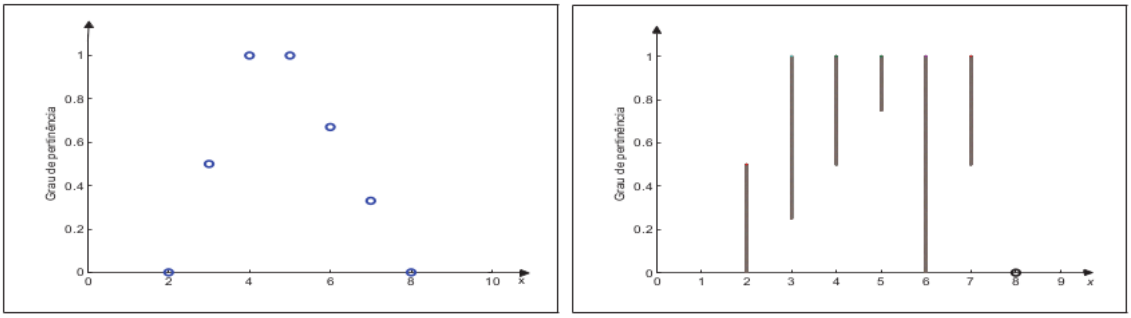
Figura 13 – Conjunto Fuzzy Valorado Intervalarmente
Adaptada de: Mendel (2007)

4.2.3 Relações entre Conjuntos Fuzzy e Tipo-2 Intervalar

Um exemplo de um conjunto fuzzy do tipo-1, apresentado na Figura 14(a). Quando apenas os números inteiros são considerados no domínio x, o conjunto fuzzy do tipo-1 pode ser representado como $\{0/2,\ 0.5/3,\ 1/4,\ 1/5,\ 0.67/6,\ 0.33/7,\ 0/8\}$, em que $0/2$ significa que o número 2 possui grau de pertinência 0 no conjunto fuzzy do tipo-1.

Um exemplo das pertinências primárias de um conjunto fuzzy do tipo-2, intervalar e discreto, está exposto na Figura 14(b).

Pode-se observar que, ao contrário de um conjunto fuzzy do tipo-1, cujas pertinências para cada $x \in U$ é um número, as pertinências de um conjunto fuzzy do tipo-2 intervalar é um intervalo. Por exemplo, as pertinências primárias dos números 2, 3, 4, 5, 6 e 7 são intervalos $[0, 0.5], [0.25, 1], [0.5, 1], [0.75, 1], [0, 1]$ e $[0.5, 1]$, respectivamente, e o grau de pertinência primário do número 8 é 0. Neste contexto, um conjunto fuzzy tipo-2 estende um conjunto fuzzy tipo-1 quando consideram-se valores degenerados para os graus de pertinência da função primária $J_x(\overline{J}_x = \underline{J}_x)$.



(a) Conjunto Fuzzy do Tipo 1

(b) Pertinência Primária de um Conjunto Fuzzy do Tipo-2 Intervalar Discreto

Figura 14 – Comparação de Conjuntos Fuzzy do Tipo 1 e 2 Intervalares

4.3 Lógica Fuzzy Valorada Intervalarmente

A Lógica Fuzzy Valorada Intervalarmente (IvFL - *Interval-valued Fuzzy Logic*) é considerada com base na teoria dos Conjuntos Fuzzy Valorados Intervalarmente (IvFS - *Interval-valued Fuzzy Sets*), permitindo especificar apenas um subintervalo fechado do intervalo unitário $[0, 1]$ como o grau de pertinência de cada elemento em um conjunto fuzzy (GEHRKE; WALKER; WALKER, 1996).

Assim, ao complementar a teoria dos conjuntos fuzzy, a teoria dos conjuntos fuzzy valorados intervalarmente pode modelar ambos, a incerteza e a imprecisão, característica de não especificidade, refletida no diâmetro do intervalo associado ao grau de pertinência.

4.3.1 Ordens Parciais em IVFS

Seja o conjunto de todos os intervalos reais fechados no intervalo unitário $[0, 1]$, ou seja, $L([0, 1]) = \{[a, b] : 0 \leq a \leq b \leq 1\}$.

Um conjunto fuzzy IvFS A em U é uma função $\mu_{\mathbb{A}} : U \to L([0, 1])$ (PEKALA, 2019; STARCZEWSKI, 2013; REISER; BEDREGAL; REIS, 2014). E, o conjunto de todos os conjuntos fuzzy valorados intervalarmente é indicado por $\mathcal{F}_{L(0,1)}$.

Para lidar com variáveis intervalares são consideradas as projeções $l_{L([0,1])}, r_{L([0,1])} : L([0, 1]) \to [0, 1]$ que são, respectivamente, definidas como $l_{L([0,1])}([x_1, x_2]) = x_1$ e $r_{L([0,1])}([x_1, x_2]) = x_2$. Nestes casos, os limites de $X \in L([0, 1])$, $l_{L([0,1])}(X)$ e $r_{L([0,1])}(X)$, também são denotados por $\underline{X}$ e $\overline{X}$, respectivamente.

Para cada $x \in [0, 1]$, o intervalo degenerado $[x, x] \in L([0, 1])$ é denotado por $\mathbf{x}$, sendo $\bar{D} = \{\mathbf{x} = [x, x] \colon x \in U\}$ o conjunto de todos os intervalos degenerados em $L([0, 1])$.

Seja a ordem de *Kulisch-Miranker order* também chamada de *ordem do produto*:

$$X \leq_{L([0,1])} Y \Leftrightarrow \underline{X} \leq \underline{Y} \wedge \overline{X} \leq \overline{Y}, \ \forall X, Y \in L([0, 1]);$$

e, portanto $\mathbf{0} \leq_{L([0,1])} X \leq_{L([0,1])} \mathbf{1}, \forall X \in L([0,1])$.

O reticulado dos valores fuzzy intervalares é constituído pelo conjunto $L([0,1])$, munido com a ordem parcial do produto, tendo $\mathbf{0}$ como *bottom* e $\mathbf{1}$ como topo e sendo o ínfimo e supremo de cada par $X, Y \in l([0,1])$ dado da seguinte forma

$$X \wedge Y = [\min(\underline{X}, \underline{Y}), \min(\overline{X}, \overline{Y})] \text{ e } X \vee Y = [\max(\underline{X}, \underline{Y}), \max(\overline{X}, \overline{Y})]. \tag{11}$$

Nesta sequência, as principais propriedades de negações e agregadores fuzzy valorados intervalarmente são abordados, considerando ordem parcial usual de Kulish-Miranker no reticulado $\big(L([0,1]), \leq_{L([0,1])}\big)$.

4.3.2 Negações Fuzzy Valoradas Intervalarmente

Os complementos de IvFS são definidos por negações fuzzy valoradas intervalarmente. Uma função $\mathcal{N}\colon L([0,1]) \to L([0,1])$ é uma negação fuzzy Valorada Intervalarmente (IvFN - *Interval-valued Fuzzy Negation*) se as seguintes propriedades são verificadas:

$\mathbb{N}1$ $\mathcal{N}([0,0]) = [1,1]$ e $\mathcal{N}([1,1]) = [0,0]$ (condições de contorno);

$\mathbb{N}2$ $X \leq Y$ implica que $\mathcal{N}(Y) \leq \mathcal{N}(X), \forall X, Y \in L([0,1])$ (antimonotônica pela ordem do produto).

Além disso, uma IvFN $\mathcal{N}$ é forte se é involutiva, verificando

$\mathbb{N}3$ $\mathcal{N}(\mathcal{N}(X)) = X, \forall X \in L([0,1])$.

A extensão intervalar da negação padrão é dada por $\mathcal{N}_S(X) = [1 - \overline{X}, 1 - \underline{X}]$.

Lema 1 *(BEDREGAL; MEZZOMO; REISER, 2018, Prop. 3.9) Seja $\mathcal{N}$ uma IvFN forte. Então, para cada $X, Y \in L([0,1])$, $\mathcal{N}(X \vee Y) = \mathcal{N}(X) \wedge \mathcal{N}(Y)$ e $\mathcal{N}(X \wedge Y) = \mathcal{N}(X) \vee \mathcal{N}(Y)$.*

Em (PEKALA, 2019; WU; LUO, 2011), dado um IvFS $\mathbb{A}$, seu complemento com relação a ordem do produto é o IvFS $\mathbb{A}^c$, definido para cada $z \in U$, como segue:

$$\mathbb{A}^c = \{(z, \chi_{\mathbb{A}^c}(z)\colon z \in U\}.$$

Sempre que $\chi_{\mathbb{A}}(z) = X$ então $\chi_{\mathbb{A}^c}(z) = \mathbb{N}(X)$. Se ainda, a negação valorada intervalarmente $\mathbb{N}$ é representável por uma negação fuzzy N tal que $\mathbb{N}(X) = N(x) > x \in X$, então $\mathbb{N}(X) = [N(\overline{X}), N(\underline{X})] = [1 - \overline{X}, 1 - \underline{X}]$. Em particular quando $N = N_S$ então $\mathbb{N}(X) = [1 - \overline{X}, 1 - \underline{X}]$.

Além disso, em (ASIAIN et al., 2018; BEDREGAL, 2010b; BUSTINCE et al., 2020) outras ordens, totais admissíveis, são estudadas (BEDREGAL; MEZZOMO; REISER, 2018; PALMEIRA et al., 2014; REISER; BEDREGAL, 2017) para negações, em extensões fuzzy mais gerais.

4.3.3 Agregações Fuzzy Valoradas Intervalarmente

Uma função $\mathcal{A} : L([0,1])^n \to L([0,1])$ é uma Função de Agregação Valorada Intervalarmente (IVAF - *Interval-valued Aggregation Function*) se verifica as propriedades (ZAPATA et al., 2017):

$\mathbb{A}1$ $\mathcal{A}([0,0],\ldots,[0,0]) = [0,0]$ e $\mathcal{A}([1,1],\ldots,[1,1]) = [1,1]$ (preserva as condições de contorno)

$\mathbb{A}2$ Para $X_1,\ldots,X_n,Y \in L([0,1])$ e $i \in \{1,\ldots,n\}$, se $X_i \leq Y$, $\mathcal{A}(X_1,\ldots,X_n) \leq \mathcal{A}(X_1,\ldots,X_{i-1},Y,X_{i+1},\ldots,X_n)$ (monotônica pela ordem do produto)

Pode-se também considerar outras propriedades que podem ser verificadas para algumas IVAF. Seja $\mathcal{F} : L([0,1])^2 \to L([0,1])$ uma IVAF, para todo $X,Y,Z \in L([0,1])$, então temos:

$\mathbb{A}3$ $\mathcal{F}(X,Y) = \mathcal{F}(Y,X)$ (comutatividade);

$\mathbb{A}4$ $\mathcal{F}(X,\mathcal{F}(Y,Z)) = \mathcal{F}(\mathcal{F}(X,Y),Z)$ (associatividade);

$\mathbb{A}5$ $\exists E \in L([0,1])$ s.t. $\mathcal{F}(X,E) = \mathcal{F}(E,X) = X$ (elemento E-neutro);

$\mathbb{A}6$ $\mathcal{F}(X,X) = X$ (idempotência);

Diversas classes de IVAF bivariadas foram definidas na literatura com base em propriedades selecionadas e relacionadas à aplicação realizada. De acordo com as quatro classes relatadas anteriormente, pode-se ter, por exemplo: a classe das t-normas valoradas intervalarmente (IVAF conjuntiva), com elemento neutro $E_S = [1,1]$, e na construção dual, sendo $E_T = [0,0]$, as t-conormas de valoradas intervalarmente (IVAF disjuntiva). (BUSTINCE et al., 2013; DESCHRIJVER, 2013; PEKALA, 2019).

4.3.4 Conectivos Fuzzy e Ordens Admissíveis em IVFS

Na concepção da abordagem FuzzyNetClass, empregam-se metodologias e métricas de comparação e análise de resultados baseadas em ordens admissíveis (ZAPATA et al., 2017), considerando a necessidade de comparar os intervalos produzidos como saída para o sistema de inferência fuzzy da abordagem FuzzyNetClass, visando a classificação do tráfego de vídeo, com modelagem baseada em T2FS.

A expectativa com o emprego de ordens admissíveis é contornar as situações onde dois intervalos possam ser entendidos como incomparáveis por métodos usuais de ordenação dos intervalos reais. Um exemplo neste sentido seria a ordem produto "$\leq_{L([0,1])}$", que é uma relação de ordem parcial, não total, e portanto, permite que dois resultados intervalares possam ser incomparáveis ($X \not\leq_{l([0,1])} Y$ e $Y \not\leq_{l([0,1])} X$).

Ordens lineares em $L([0,1])$ são relações binárias reflexivas, antissimétricas, transitivas e totais. Ou seja, uma ordem linear em $L([0,1])$ é uma ordem onde quaisquer dois pares de subintervalos do intervalo unitário $[0,1]$ são comparáveis.

Uma ordem parcial $\leq_{L([0,1])}$ em $L([0,1])$ pode ser estendida por uma ordem admissível $\preceq_{L([0,1])}$, sempre que a ordem $\preceq_{L([0,1])}$ for linear (total) e preservar as relações já estabelecidas pela ordenação parcial $\leq_{L([0,1])}$.

As classes de ordens admissíveis, são atualmente estudadas em muitas abordagens que fazem uso da lógica fuzzy valorada intervalarmente (COSTA et al., 2019; MATZENAUER et al., 2022), e os resultados recentes já garantem a ordenação total e a preservação dos diâmetros dos dados intervalares (BUSTINCE et al., 2020). Para fundamentação desta Tese, a definição dos conectivos considerando estas propostas de estudos será brevemente caracterizada e exemplificada logo a seguir.

Exemplificação 5 *Em (TAKÁC et al., 2019), tem-se exemplificação de ordens admissíveis com relação a ordem de produto (Kulish-Miranker), geradas por operadores de agregação:*

1. *Sejam $M_1, M_2 : [0,1]^2 \to [0,1]$ funções de agregação fuzzy tais que, $\forall X, Y \in [0,1]$, as igualdades $M_1(\underline{X}, \overline{X}) = M_1(\underline{Y}, \overline{Y})$ e $M_2(\underline{X}, \overline{X}) = M_2(\underline{Y}, \overline{Y})$ valem, simultaneamente, somente se $X = Y$. Tem-se então que a relação*

$$X_{M_1,M_2}Y \Leftrightarrow \begin{cases} M_1(\underline{X}, \overline{X}) < M_1(\underline{Y}, \overline{Y}); \textit{ou} \\ M_1(\underline{X}, \overline{X}) = M_1(\underline{Y}, \overline{Y}) \textit{ e } M_2(\underline{X}, \overline{X}) \leq M_2(\underline{Y}, \overline{Y}). \end{cases} \tag{12}$$

   é uma ordem admissível em $L([0,1])$.

2. *Seja $\alpha \in [0,1]$ e considere a função de agregação $K\alpha(x,y) = (1-\alpha)x + \alpha y$. De acordo com a Eq.(12), quando $\alpha, \beta \in [0,1]$ com $\alpha \neq \beta$, obtém-se uma ordem linear admissível $\preceq_{\alpha,\beta}$, referente a ordem produto , considerando $M_1(x,y) = K_\alpha(x,y)$ e $M_2(x,y) = K_\beta(x,y)$.*

3. *Em particular, pela Eq.(12), se $M_1(x,y) = (x+y)/2$ e $M_2(x,y) = y$, tem-se a ordem de Xu and Yager (XU; YAGER, 2006) em $L([0,1])$, dada pela expressão:*

$$[\underline{X}, \overline{X}] \leq_{XY} [\underline{Y}, \overline{Y}] \Leftrightarrow \begin{cases} \underline{X} + \overline{X} < \underline{Y} + \overline{Y}; \textit{ou} \\ \underline{X} + \overline{X} = \underline{Y} + \overline{Y} \textit{ e } \overline{X} - \underline{X} \leq \overline{Y} - \underline{Y}, \forall\, X, Y \in \mathbb{U}. \end{cases} \tag{13}$$

4. *As ordens Lexicográficas $\leq_{Lex1}$ relacionada a primeira variável, e $\leq_{Lex2}$ a segunda*

variável, são respectivamente definidas pelas expressões:

$$[\underline{X}, \overline{X}] \leq_{Lex1} [\underline{Y}, \overline{Y}] \Leftrightarrow \begin{cases} \underline{X} < \underline{Y}; \textit{ ou} \\ \underline{X} = \underline{Y} \textit{ e } \overline{X} \leq \underline{Y}; \end{cases} \tag{14}$$

$$[\underline{X}, \overline{X}] \leq_{Lex2} [\underline{Y}, \overline{Y}] \Leftrightarrow \begin{cases} \overline{X} < \overline{Y}; \textit{ ou} \\ \overline{X} = \overline{Y} \textit{ e } \underline{X} \leq \underline{Y}. \end{cases} \tag{15}$$

Neste caso, tem-se também um caso particular da Eq.(12), considerando $M_1(x, y) = x$ *e* $M_2(x, y) = y$ *para* $\leq_{Lex1}$*, e a correspondente projeção reversa para* $\leq_{Lex2}$*.*

Uma revisão ampla, explorando as principais propriedades das ordens admissíveis e conectivos definidos no seu contexto, incluindo aquelas já descritas acima, está disponível em (BUSTINCE et al., 2013, 2020; SANTANA et al., 2020).

Os conectivos lógicos no reticulado $(L([0,1], \preceq_{L([0,1)])})$, são definidos de forma análoga a $(L([0,1], \leq_{L([0,1)])})$, considerando as propriedades satisfeitas pela ordenação total provida pela ordem admissível $\preceq_{L([0,1)])}$ que refina a ordem parcial $\leq_{L([0,1)])}$.

Na sequência são apresentadas as negações e agregações em $(L([0,1]$ aplicadas no modelo FuzzyNetClass, as quais foram definidas considerando a ordem admissível de Xu-Yager, $\preceq_{XY})$.

Definição 12 *De acordo com (ZAPATA et al., 2017), sejam* $c = \frac{\underline{X}+\overline{X}}{2}$*,* $\alpha = \wedge(c, 1-c)$ *e* $r = \frac{\overline{X}-\underline{X}}{2}$*. A função* $\mathbb{N}_{XY} : L([0,1]) \to (L[0,1])$ *dada pela expressão:*

$$\mathbb{N}(X)_{XY} = [(1-c) - (\alpha - r), (1-c) + (\alpha - r)] \tag{16}$$

is é uma IVFN forte referente à ordem Xu-Yager. E, sua expressão pode ser descrita da seguinte forma:

$$\mathbb{N}_{XY}(X) = \begin{cases} \left[1 - \frac{\overline{X}+3\underline{X}}{2}, 1 - \frac{\overline{X}-\underline{X}}{2}\right], & \textit{if } \overline{X}+\underline{X} \leq 1\ ; \\ \left[\frac{\overline{X}-\underline{X}}{2}, 2 - \frac{3\overline{X}+\underline{X}}{2}\right], & \textit{otherwise,} \end{cases} \tag{17}$$

com ponto de equilíbrio $E_{XY} = [\frac{1}{4}, \frac{3}{4}]$*.*

Definição 13 *A função* $\mathbb{M}_n : L([0,1])^n \to L([0,1])$ *dada pela expressão*

$$\mathbb{M}_n(\overrightarrow{X}, \overrightarrow{Y}) = \begin{cases} \mathbf{0}, \textit{ if there exist } X_i = \mathbf{0}, 0 \leq i \leq n, \\ \left[\frac{1}{n}\Sigma_{i=1}^{n}\underline{X_i}, \frac{1}{n}\Sigma_{i=1}^{n}\overline{X_i}\right], \textit{ otherwise;} \end{cases} \tag{18}$$

é uma agregação com relação a ordem admissível de Xu-Yager ($\preceq_{XY}$*).*

Na definição seguinte, a função EN definida por propriedades análogas à entropia valoradas intervalarmente, mas restrita aos valores do reticulado $(L([0,1], \preceq_{L([0,1)])})$ dos valores fuzzy intervalares considerando a ordem linear $\preceq_{L([0,1)])})$.

4.3.5 Entropia Intervalar

O conceito de entropia, associado ao processo de comunicação ou informação clássica refere-se aos processos onde há perda, desorganização de informação, ou ainda, uma situação na qual os processos ganham entropia. Assim, as interpretações fuzzy consideram a modelagem da incerteza no cálculo da entropia, e quanto maior as informações sobre um sistema, menor será sua entropia.

No caso da abordagem fuzzy intervalar, tem-se agregado a medida da imprecisão da informação no cálculo da entropia fuzzy, modelando a falta de conhecimento preciso de especialistas sobre o grau de pertinência de um elemento, ou seja, quando seu grau de pertinência é interpretado por um intervalo, definido por uma função de pertinência intervalar.

Por outro lado, a saída de uma função de intervalo pode exibir menos ou mais incerteza do que suas entradas. Neste contexto, mostra-se interessante a análise da largura do intervalo de pertinência de varáveis de entrada e de saída, que devem estar relacionadas pela preservação dos intervalos referentes a todo informação do processo, variáveis e operadores aplicados na modelagem.

No contexto desta Tese, considera-se o conceito de entropia intervalar, modelando a imprecisão dos dados de entrada e preservando tal informação imprecisa, via o diâmetro dos intervalos, durante as computações até obter as saídas. Este conceito de entropia é construído por agregando funções e funções-normais EN. E ainda, permitem a comparação e/ou ordenação dos resultados intervalares por aplicarem o conceito de ordens ordem admissíveis lineares.

Esta abordagem intervalar para entropia apresentada logo a seguir, será aplicada como métrica na validação dos conjuntos fuzzy gerados para os atributos de entrada e de saída, definindo suporte para o raciocínio aproximado modelado para abordagem FuzzyNetClass.

Para conceitos adicionais em IVFS, veja (ASIAIN et al., 2018; TAKÁC et al., 2019; ASIAIN et al., 2017; BENTKOWSKA et al., 2015) e ainda, mais recentemente, os resultados apresentados em (SANTANA et al., 2020; BUSTINCE et al., 2020)

(TAKÁC et al., 2019, Def.5) Seja $\mathbb{N} : L([0,1]) \to L([0,1])$ uma IVFN forte referente à ordem total $\leq_{TL}$ com ponto de equilíbrio ε ($N(\varepsilon) = \varepsilon$). Uma função $EN_{IV} : \mathbb{U} \to \mathbb{U}$ que satisfaz seguintes propriedades:

1. $EN_{IV}(\varepsilon) = [1 - \omega(\varepsilon), 1]$;
2. $EN_{IV}(X) = 0_L$ se, e somente se, $X = 0_L$ or $X = 1_L$;

3. Se $Y \leq_{TL} X \leq_{TL} \varepsilon$ ou $Y \geq_{TL} X \geq_{TL} \varepsilon$ onde $w(X) = w(Y)$ então $EN_{IV}(X) \geq_{TL} EN_{IV}(Y)$;

EN_{IV} é uma função-normal EN valorada intervalarmente referente à IVFN $\mathbb{N}$.

Exemplificação 6 *Seja $\leq_{XY}$ a ordem admissível de Xu-Yager dada pela Eq. 13, $\mathbb{N}_{XY} : L([0,1]) \to L([0,1])$ uma IVFN forte referente à ordem $\leq_{XY}$, dada pela Eq. 17.*

A função $EN_{IV} : L^2([0,1]) \to L([0,1])$ definida por:

$$EN_I(X) = [1 - |2M(X) - 1|^p - W(X), 1 - |2M(X) - 1|^p] \tag{19}$$

onde Para $p \in [1, \infty)$ e $M(X) = \frac{X+\overline{X}}{2}$ e $W(X) = \frac{\overline{X}-X}{2}$ é uma função-normal EN valorada intervalarmente, referente a $\mathbb{N}_{XY}$.

Definição 14 *Seja $\preceq_{TL}$ uma ordem total, $\mathbb{N} : L([0,1]) \to L([0,1])$ uma IVFN forte referente à ordem total $\preceq_{TL}$. A função $E : \mathcal{F}_{L(0,1)} \to L(0,1)$ é uma função de entropia valorada intervalarmente (IVFE) referente a IVFN $\mathbb{N}$ sempre que, para todo $A, B \in \mathcal{F}$ as seguintes propriedades são satisfeitas*

(E1) $E(A) = \mathbf{0}$ *se, e somente se, A é um conjunto crisp;*

(E2) $E(\tilde{\epsilon}) = 1 - W(\epsilon)$*;*

(E3) $E(A) \preceq_{TL} E(B)$ *se* $W(A(\mathrm{x})) = W(B(\mathrm{x}))$ *e* $W(A(\mathrm{x})) \preceq_{TL} W(B(\mathrm{x})) \preceq_{TL} \epsilon$ *ou* $W(A(\mathrm{x})) \preceq_{TL} \geq W(B(\mathrm{x})) \preceq_{TL} \epsilon$*.*

Proposição 1 *Sejam um conjunto universo $\chi_n = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\}$. Sejam a ordem total $\leq_{TL}$, sobre a qual tem-se definidas uma IVFN forte $\mathbb{N} : L([0,1]) \to L([0,1])$ e uma agregação fuzzy valorada intervalarmente $\mathbb{M} : L([0,1])^n \to L([0,1])$ que satisfaz :*

$\mathbb{A}7$ $\mathbb{M}(X, X, \ldots, X) = XS$, $\forall X \in L([0,1])$*;*

$\mathbb{A}8$ $\mathbb{M}(X_1, X_2, \ldots, X_n) = \mathbf{0}$ *se, e somente se,* $X_1 = X_2 = \ldots = X_n = \mathbf{0}$*.*

Seja EN_I uma função-normal valorada intervalarmente referente à IVFN $\mathbb{N}$. Então a função $E_{IV} : \mathcal{F}_{L([0,1])} \to L([0,1])$ definida por:

$$E_{IV}(\tilde{A}) = \mathbb{M}\left(\mu_{\tilde{A}}(\mathrm{x}_1), \mu_{\tilde{A}}(\mathrm{x}_2), \ldots, \mu_{\tilde{A}}(\mathrm{x}_n)\right), \tilde{A} \in \mathcal{F}_{L([0,1])} \tag{20}$$

é uma entropia valorada intervalarmente referente a $\mathbb{N}$, considerando a ordem total $\leq_{TL}$.

Exemplificação 7 *Seja $\tilde{A} \in \mathcal{F}_{L([0,1])}$ com conjunto universo $\chi_n = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\}$. Considere $\leq_{XY}$ a ordem admissível dada pela Eq. 13, $\mathbb{N}_{XY} : L([0,1]) \to L([0,1])$ uma*

negação fuzzy valorada intervalarmente forte referente à ordem total $\leq_{XY}$*, dada pela Eq. 17. Para* $p \in [1, \infty)$*, a função* $EN_{IV} : \mathcal{F}_{L([0,1])} \to L([0,1])$ *definida por:*

$$EN_{IV}(\tilde{A}) = \mathbb{M}_{i=1}^{n} \left([1 - |2(X_M) - 1|^p - W(X), 1 - |2(X_M)) - 1|^p] \right), \tag{21}$$

é uma função de entropia valorada intervalarmente, referente a $\mathbb{N}_{XY}$*, sempre que* $M(X) = \frac{\underline{X}+\overline{X}}{2}$ *e* $W(X) = \frac{\overline{X}-\underline{X}}{2}$ *onde* $\mu_{\tilde{A}}(\mathrm{x}) = [\underline{X}, \overline{X}] \in L([0,1])$*,* $\forall \mathrm{x} \in \chi_n$*.*

4.4 Considerações do Capítulo

Este Capítulo apresentou conceitos e definições referentes aos conjuntos fuzzy do tipo-2, salientando algumas diferenças entre os conjuntos fuzzy tipo-1 e tipo-2. Destaca-se que o emprego dos conjuntos fuzzy tipo-2 está relacionado com a insuficiência da teoria de conjunto fuzzy tradicional em modelar as incertezas e imprecisões inerentes à definição das funções de pertinência dos antecedentes e consequentes em tomadas de decisões.

E, de forma mais específica para a proposta desta Tese, este Capítulo abordou os fundamentos relativos a lógica fuzzy valorada intervalarmente, conectivos (negações e agregações) e relações, destacando assuntos relacionados às ordens parciais em conjuntos fuzzy valoradas intervalarmente, reticulados de valores fuzzy intervalares e entropia intervalar.

5 ESTADO DA ARTE EM CLASSIFICAÇÃO DO TRÁFEGO DE REDES

> “E o que há algum tempo era jovem
> novo, Hoje é antigo,
> e precisamos todos rejuvenescer”.
>
> Velha Roupa Colorida
> Belchior

Este Capítulo discute a RAL (Revisão Assistemática da Literatura), desenvolvida durante o processo de concepção desta Tese de doutorado. O resultado central desta revisão foi a seleção dos principais trabalhos de pesquisa encontrados no tema desta Tese.

O princípio da abordagem para a revisão de literatura foram sobre trabalhos que abordavam classificação do tráfego de rede com métodos clássicos. A partir dos trabalhos encontrados, houve o direcionamento para buscas sobre aprendizagem de máquina em classificação de tráfego de rede. Devido ao foco no problema de pesquisa buscou-se trabalhos que abordassem modelos que tratassem incerteza e imprecisão havendo destaque para lógica fuzzy valorada intervalarmente como uma tendência. Aliado aos critérios já citados, o foco em aplicações de *streaming* de vídeo foi o fator determinamente para a seleção dos trabalhos relacionados.

Em relação aos critérios adotados para a revisão de literatura, pode-se citar:

- Trabalhos dos últimos 5 anos
- Bases de artigos: IEEE, ACM, Scopus, Elsevier
- Busca de citações aos artigos seminais
- Número de citações dos artigos de interesse
- Keywords iniciais: "*network traffic classification AND streaming video AND (machine learning or fuzzy logic)*"

O princípio da abordagem para a revisão de literatura foram sobre trabalhos que abordavam classificação do tráfego de rede com métodos clássicos. A partir dos trabalhos encontrados, houve o direcionamento para buscas sobre aprendizagem de máquina em classificação de tráfego de rede. Devido ao foco no problema de pesquisa buscou-se trabalhos que abordassem modelos que tratam incerteza e imprecisão havendo destaque para lógica fuzzy valorada intervalarmente como uma tendência. Aliado aos critérios já citados, o foco em aplicações de *streaming* de vídeo foi o fator determinamente para a seleção dos trabalhos relacionados.

5.1 Classificação do Tráfego de Redes Explorando Aprendizagem de Máquina

As abordagens para classificação do tráfego de rede usualmente são baseadas em informações dos cabeçalhos da camada de transporte (portas de comunicação), comportamento dos *hosts* (número de conexões, endereços dos *hosts* pares), estatísticas (número de pacotes, quantidade de bytes, tempo entre os pacotes), análise da carga útil com técnicas de DPI (*Deep Packet Inspection*) (FINSTERBUSCH et al., 2013) e uso de técnicas de aprendizagem de máquina (NGUYEN; ARMITAGE, 2008).

Sendo assim, as técnicas de classificação do tráfego a serem aplicadas nestes cenários devem ser avaliadas quanto a sua acurácia e ao consumo de recursos computacionais que necessitam. Na literatura, é possível identificar trabalhos que abordam técnicas híbridas para classificação do tráfego, como discutido a seguir, sendo oportuno ressaltar que os mesmos apresentam resultados significativamente dispares quanto a acurácia atingida, o que vem constituindo uma motivação para continuidade das pesquisas na área.

Pelo estudo realizado nesta Tese, foi possível constatar que as propostas da literatura para classificação do tráfego possuem etapas que são adotadas de forma consensual. Neste sentido, na Figura 15, é apresentada uma taxionomia para classificação do tráfego de rede onde estão as etapas e as respectivas técnicas utilizadas em cada uma delas.

Neste sentido, os trabalhos (YAMANSAVASCILAR et al., 2017) e (SOLEYMANPOUR; SADR; BEHESHTI, 2020), entre outros realizam avaliações de técnicas de aprendizagem de máquina com um mesmo *Dataset* (GERARD DRAPPER GIL, 2022a). Por sua vez, em trabalhos como (DIAS et al., 2019) e (NIU et al., 2019) foram realizadas avaliações de técnicas para classificação do tráfego com aprendizagem de máquina em dados coletados em uma rede real e/ou com geração do tráfego simulado.

Na etapa de coleta INPUT é onde realiza-se a captura do tráfego de rede ou a submissão do tráfego já capturado. Nesta etapa são utilizadas técnicas consagradas, baseadas no uso de ferramentas que empregam a biblioteca Libpcap (Tcpdump,

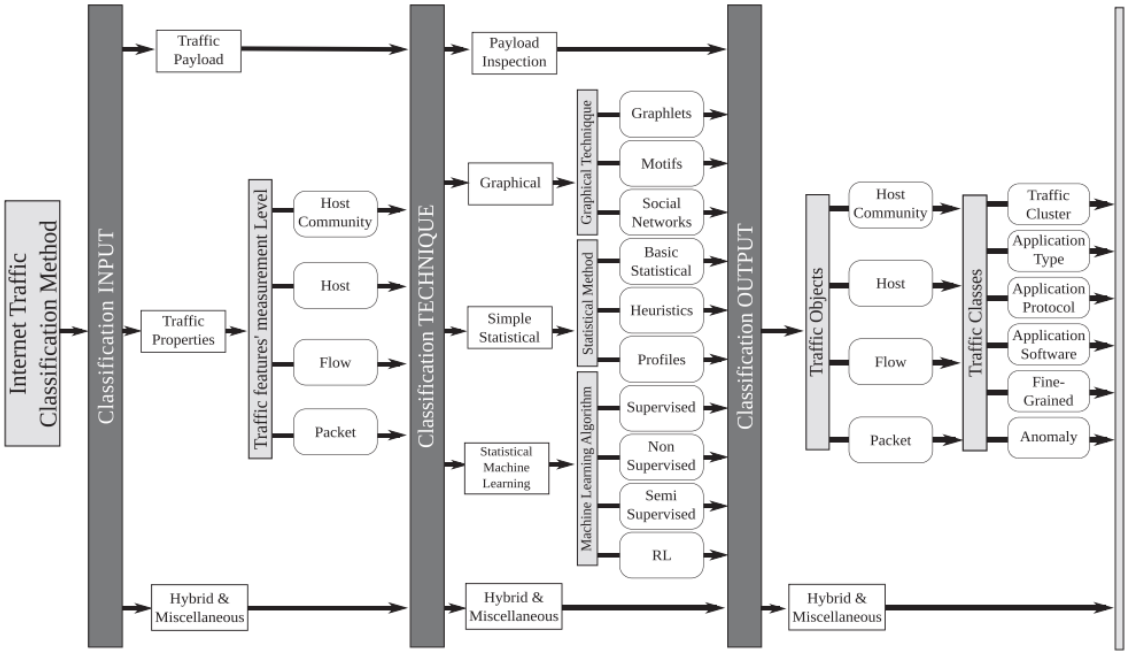


Figura 15 – Taxionomia para Classificação do Tráfego de Rede
Fonte: Velan et al. (2015)

Wireshark, Python Scapy) (LIBPCAP, 2022) para uso com pacotes e/ou *probes* do protocolo Netflow (CLAISE et al., 2004) para uso com fluxos de rede. O tráfego de rede a ser analisado pode ser fornecido gravado, sem a coleta em tempo de execução, constituindo uma alternativa baseada no o uso de *Datasets*. Oportuno registrar que o emprego de *Datasets* distintos nos trabalhos da literatura, é uma consequência sobretudo das restrições relativas a privacidade dos dados contidos nos pacotes.

Outro desafio diz respeito a dificuldade de manter *Datasets* atualizados, pois critério como confiança nas informações disponibilizadas implicam em esforço manuais significativos, rotulando e/ou pré-classificando as informações a serem disponibilizadas para os diferentes mecanismos de avaliação, baseados em lógicas de especificação e/ou aprendizado.

Na etapa de pré-processamento, que acontece após a etapa de INPUT, é onde realiza-se a formatação dos dados de acordo com os métodos que serão utilizados na próxima etapa de classificação. Com a disseminação do emprego de métodos de aprendizagem de máquina os pacotes ou fluxos de rede devem ser formatados de acordo com o algoritmo a ser aplicado.

Outra funcionalidade desta etapa é a extração das características do tráfego que possuem maior relevância para serem aplicadas aos algoritmos de aprendizagem de máquina. A quantidade de características afeta o tempo de treinamento, a acurácia do resultado e é diretamente proporcional ao consumo de recursos computacionais (DHOTE; AGRAWAL; DEEN, 2015). Esta etapa é fundamental para obtenção de melhores resultados na classificação do tráfego.

A etapa de classificação CLASSIFICATION, realiza a análise do tráfego a partir de técnicas únicas ou híbridas, com o uso de múltiplas abordagens para obtenção de melhores resultados. O objetivo desta etapa é realizar a classificação com a melhor acurácia, com menor tempo e com o menor uso de recursos computacionais. O uso de dados em lote, gravados, não demandam o fator tempo como algo crítico, entretanto, com o uso do tráfego coletado de forma online, tempo de execução, o fator tempo é fundamental.

Desta forma, esta etapa deve realizar um balanceamento entre os fatores para possibilitar o uso em um cenário real. Nesta etapa é onde se concentra a maior possibilidade de contribuições para a área de pesquisa foco desta Tese. Esta situação se reflete em um significativo número de publicações disponíveis na literatura da área, relacionadas a esta etapa de classificação.

Os resultados da classificação do tráfego, etapa de OUTPUT, definem a classificação do tráfego de entrada em alguma categoria/classe, de acordo com o nível de profundidade do classificador. Este tráfego classificado poderá prover insumos para aplicações de *firewall*, marcação de pacotes ou fluxos, em alertas para detectores de intrusão e para monitoramento por meio de alguma console de visualização de dados (*dashboards*).

O aproveitamento dos resultados da classificação do tráfego está relacionado de acordo com a forma de como os dados são coletados, capturados *online* ou gravados, bem como ao tempo para produção destes resultados. A restrição do tempo para gerar resultados é dependente do método usado para classificação e os recursos computacionais disponíveis para realizar o processo.

Considerando a disseminação do uso de criptografia em protocolos de aplicação, o uso de VPNs (*Virtual Private Network*), o aumento no volume de dados, a diversidade de protocolos e dispositivos e o uso de técnicas de ofuscação, os métodos clássicos foram afetados negativamente no seu desempenho e acurácia.

Por sua vez, é importante registrar, que em razão da diversidade da origem dos dados considerados nos diferentes trabalhos, uma análise comparativa direta entre os resultados destes estudos não se mostra oportuna.

Neste Capítulo são discutidos trabalhos relacionados a classificação do tráfego de rede criptografado. Os artigos selecionados abordam trabalhos com propostas de classificação do tráfego criptografado com ênfase em aprendizagem de máquina.

HEDGE: Efficient Traffic Classification of Encrypted and Compressed Packets

O artigo *HEDGE: Efficient Traffic Classification of Encrypted and Compressed Packets* (CASINO; CHOO; PATSAKIS, 2019) apresenta um novo método de classificação do tráfego denominado HEDGE (*High Entropy DistinGuishEr*) que tem como objetivo

classificar tráfego compactado e criptografado. O HEDGE é baseado na avaliação de fluxos de dados de forma aleatória e pode ser aplicado a pacotes individuais sem a necessidade de analisar todo o fluxo.

O trabalho mostra que a maioria dos mecanismos de privacidade e segurança se baseiam em algoritmos de criptografia para proteção das comunicações. Entretanto, implementações falhas utilizam comunicação em texto plano para transferência de informações restritas sem nenhuma proteção. Este é o caso de dispositivos com baixos recursos computacionais tais como dispositivos IoT (*Internet of Things*).

O uso de compressão ao invés de criptografia pode tornar a classificação do tráfego incorreta e afetar a identificação por meio de mecanismos de segurança. Devido a isto, é fundamental distinguir tráfego compactado do tráfego criptografado para garantir o gerenciamento, auditoria e detecção de comportamento suspeito. Segundo os autores, os métodos atuais são capazes de detectar tráfego em texto plano ou com baixa entropia, mas não são capazes de detectar com precisão dados com entropia mais complexa.

Diferente da maioria dos estudos existentes, os autores não analisam todo o tráfego de rede. Ao invés disto, são analisados apenas subconjuntos aleatórios para possibilitar a detecção em tempo real. Foram selecionados testes citados na literatura de acordo com a acurácia e eficiência e implementada uma abordagem que provê a distinção entre tráfego criptografado e não criptografado de alta entropia em fluxos de dados em tempo real.

Para validação da proposta foi construído um conjunto de dados utilizando *benchmarks* conhecidos, bem como selecionado um conjunto de estratégias de avaliação. O conjunto de dados possui um número balanceado de tipos de arquivos para evitar algum favorecimento. Segundo os autores, o método utilizado para classificação obteve 94,72% de acerto para pacotes randômicos de 64KB. O pior resultado de classificação foi no caso de pacotes de 1 KB, onde a acurácia ficou em 68,68%. Os resultados de outros trabalhos com o mesmo objetivo alcançaram 66,9%, entretanto, em um conjunto de dados diferente. O HEDGE obteve acurácia de 70.6% no conjunto de dados que foi avaliado em trabalhos relacionados.

A metodologia utilizada contemplou a geração de um conjunto de dados, empregando dados preexistentes gerados a partir do monitoramento do tráfego de arquivos de imagens, vídeos, binários, áudios, textos e PDF (*Portable Document Format*). Foram gerados conjuntos de fluxos de tamanho fixo, onde a carga útil dos pacotes ficou entre 1KB e 64KB com o uso de compressão e criptografia. Este conjunto foi formado com a metade dos dados criptografados e a outra metade dos dados compactados, sendo usado 10% dos dados para treinamento e 90% para testes e validação.

Para comparar o resultado da aplicação de algoritmos de testes de aleatoriedade, ond forma aplicados os algoritmos Chi Square Test e NIST SP 800-22, foram usadas

as somas dos falsos positivos e falsos negativos. Os autores relataram que arquivos binários compactados, de texto e vídeo puderam ser facilmente detectados na proposta apresentada e apontam que o uso de aprendizagem de máquina é uma alternativa promissora para a distinção entre tráfego com compressão e tráfego compactado.

Characterization of Encrypted and VPN Traffic using Time-related Features

O artigo *Characterization of Encrypted and VPN Traffic using Time-related Features* (DRAPER-GIL et al., 2016) aborda a efetividade para classificação do tráfego por meio das características de tempo dos fluxos de rede. O foco do trabalho foi em tráfego tunelado em VPN (*Virtual Private Network*) e tráfego com uso de criptografia. Foram utilizadas duas técnicas de aprendizagem de máquina (C4.5 e KNN) para testar a acurácia e validação da proposta.

Os resultados mostraram alta acurácia e desempenho confirmando os trabalhos anteriores da literatura sobre as características de tempo dos fluxos de rede como relevantes para a classificação do tráfego criptografado. Os autores definem duas categorias principais para classificação do tráfego: baseada em fluxo, usando propriedades tais como quantidade de bytes por segundo e duração do fluxo; baseada em pacotes, usando propriedades como tamanho e tempo de duração entre os pacotes. Segundo os autores, a caracterização do tráfego por VPN ainda é um desafio que não possui uma solução ótima.

A contribuição do artigo se deu em duas formas. Primeiro, a proposta de um método de classificação baseado em fluxo usando somente propriedades de tempo para caracterizar tráfego criptografado e tráfego tunelado por meio de VPN. Em segundo, a disponibilização de um *Dataset* com 14 tipos de protocolos identificados, sendo 7 com protocolos criptografados e 7 com protocolos tunelados em VPN.

A proposta teve como objetivo o uso de poucos recursos computacionais e a redução das características dos fluxos a serem analisados pelos métodos de aprendizagem de máquina. Segundo os autores, este foi o primeiro trabalho proposto de um método para caracterização do tráfego tunelado em VPN que realiza a categorização de 7 diferentes tipos de protocolos.

Para realizar os testes foram gerados conjunto de dados com capturas realizadas nos laboratórios dos autores com o uso de aplicações tais como Skype[1] e Facebook[2]. Foram capturadas sessões do tráfego com a aplicação nativa e com o uso de VPN. Desta forma, foram criadas 14 categorias, sendo 7 com a aplicação nativa e 7 com a aplicação tunelada em VPN, por exemplo P2P e VPN-P2P. O tráfego foi capturado com

[1] `https://www.skype.com/`
[2] `https://facebook.com`

as ferramentas Wireshark[3] e Tcpdump[4] gerando um total de 28GB de dados. Para o tráfego com VPN foi utilizada a ferramenta OpenVPN[5].

Para extrair as características dos fluxos de rede foi desenvolvida a ferramenta ISCXFlowMeter (GERARD DRAPPER GIL, 2022b), renomeada posteriormente para CicFlowMeter. Segundo os autores, esta ferramenta possibilita maior flexibilidade na escolha das propriedades dos fluxos usadas nos cálculos e o melhor controle de tempo de duração dos fluxos. Os tempos de duração dos fluxos UDP e TCP neste trabalho foram de 15, 30, 60 e 120 segundos.

Para realizar os experimentos foi usada a ferramenta WEKA (WEKA, 2022), com os parâmetros padrão, com os algoritmos de clusterização (KNN) e classificação (árvore de decisão C4.5). De forma geral, os algoritmos C4.5 e KNN obtiveram resultados similares, embora o C4.5 tenha apresentado desempenho um pouco melhor.

Foram criados dois cenários, A e B, para os testes. No cenário A, para classificação do tráfego tunelado em VPN ou não, os melhores resultados foram obtidos com o algoritmo C4.5 e 15 segundos de tempo limite dos fluxos. No cenário B, houve variação do tempo de fluxo em relação a precisão, onde o valor de 120 segundos apresentou melhor desempenho em ambos algoritmos testados.

Os resultados do artigo demonstraram que o conjunto de características relacionadas ao tempo são bons parâmetros para classificação atingindo níveis acima 80% de acurácia. Os algoritmos C4.5 e KNN obtiveram desempenho similar em todos os experimentos, embora o C4.5 tenha atingido melhores resultados. Outra contribuição do artigo foi no tempo menor de duração dos fluxos, menor que os 600 segundos usados como padrão na literatura, o que possibilita melhor desempenho para classificação do tráfego de rede.

Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning

No artigo *Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning* (LOTFOLLAHI et al., 2017) é proposto um sistema baseado em *deep learning* que integra a extração de características e a classificação do tráfego. O sistema denominado de *Deep Packet* realiza a caracterização do tráfego em categorias, por exemplo P2P ou VoIP, e faz a identificação de aplicações, tais como Skype e Bittorrent. Segundo os autores, diferente dos métodos existentes divulgados na literatura o sistema *Deep Packet* pode identificar tráfego criptografado e também pode distinguir tráfego tunelado em VPN e tráfego não tunelado em VPN.

O sistema utilizou o algoritmo CNN (*Convolutional Neural Network*) como modelo

[3] https://www.wireshark.org
[4] https://www.tcpdump.org/
[5] https://openvpn.net/

de classificação e obteve como resultado 0,98 de *recall* na tarefa de identificação de aplicações e 0,94 na tarefa de categorização do tráfego. De acordo com os autores, o método apresentado superou todas as propostas anteriores publicadas que fizeram uso do *Dataset* UNB ISCX VPN-nonVPN (GERARD DRAPPER GIL, 2022a) para a realização de categorização do tráfego e identificação de aplicações em rede.

Na proposta do sistema *Deep Packet* não há necessidade da intervenção de um especialista em rede para extrair características relacionadas ao tráfego de rede. Este procedimento é realizado de forma automatizada. Segundo os autores, o *Deep Packet* pode classificar tráfego P2P com grande acurácia, o que é uma tarefa complexa devido as várias técnicas de ofuscação usadas por este tipo de aplicação.

Originalmente, o *Dataset* ISCX VPN-nonVPN foi capturado com informações da camada de enlace de dados, incluindo o cabeçalho Ethernet nos pacotes. Entretanto, as informações deste cabeçalho não colaboram para a classificação do tráfego de rede. Sendo assim, os autores realizaram uma fase de pré-processamento dos pacotes para remoção das informações do cabeçalho Ethernet. Na camada de transporte os cabeçalhos dos pacotes TCP, 20 Bytes de comprimento, UDP, com 8 Bytes de comprimento, possuem tamanhos distintos. Para tornar os pacotes uniformes foram inseridos zeros ao final do cabeçalho UDP para tornar o tamanho padrão em 20 Bytes.

A partir disto, os pacotes foram convertidos de bits para Bytes para redução na entrada de dados das redes neurais usadas na proposta. Com o mesmo propósito, para uniformizar o tamanho dos pacotes foram inseridos zeros nos pacotes com menos de 1480 Bytes. Para obter melhor desempenho todos os Bytes de cada pacote foram divididos por 255, valor de combinações máximo em um Byte, para que todos os valores de entrada ficassem entre 0 e 1.

O *Dataset* não foi balanceado e foi realizado o balanceamento entre as classes aleatoriamente removendo amostras das classes com maior número de pacotes. Foi utilizada a biblioteca Keras (KERAS, 2022) e a ferramenta Tensorflow (TENSORFLOW, 2022) para implementação do sistema de classificação.

O *Deep Packet* considera o tráfego de rede no nível de pacote e, segundo os autores, pode classificar cada pacote de um fluxo de rede, com o uso do maior número de informações que podem ser aproveitadas de um fluxo. Ao contrário de métodos que fazem uso de DPI, o *Deep Packet* não busca por assinaturas nos pacotes sendo apenas usadas as características aprendidas para cada tipo de aplicação. Desta forma, não há necessidade de análise dos dados de carga útil dos pacotes o que torna a classificação de pacotes criptografados possível.

Os resultados deste trabalho mostraram que a solução proposta superou todos os trabalhos similares que fizeram uso do *Dataset* ISCX VPN-nonVPN, para a classificação do tráfego. Segundo os autores, o *Deep Packet* pode ser modificado para tratar de tarefas mais complexas tais como a classificação multi-canal para distinguir

diferentes ações das aplicações e melhorar a acurácia de classificação do tráfego na rede Tor[6]. Uma contribuição importante foi a extração automática de características do tráfego sem necessidade de auxílio de especialistas em rede que, futuramente, poderá facilitar a aplicação desta estratégia em grandes volumes de dados e melhorar os resultados da classificação do tráfego de rede em geral.

TLS/SSL Encrypted Traffic Classification with Autoencoder and Convolutional Neural Network

No artigo *TLS/SSL Encrypted Traffic Classification with Autoencoder and Convolutional Neural Network* (YANG et al., 2018), os autores propuseram duas abordagens de *deep learning* para aprendizagem de características do tráfego e compararam com o estado da arte. Um dos modelos foi com o uso baseado em Autoencoder com o objetivo de extrair características representativas dos pacotes. O outro modelo usado foi o CNN (*Convolutional Neural Network*). Este modelo possui aprendizagem por meio de múltiplas dimensões, provê melhorias na acurácia da classificação e é largamente utilizado em diversas áreas de pesquisas.

Nos resultados obtidos no trabalho, o CNN mostrou-se superior em relação aos outros algoritmos analisados. O objetivo deste trabalho foi a necessidade de utilizar algoritmos de classificação para discriminar tráfego malicioso do tráfego normal de rede.

Nos últimos anos, a demanda por privacidade dos usuários e a necessidade de melhorias na segurança com a aplicação de criptografia de dados causou o aumento do tráfego criptografado. As técnicas de criptografia que usam o SSL (*Secure Sockets Layer*) e TLS (*Transport Layer Security*) são populares para a proteção da privacidade dos usuários finais. Este trabalho utilizou modelos de *deep learning* para classificar tráfego criptografado por meio de SSL e TLS.

No trabalho foram utilizados os tamanhos dos pacotes e os tempos entre o recebimento e envio de pacotes como características para classificação do tráfego. O tamanho dos pacotes causou maior assertividade do que o tempo de envio e recebimento de pacotes no contexto de classificação do tráfego criptografado.

Foi utilizado o Autoencoder, que é uma forma de rede neural com o uso de algoritmo não-supervisionado com o objetivo de extrair as características relevantes do tráfego a ser usado como entrada para a classificação. Este método pode ser utilizado com compressão que possibilita a redução do número de parâmetros a serem analisados. Esta redução facilita a classificação, visualização, comunicação e armazenamento de dados que possuem um grande número de dimensões, tal como o tráfego de rede.

[6] `https://www.torproject.org/`

As características do tráfego foram extraídas dos metadados dos fluxos, tamanhos dos pacotes, tempos de envio e recebimento dos pacotes e informações dos cabeçalhos não criptografados do protocolo TLS. A extração resultou em 811 dimensões. Foram analisados apenas os primeiros 100 pacotes de cada fluxo para servirem de entrada aos algoritmos de extração de características e classificação do tráfego.

Para realizar a coleta do tráfego foi utilizada a ferramenta Chromedriver (CHROMEDRIVER, 2022) para acessar a API (*Application Programming Interface)* do navegador Chrome no acesso dos top 50 sites listados no serviço Alexa (ALEXA, 2022). A ferramenta Tshark[7] foi utilizada para capturar os fluxos de rede. A ferramenta Joy (JOY, 2022) foi usada para gerar em formato JSON a saída dos arquivos de captura da ferramenta Tshark.

A solução proposta, Autoencoder e CNN, comparou o CCN com outros 5 algortimos (*Naive Bayes, Logistic Regression, K-Nearest Neighbors, Decision Tree e Random Forest*) obtendo resultados superiores aos relatados nos trabalhos relacionados. Com o uso do Autoencoder não foi obtido um resultado superior em todos os testes, havendo alguns resultados piores com dependência do tipo de conjunto de características a serem extraídas.

End-to-end Encrypted Traffic Classification with One-dimensional Convolution Neural Networks

No artigo *End-to-end Encrypted Traffic Classification with One-dimensional Convolution Neural Networks* (WANG et al., 2017), os autores propõem um método de classificação do tráfego criptografado fim a fim utilizando uma rede neural CCN (*Convolutional Neural Network*) com uma dimensão (1D-CNN).

Este método integrado engloba a extração e a seleção de características do tráfego e um classificador, agrupados em um único *framework* com o objetivo do aprendizado automatizado entre os dados não-tratados de entrada e a saída. Os autores alegam ser este o primeiro trabalho que aplicou um método fim a fim para classificação do tráfego criptografado. O método foi validado com o uso do *Dataset* ISCX VPN-nonVPN.

No trabalho, é proposta uma divisão em categorias relacionadas a granularidade de requisitos para classificação do tráfego criptografado. A classificação do tráfego criptografado é categorizado em classificação do tráfego criptografado, caracterização do tráfego criptografado e classificação detalhada do tráfego criptografado. Os autores focaram na caracterização do tráfego criptografado.

O *framework* proposto não contém módulos independentes tais como extração de características, seleção de características e classificador. Estes módulos foram integrados a um modelo baseado em CNN. As características do tráfego são auto-

[7] https://www.wireshark.org/docs/man-pages/tshark.html

maticamente aprendidas e o tráfego é diretamente classificado utilizando *softmax*. A fase de pré-processamento é responsável por transformar os dados brutos de captura do *Dataset* em entrada para o modelo CNN. O pré-processamento foi realizado por ferramenta desenvolvida pelos autores.

No artigo, os autores consideraram o tráfego de rede como dados sequenciais, sendo um fluxo de Bytes unidimensional organizado em uma estrutura hierárquica. Esta estrutura de Bytes, pacotes e sessões (fluxos) e todo o tráfego é muito similar a letra, palavra, sentença e artigo usado no domínio de processamento natural de linguagem (NLP - *Natural Language Processing*). Neste domínio, aplicações com sucesso tais como a análise de sentimento e a classificação do tráfego, fizeram uso do modelo 1D-CNN (*One Dimension CNN*). Os autores buscaram inspiração no domínio de NLP para realizar a classificação do tráfego de rede criptografado com 1D-CNN e comparar o desempenho com o modelo 2D-CNN (*Two Dimension CNN*).

A primeira comparação foi entre o uso de pacotes ou sessões (fluxos) para representação dos dados. Os resultados indicaram que as utilizações dos dados no formato de sessões obtiveram maior acurácia em quase todos os testes. Nos testes onde o uso de sessões não foi melhor, houve uma diferença de apenas 0,2% na acurácia a favor do uso de representação por pacotes. Os autores assumiram que o melhor desempenho com o uso de sessões foi devido a haver informações em ambos os sentidos do tráfego e maior quantidade de características que podem ser usadas na classificação do tráfego a partir da interação entre os fluxos.

A segunda comparação a respeito do formato dos dados se deu pelo uso de informações de todas as camadas de protocolos (cabeçalhos das camadas de enlace de dados, rede, transporte e aplicação) ou somente da camada de aplicação. O uso de todas as camadas obteve, em média, 4,85% de acurácia acima da utilização somente da camada de aplicação. Desta forma, os autores concluíram que a representação dos dados ideal seria em fluxos e analisando informações de todas as camadas de protocolos.

Em relação ao modelo para classificação do tráfego criptografado, o uso de CCN com apenas uma dimensão mostrou-se com melhor desempenho do que com o uso de duas dimensões. Os quatro experimentos realizados neste trabalho obtiveram desempenho superior ao estado da arte. Os autores concluíram que o uso de um modelo fim a fim com apenas uma dimensão para classificação do tráfego criptografado pode ser mais efetivo do que um modelo com duas dimensões.

5.2 Discussão sobre Trabalhos Selecionados Explorando Aprendizagem de Máquina

A análise comparativa realizada entre os trabalhos selecionados teve como critério os métodos de classificação utilizados, os tipos do tráfego analisados, a granularidade do tráfego, a origem dos dados usados para validação e as ferramentas utilizadas. Esta comparação está sumarizada na Tabela 3.

A revisão de literatura realizada durante este trabalho, permitiu constatar a tendência no emprego de métodos estatísticos baseados em lógica clássica para abordagem de algoritmos de aprendizagem de máquina, considerando neste caso a classificação do tráfego criptografado. Dentre as técnicas encontradas destacou-se o uso de redes neurais tal como a CNN (*Convolutional Neural Network*). Esta situação também está contemplada na Tabela 3.

Oportuno destacar que os trabalhos (WANG et al., 2017), (LOTFOLLAHI et al., 2017) e (DRAPER-GIL et al., 2016) utilizaram tráfego criptografado e tunelado em VPN. Com o emprego de tunelamento torna-se mais complexa a classificação do tráfego devido a mudanças nas características dos protocolos originais.

Outrossim, a análise de fluxos de pacotes foi utilizada em todos os trabalhos. Mesmo em (YANG et al., 2018), no qual foi feito o uso de características dos pacotes, também foi empregado fluxos de pacotes para servir como entrada aos algoritmos de classificação.

Deste modo, foi possível identificar que a tendência pelo emprego do fluxo de pacotes é decorrência direta da maior quantidade de informações que disponibilizam, o que contribui para uma maior acurácia dos algoritmos de classificação. Entretanto, como contrapartida, pelo grande número de características que os fluxos podem disponibilizar, surge um novo desafio de pesquisa associado a identificação e extração de características relevantes para classificação do tráfego de rede, por meio de técnicas de pré-processamento automatizadas.

Neste sentido, este pré-processamento, operando de forma automatizada, foi discutido em (YANG et al., 2018), situação esta diferente dos outros trabalhos nos quais houve a escolha manual das características relevantes dos fluxos de rede a serem classificados.

Foi possível também constatar, que os *Datasets* utilizados nos trabalhos não obedeceram um padrão. Este fato é um dos desafios nesta área de pesquisa para comparação dos resultados obtidos pelas diferentes pesquisas, e a decorrente potencial melhoria nas estratégias para classificação do tráfego.

Esta situação do emprego de *Datasets* heterogêneos, foi decorrência destes terem sido gerados pela captura do tráfego em ambientes específicos de cada grupo de pesquisa e/ou, terem sido construídos a partir de segmentos de *Datasets* já empregados

Tabela 3 – Comparação entre os Artigos que Abordam Classificação do Tráfego Criptografado

Trabalho	**Métodos de Classificação**	**Tipos de Tráfegos Analisados**	**Granularidade do Tráfego**	***Dataset***	**Ferramentas**
(CASINO; CHOO; PATSAKIS, 2019)	Informações não criptografadas dos protocolos de criptografia	Arquivos compactados e criptografados (documentos, imagens e vídeo)	Fluxo	Criados a partir de *benchmarks* existentes	Testes de Aleatoridade (Chi Square Test e NIST SP 800-22)
(DRAPER-GIL et al., 2016)	C.45 e KNN	Tráfego tunelado e criptografado	Fluxo	Criado *Dataset* próprio	Wireshark, Tcpdump, ISCX-FlowMeter, WEKA
(LOTFOLLAHI et al., 2017)	CNN	Tráfego tunelado e criptografado (rede Tor)	Fluxo	UNB ISCX VPN-nonVPN	Biblioteca Keras, Tensorflow
(YANG et al., 2018)	Autoencoder, CNN	SSL, TLS	Fluxo, Pacotes	Baseado no Top 50 sites (Alexa)	Chromedriver, Tshark. Joy
(WANG et al., 2017)	CNN	Tráfego tunelado e criptografado	Fluxos	UNB ISCX VPN-nonVPN	TensorFlow

pelo grupo.

Entretanto, foi possível observar nos trabalhos mais recentes a tendência de buscar um *Dataset* comum as diferentes pesquisas. Destacou-se neste sentido a citação do *Dataset* ISCX VPN-nonVPN (GERARD DRAPPER GIL, 2022a). Este *Dataset* está disponível publicamente e apresenta capturas recentes do tráfego criptografado e/ou tunelado.

No que diz respeito ao ferramental computacional empregado na área de classificação do tráfego, foi constato pela literatura que as ferramentas WEKA e TensorFlow, bem como o uso de bibliotecas em Python para aprendizagem de máquina estão presentes em grande parte dos trabalhos. Na fase de captura do tráfego, o uso da biblioteca Libpcap com a criação de ferramentas proprietárias e ferramentas conhecidas tais como Wireshark, Tshark e Tcpdump são bastante usadas.

Nos trabalhos analisados, foi constatado que parte significativa dos esforços de prototipação e testes ficaram associados a criação de *scripts* utilizados para ajustes nos *Datasets* antes do seu processamento.

5.3 Classificação do Tráfego Explorando Lógica Fuzzy

A revisão de literatura realizada identificou onze trabalhos relacionados que utilizam lógica fuzzy na classificação do tráfego de rede, e relatados nesta seção, suas principais características estão resumidas na Tabela 4.

Tabela 4 – Trabalhos Relacionados

Artigo	TTR	ALFG	TCR	ELF
(LIANG; MENDEL, 2001)	MPEG (VBR) vídeo	Classificadores Fuzzy (Tipo-1 e Tipo-2) Singleton e Nonsingleton Interval		T1FL/T2FL
(JAMMEH et al., 2009)	Streaming de Vídeo	Interval Tipo-2 FLC		T2FL
(RIZZI et al., 2015)	Fluxos TR	Min–Max neuro-fuzzy networks treinados em algoritmo PARC	✓	Neuro-Fuzzy
(ASMUSS; LAUKS, 2015)	Classificação do tráfego e Detecção de Anomalias	fuzzy C-means, Fukuyama and Sugeno index, Xie e Beni index, separation e compactness index		Clustering
(SHALAGINOV; FRANKE, 2015)	TR anômalo e malicioso	Neuro-Fuzzy (NF), Self-Organizing Maps (SOM), Mean Absolute Error (MAE), Relative Absolute Error (RAE) e Mean Absolute Percent Error (MAPE)		Neuro-Fuzzy
(QADER; ADDA; AL-KASASSBEH, 2017)	Classificação de Falhas de Rede	K-Means, Fuzzy C Means e Expectation Maximization		Clustering
(DUCANGE et al., 2017)	Fluxos TR	Multi-objective evolutionary fuzzy classifiers (MOEFCs)	✓	MOEFCs
(ABDULLAH; AL-HASHMI, 2018)	Inspeção de Anomalias em TR	Time Series Evolving Fuzzy Engine (TiSEFE)		T1FL
(AL-OBEIDAT; EL-ALFY, 2019)	TR anômalo e malicioso	Abordagem híbrida combinando decision tree learning e fuzzy multicriteria classification method	✓	Hybrid
(IGLESIAS; MILOSEVIC; ZSEBY, 2019)	Classificação de TR anômalo e identificação de de ataques por tipo	Multiclass Fuzzy Classification, neuro-fuzzy classification	✓	Fuzzy Decision Trees
(PARFENOV et al., 2020)	Identificação de ataques em TR	Funções de fuzzificação triangulares de pertinência e criação de bases de regras a partir de abordagem com árvores de decisão	✓	Neuro fuzzy/T1FL
FNC	Streaming de Vídeo	Lógica Fuzzy Valorada Intervalarmente	✓	T2FL

Tipos do tráfego de Rede (TTR) Abordagem de Lógica Fuzzy Geral (ALFG) Tráfego Criptografado (TCR) Extensões Lógica Fuzzy (ELF) FuzzyNetClass (FNC) Tráfego de Rede (TR) T1FL (Lógica Fuzzy Tipo-1) T2FL (Lógica Fuzzy Tipo-2)

MPEG VBR Video Traffic Modeling and Classification using Fuzzy Technique

O artigo (LIANG; MENDEL, 2001) propôs o uso de classificadores fuzzy tipo 2 para classificar vídeos com compressão. Devido a ser um trabalho do ano de 2001, não foram analisados os vídeos em formato de fluxos em rede, apenas as características do resultado do vídeo codificado.

Foram analisados 5 tipos de classificadores fuzzy Tipo 1 e Tipo 2 e comparados com uma abordagem usando *Bayesian* para classificação de vídeos. A abordagem para classificação foi baseada nos quadros (*frames*) de vídeos codificado em MPEG com taxa variável de compressão (VBR - (*Variable Bit Rate*). As contribuições deste trabalho mostraram que um classificador fuzzy tipo 2 no qual a entrada é modelada como um classificador de conjunto fuzzy tipo 2 e com o uso de funções de pertinência antecedentes obtiveram os melhores resultados.

Interval Type-2 Fuzzy Logic Congestion Control for Video Streaming across IP Networks

O artigo (JAMMEH et al., 2009) propõe um controlador de congestionamento baseado em lógica fuzzy tipo 2 (FLC - *Fuzzy Logic Congestion Controller*) obtendo um resultado com qualidade de vídeo superior em comparação com os controladores tradicionais existentes e um controlador de congestionamento com abordagem de lógica fuzzy tipo 1.

Um controlador de congestionamento baseado em lógica fuzzy tipo 1 não provê acurácia devido a não lidar com as incertezas associadas a ambientes de rede dinâmicos. Já um FLC em lógica fuzzy tipo 2 pode lidar com as incertezas nas variações dos recursos de rede e gerar melhores resultados. Segundo os autores, este seria o primeiro artigo com o uso de controle de congestionamento baseado em lógica fuzzy tipo 2 em *streaming* de vídeo em redes IP.

A Low Complexity Real-time Internet Traffic Flows Neuro-Fuzzy Classifier

O artigo (RIZZI et al., 2015) propõe o uso de um sistema neuro-fuzzy, mais especificamente uma rede Min-Max treinada pelo algoritmo PARC (RIZZI; PANELLA; MASCIOLI, 2002) para classificação de fluxos do tráfego de rede em tempo real com recursos simples extraídos dos primeiros pacotes dos fluxos. A técnica de classificação proposta pode ser aplicada sempre que for possível extrair fluxos de rede individuais, mesmo que as cargas úteis do pacote sejam criptografadas.

Os autores mostram que as redes Min–Max alcançam alta acurácia, com resultados comparáveis aos algoritmos de melhor desempenho em WEKA (SVM, Random Tree, Random Forest). O nível de complexidade necessária para o modelo de classificação necessária é muito menor com redes Min-Max em relação aos outros modelos, permitindo a implementação de algoritmos de classificação eficazes com baixo desempenho e em plataformas computacionais com recursos reduzidos. Foram aplicados 4 *Datasets* gerados pelos próprios autores e além de implentações em software do algoritmo, foi utilizada a implementação em hardware com o uso de FPGA (*Field Programmable Gate Array*).

Network Traffic Classification for Anomaly Detection Fuzzy Clustering based Approach

O artigo (ASMUSS; LAUKS, 2015) apresenta o desenvolvimento de método para a classificação e detecção de anomalias do tráfego de rede com base em séries temporais usando a técnica de *clustering*. Os autores tiveram como base o desvio anômalo do perfil do tráfego normal, definido empiricamente com base em informações coleta-

das sobre as propriedades do tráfego em condições normais.

Segundo os autores, a abordagem tradicional de *clustering* não pode ser efetivamente usado em casos de classificação do tráfego devido ao comportamento dinâmico e as alterações ao longo do tempo do fluxo. Estas alterações do tráfego de rede resultam em mudanças nos agrupamentos durante a duração do fluxo de rede. Este cenário de mudanças de estado e incertezas pode ser modelado usando a abordagem fuzzy.

Para caracterizar o tráfego normal, foram analisados os algoritmos de agrupamento, incluindo fuzzy C-means e algoritmos C-means possibilísticos, métodos de agrupamento possibilístico e técnica de agrupamento difuso possibilístico não supervisionado. Os autores utilizaram apenas o volume do tráfego, considerando a carga útil do pacote e não foram consideradas situações de congestionamento da rede, por exemplo a quantidade de retransmissões.

Automated Generation of Fuzzy Rules from Large-scale Network Traffic Analysis in Digital Forensics Investigations

O artigo (SHALAGINOV; FRANKE, 2015) descreve a aplicação de NF (*Neuro-Fuzzy*) para classificação do tráfego de rede com a redução do número de regras. Em particular, a concentração do trabalho foi na detecção de padrões em atividades maliciosas a partir de *Datasets* do tráfego de rede. Foram propostas melhorias para o algoritmo NF que resulta no manuseio adequado de conjuntos de dados em grande escala, reduz o número de regras e produz uma complexidade diminuída do modelo de classificação.

Para avaliação do trabalho foi utilizado o *Dataset* de detecção de intrusão da KDD Cup 1999 (TAVALLAEE et al., 2009). Para realizar a classificação foram usados 41 atributos dos pacotes, sendo os atributos definidos como pacotes de rede “ruins” ou “bons”. Os resultados foram validados na ferramenta WEKA. A precisão alcançada com a redução das regras foi de 98,787% ao usar 39 regras contra 98,790% com 10.231 regras.

Comparative Analysis of Clustering Techniques in Network Traffic Faults Classification

O artigo (QADER; ADDA; AL-KASASSBEH, 2017) são comparados três algoritmos de mineração de dados diferentes como parte da solução proposta para classificação de falhas de rede: K-Means, FCM (Fuzzy C Means) e Expectation Maximization. A abordagem proposta pretende ajudar a capturar comportamentos anormais em redes de dados propiciando uma alternativa para a classificação e gerenciamento de falhas

de rede em tempo real.

Os três algoritmos de *clustering* (agrupamento) foram executados na ferramenta WEKA em conjuntos de dados gerados pelos próprios autores. Os agrupamentos foram classificados em tráfego normal, tráfego de falha de link, falha de servidor, tempestade de transmissão (*broadcast storm*) e estação tagarela (*babbling* - quando a placa de rede transmite pacotes de forma ininterrupta, sem respeitar os protocolos de compartilhamento de meio físico).

Os resultados revelaram que existem diferenças sutis de desempenho entre os três algoritmos. Os algoritmos K-Means e EM são relativamente mais rápidos que FCM, enquanto avaliação com precisão e *recall* em FCM é mais preciso, embora exija mais tempo para processar os dados.

A Novel Approach for Internet Traffic Classification based on Multi-objective Evolutionary Fuzzy Classifiers

O artigo (DUCANGE et al., 2017) propõe uma abordagem para resolver o problema de classificação do tráfego usando classificadores fuzzy evolutivos multiobjetivos (MOEFCs - *Multi-objective Evolutionary Fuzzy Classifiers*). Os classificadores MOEFCs lidam com a criação de base de regras fuzzy por meio de algoritmos evolutivos multiobjetivos: durante o processo de design evolutivo, tanto a precisão quanto o nível de interpretabilidade dos modelos são simultaneamente otimizados. Para avaliação da proposta os autores utilizaram dois *Datasets* com tráfego de rede extraído de duas redes do mundo real.

Neste artigo, um conjunto composto de 21 atributos estatísticos dos fluxos de rede foram extraídos para caracterizar os variados tipos de protocolos analisados. Nos resultados apresentados, foram apresentados conjunto de regras 4 regras para classificar o protocolo HTTP e apenas 1 regra para classificar o protocolo SSH, por exemplo. Para validar os resultados da classificação proposta no artigo os autores fizeram uso de técnicas de DPI (*Deep Packet Inspection*).

TiSEFE: Time Series Evolving Fuzzy Engine for Network Traffic Classification

O artigo (ABDULLAH; AL-HASHMI, 2018) propõe um novo sistema fuzzy evolutivo para classificar anomalias inspecionando o tráfego de rede. O objetivo deste trabalho é usar séries temporais e a lógica fuzzy para detectar anomalias em tempo de execução (*online*). O sistema foi denominado como TiSEFE (*Time Series Evolving Fuzzy Engine for Network Traffic Classification*).

O tráfego de rede foi obtido por meio do uso do protocolo NetFlow, usando um *probe* em um roteador. Segundo os autores, a principal vantagem do TiSEFE é a

capacidade de lidar com a natureza da incerteza de tráfego de rede. Isso é feito através da evolução da base de conhecimento usando a lógica fuzzy. Foram usados conjuntos fuzzy para enquadrar os fluxos exportados por meio do Netflow, por exemplo "Normal" se μ < 0,3, "Suspeito" se μ entre 0,3 e 0,6, e "Ataque" se μ > 0,6.

A ideia desta pesquisa é inspirada em dois atributos do tráfego de rede atual: o *big data* e a natureza de incerteza. Outra vantagem do TiSEFE é a capacidade de usado tanto para classificação binária, por exemplo tráfego normal ou ataque, bem como para casos de múltiplas classes, por exemplo tráfego normal, suspeito ou ataque. Segundo os resultados apresentados no artigo o TiSEFE apresentou 81% de precisão na classificação binária e 80% de precisão em classificação de múltiplas classes.

Hybrid Multicriteria Fuzzy Classification of Network Traffic Patterns, Anomalies, and Protocols

No artigo (AL-OBEIDAT; EL-ALFY, 2019) é proposta uma nova abordagem híbrida de aprendizagem de máquina para classificação do tráfego de rede baseada em árvores de decisão fuzzy multicritério com seleção de atributos. Neste trabalho foram avaliados *Datasets* do tráfego de rede com protocolos e ataques, com e sem uso de criptografia. O método híbrido proposta foi denominado PROAFTN (*PROcdure dAffectation Floue pour la problmatique du Tri Nominal*).

Segundo os autores, o motivo para o uso de árvores de decisão em conjunto com a lógica fuzzy é devido a geração de resultados diretos e com o uso de regras estritas classificar um tipo de classe do tráfego de rede. Sendo assim, não há área para que se tenham valores entre sim e não. A proposta do PROAFTN também tem características interessantes, incluindo a geração de regras compreensíveis e o uso de associação para lidar com a incerteza na atribuição de um objeto a uma classe.

O trabalho apresentou a aplicação da proposta em vários *Datasets*, 13 conjuntos de dados, com diversas características e condições do tráfego de rede. Além disto, o desempenho foi comparado com outras três abordagens de aprendizagem de máquina: Naive Bayes, SVM e KNN. Nos resultados apresentados o PROAFTN obteve desempenho superior ao SVM e KNN e comparável ao Naive bayes.

Fuzzy Classification Boundaries Against Adversarial Network Attacks

O artigo (IGLESIAS; MILOSEVIC; ZSEBY, 2019) trata os limites de abordagens em aprendizagem de máquina para classificação e de detecção de ataques em rede os quais exploram as fraquezas deste tipo de abordagem. São apresentados os conceitos por meio de uma configuração experimental com tráfego de rede em que as árvores de decisão linear são envolvidas por um algoritmo de pontuação de associa-

ção por classes de ataques.

Os resultados mostram que ataques evasivos, ou seja, falsos negativos, tendem a ser classificados com baixos níveis de associação para as classes de ataques, o que significa que eles estão localizados em zonas próximas aos limites de classificação.

No desenvolvimento deste trabalho foi desenvolvido um algoritmo denominado OTDF (*Decision Tree Fuzzifier*), que envolve uma árvore de decisão linear ou uma árvore discriminante linear (LDT - *Linear Discriminant Tree*) e calcula pontuações de associação de uma classe com base na distância das amostras aos limites de decisão.

Foram utilizados os *Datasets* NSL-KDD com 41 atributos e o UNSW-NB15 com 42 atributos. Para seleção dos atributos mais relevantes foi a aplicado o algoritmo PCA (*Principal Component Analysis*) e ignorados no processo de seleção os atributos não numéricos. Como resultados apresentados foi realizada uma análise comparativa entre as abordagens LDT, SDT e ODTF, onde a proposta do trabalho obteve resultados de acurácia comparáveis aos métodos tradicionais e com o consumo moderado de recursos computacionais.

Research of Multiclass Fuzzy Classification of Traffic for Attacks Identification in the Networks

Na perspectiva de (PARFENOV et al., 2020), foi desenvolvido um método para identificar *Exploits, Fuzzers* e ataques genéricos com base em lógica fuzzy para classificação multiclasse. Na parte experimental do estudo, os autores realizaram uma análise comparativa com os métodos Naïve Bayes, SVM e KNN, obtendo maior desempenho e menor consumo de recursos para com o uso de grandes fluxos de dados.

Neste trabalho foi utilizado o *Dataset* UNSW-NB15 (MOUSTAFA; SLAY, 2016), que contém dados sobre o tráfego normal e data 9 tipos de ataques reais: *Worms, DoS, Analysis, Fuzzers, Shellcode, Generic, Reconnaissance, Exploits e Backdoor*. Para a montagem das regras fuzzy foram extraídas usando árvore de decisão, com o algoritmo C4.5, em uma base de treinamento originada do *Dataset* UNSW-NB15. Desta etapa foram obtidas 550 regras a partir das dos atributos do tráfego de rede e os tipos de classes desejadas.

Nos resultados do estudo, os autores mostraram que o sistema de inferência fuzzy permite identificar o tipo de ataque de rede com uma precisão de 64,58% e 74,94% para o conjunto de treinamento. Na análise comparativa com os métodos de aprendizagem de máquina, a proposta deste trabalho apresentou melhor precisão em 3 classes e resultados similares para as demais classes.

5.4 Discussão sobre Trabalhos Selecionados Explorando Lógica Fuzzy

A proposta do emprego de classificadores com uso da lógica fuzzy valorada intervalarmente (T2FL) para classificar o tráfego de vídeo usando dados compactados foi abordada em (LIANG; MENDEL, 2001). Um sistema T2FL foi concebido visando alcançar a entrega de melhor qualidade de vídeos. Os resultados foram obtidos com relação a controladores tradicionais, e também a um controlador de tráfego de rede fuzzy (JAMMEH et al., 2009).

No trabalho de (RIZZI et al., 2015) foi utilizado abordagem de redes neuro-fuzzy, treinadas pelo algoritmo PARC e comparadas com um sistema de classificação popular baseado em aprendizagem de máquina.

Em (ASMUSS; LAUKS, 2015), foram desenvolvidos métodos de classificação do tráfego de rede e detecção de anomalias baseados na análise de séries temporais do tráfego, usando a técnica de agrupamento fuzzy.

Em (SHALAGINOV; FRANKE, 2015), os autores descreveram o estudo em andamento e os primeiros resultados sobre a aplicação do modelo neuro-fuzzy para apoiar a investigação forense do tráfego em larga escala. Já no artigo de (QADER; ADDA; AL-KASASSBEH, 2017), três diferentes algoritmos de mineração de dados foram discutidos como parte da solução proposta para classificação de falhas de rede: K-Means, Fuzzy C-Means e Expectation Maximization. No trabalho de (DUCANGE et al., 2017), foi proposta uma abordagem para tratar o problema de classificação do tráfego usando classificadores fuzzy evolutivos multi-objetivos.

Na pesquisa de (ABDULLAH; AL-HASHMI, 2018), foi proposto um sistema fuzzy evolutivo para discriminar anomalias inspecionando o tráfego da rede. Os resultados comprovaram a adequação do método fuzzy evolutivo de séries temporais para classificação em redes. Na pesquisa de (AL-OBEIDAT; EL-ALFY, 2019), foi concebida uma abordagem de aprendizagem de máquina híbrida supervisionada, para classificação do tráfego de rede, baseado em árvores de decisão fuzzy multi-critério.

Em (IGLESIAS; MILOSEVIC; ZSEBY, 2019), foi apresentada uma abordagem para classificação de ataques em rede baseada em árvores de decisão lineares simples, e árvores de decisão fuzzy.

Na perspectiva de (PARFENOV et al., 2020), a pesquisa teve como objetivo desenvolver um sistema de inferência fuzzy para classificar o tráfego de rede anormal e identificar os ataques atuais por tipo, extraindo regras para o sistema de inferência através de método de árvore de decisão.

Com base nos trabalhos elencados nesta seção, foi buscada uma abordagem híbrida explorando lógica fuzzy valorada intervalarmente e aprendizagem de máquina, para modelar um sistema de inferência e tratar as incertezas e imprecisões na classi-

ficação do tráfego de *streaming* de vídeo denominada FuzzyNetClass.

5.5 Considerações do Capítulo

Neste Capítulo foram apresentados e discutidos trabalho relacionados a esta Tese. Estes trabalhos foram importantes para a definição da abordagem a ser aplicada para a criação da FuzzyNetClass. As buscas de trabalhos relacionados começaram pela abordagem clássica em aprendizagem de máquina, abordagem utilizada pela grande maioria dos trabalhos relacionados a classificação do tráfego de rede. O formato utilizado nos trabalhos pesquisados auxiliaram na definição dos modelos, das ferramentas e na forma como apresentar os resultados obtidos.

Os trabalhos avaliados em lógica fuzzy apresentaram os conceitos teóricos e os algoritmos que serviram como base para as definições implementadas nesta Tese. Na pesquisa realizada não foram encontrados trabalhos que abordassem a classificação de *streaming* de vídeo com aplicação de lógica fuzzy intervalar e com validação em *Datasets* atualizados.

Com a disseminação do uso de criptografia no tráfego das aplicações a análise pelo conteúdo dos pacotes tornou-se uma abordagem pouco promissora. Por outro lado, é possível observar que técnicas baseadas em lógica fuzzy e aprendizagem de máquina, vem ganhando destaque na classificação do tráfego de rede criptografado.

6 FUZZYNETCLASS: MODELAGEM ARQUITETURAL

"Numa folha qualquer,
Eu desenho um sol amarelo,
E com cinco ou seis retas,
É fácil fazer um castelo".

Aquarela
Toquinho

Este Capítulo descreve a arquitetura concebida para a abordagem FuzzyNetClass, as estratégias disponibilizadas para a classificação do tráfego de *streaming* de vídeo, incluindo uma discussão das diferentes etapas contemplas e sua integração.

A Figura 16 apresenta a visão geral da arquitetura proposta para a abordagem FuzzyNetClass, apresentando cada etapa considerada na classificação, sendo caracterizada a opção de escolha do tipo de classificador a ser empregado. Uma das opções é por uma classificação explorando uma abordagem fuzzy e a outra opção é por uma classificação explorando abordagem híbrida.

Segue uma breve apresentação de cada uma das etapas, as quais foram divididas em três classes na visão arquitetural da abordagem FuzzyNetClass, identificadas pela inserção de dados, classificação e extração dos dados, interpretando a geração/manipulação/análise dos diferentes tipos de dados.

6.1 Inserção de Dados

Para realizar a inserção dos dados são consideradas três etapas, Captura de Fluxos de Redes, Extração de Atributos e Seleção de Atributos.

Captura de Fluxos de Rede

O módulo de Captura de Fluxos de Redes é responsável por coletar da infraestrutura os pacotes de rede por meio de uma ferramenta configurada com filtros de seleção de tipos de pacotes e por um tempo limite determinado. O tráfego de rede

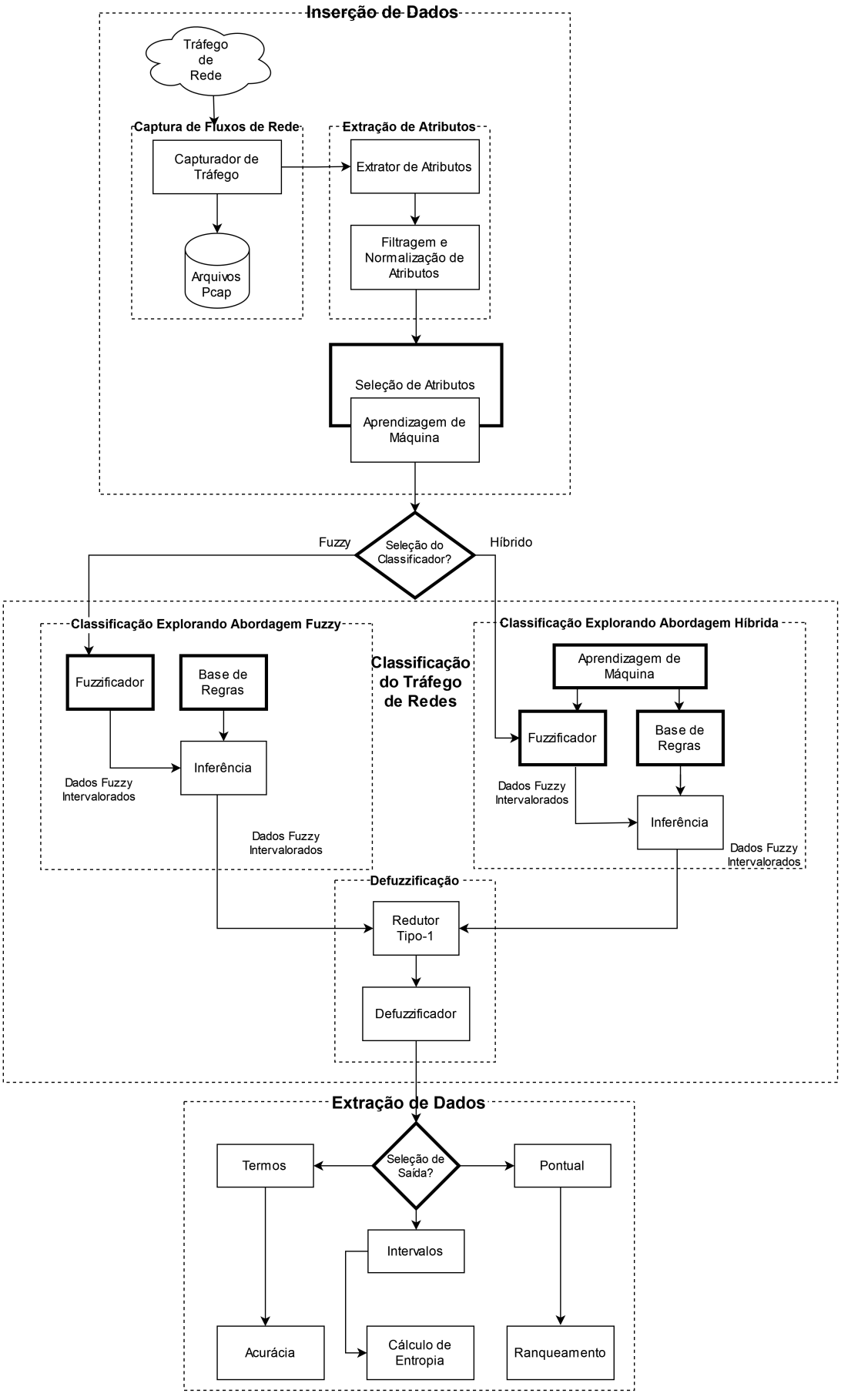


Figura 16 – FuzzyNetClass: Visão Geral da Arquitetura

possui diversos comportamentos que obedecem modelos descritos na literatura, por exemplo o tráfego do tipo *streaming* de vídeo segue o modelo ON-OFF (POYMANOVA; TATARNIKOVA, 2018).

Estes pacotes podem ser tratados de duas maneiras: (i) registrados em arquivos do tipo pcap (*Packet Capture*) empregando as ferramentas Wireshark e/ou Tcpdump; ou (ii) por uma coleta *online*, sendo os pacotes já tratados diretamente pela ferramenta que realiza a montagem dos fluxos de rede e já disponibiliza os seus atributos. Quando se opta pela primeira maneira de operação, conhecida como *offline*, todo processamento de fluxos e atributos acontecem a partir dos arquivos pcap.

Extração de Atributos

O módulo de Extração de Atributos trata da montagem dos fluxos de rede, da extração dos atributos dos fluxos e da filtragem e normalização dos atributos. A etapa realizada pelo Extrator de Atributos compreende o uso de uma ferramenta que realiza a montagem dos fluxos de rede e gera como saída os atributos que caracterizam valores estatísticos do comportamento dos fluxos.

Na etapa de Filtragem e Normalização de Atributos são realizadas as formatações necessárias nos dados de saída da etapa anterior, a redução do número de atributos de acordo com a configuração desejada e são realizados os cálculos de normalização dos atributos.

Seleção de Atributos

O módulo de Seleção de Atributos realiza a escolha dos atributos a serem utilizados no processo de classificação. A escolha dos atributos é realizada por meio do uso de técnicas de aprendizagem de máquina onde são aplicados algoritmos de seleção de atributos. Estes algoritmos são os responsáveis por auxiliar na escolha dos atributos mais relevantes de acordo com o objetivo da classificação desejada.

6.2 Classificação do Tráfego de Redes

Nesta seção são apresentadas as etapas para realização a classificação do tráfego de redes. As etapas consideradas são denominadas de fuzzificação, aprendizagem de máquina, base de regras, inferência e defuzzificação.

6.2.1 Visão das Estratégias de Classificação da Abordagem FuzzyNetClass

Esta seção apresenta as duas estratégias concebidas para os classificadores da abordagem FuzzyNetClass.

Estratégia Explorando Classificação Fuzzy

Esta estratégia tem por foco auxiliar na tomada de decisões referente à classificação de fluxos de rede baseada em lógica fuzzy valorada intervalarmente, sendo construída considerando a visão do especialista. Portanto, esta estratégia está concebida segundo os conceitos fundamentais da Computação Flexível.

Considerando isto, e sobretudo a presença de especialistas na sua instanciação, esta estratégia da FuzzyNetClass é direcionada ao emprego de poucos atributos e um reduzido número de regras, mesmo assim diferentes estudos de caso podem ser processados com esta visão.

A partir da seleção dos atributos pelo especialista, a qual conta com o apoio de aprendizagem de máquina, as variáveis linguísticas e as correspondentes funções de pertinência intervalares podem ser definidas.

Estratégia Explorando Classificação Híbrida

A estratégia da classificação híbrida para abordagem FuzzyNetClass explora a integração de sistemas de inferência baseados em lógica fuzzy valorada intervalarmente com algoritmos de aprendizagem de máquina. Tais algoritmos promovem otimizações no sistema de controle fuzzy, contribuindo para seleção de atributos, para definição das funções de pertinência e na geração da base de regras empregada quando da inferência fuzzy.

A exploração da lógica fuzzy multivalorada no mecanismo de classificação, somada ao conhecimento de especialistas, faculta a obtenção de resultados mais confiáveis e realísticos. Nesta estratégia, a integração com aprendizagem de máquina, provê uma potencial maior flexibilidade para uso da FuzzyNetClass em diferentes estudos de caso e/ou condições do tráfego de rede.

6.2.2 Etapas da Classificação

Nesta seção estão caracterizadas as etapas necessárias às duas estratégias de classificação contempladas na abordagem FuzzyNetClass. Particularmente a etapa de aprendizagem de máquina é explorada na classificação fuzzy quando da etapa de seleção de atributos, diferentemente da abordagem híbrida que a emprega também nas etapas de fuzzificação e geração da base de regras.

Aprendizagem de Máquina

A aplicação de técnicas de aprendizagem de máquina considera prospecção de atributos, incluindo com a geração de regras e funções de pertinência de forma automatizada, resultados de algoritmos de classificação com abordagem em aprendiza-

gem de máquina.

Salientam-se os principais resultados buscados com a integração do raciocínio aproximado provido pela lógica fuzzy e as técnicas de aprendizagem de máquina, na consolidação da abordagem FuzzyNetClass:

(i) construção da base de regra reproduzida durante execução;

(ii) manipulação e extração de regras;

(iii) obtenção de resultados consistentes, gerando bom desempenho e acurácia na extração dos dados;

(iv) suporte ao ajuste de configurações para os pontos limites de funções de pertinências intervalares, e ainda;

(vi) extração e seleção de atributos.

Fuzzificação

A etapa de fuzzificação em um sistema fuzzy baseado em IVFS consiste no módulo de processamento da entrada, considerando todos os aspectos já descritos na Seção 3.4 para modelagem de sistemas baseados em T1FS. Entretanto, esta extensão considera valores fuzzy intervalares modelados por funções de pertinência intervalares. Esta representação para cada conjunto fuzzy $\tilde{A}$ é concebida a partir de pares de funções, $(\underline{\mu}_{\tilde{A}}, \overline{\mu}_{\tilde{A}})$, modelando os limites inferior e superior $(\inf, \sup)$ que limitam a FOU, respectivamente. Ambas funções são definidas por expressões trapezoidais.

Considerou-se oportuna a modelagem dos conjuntos fuzzy empregando funções de pertinência trapezoidais, pois estas são mais simples sendo formadas usando linhas retas. Devido à sua simplicidade de formulação e eficiência computacional, tanto as funções de pertinência trapezoidais quanto as funções triangulares têm sido usadas extensivamente, especialmente em aplicações que tratam dados capturados em tempo real, como é o caso dos *Datasets* correspondentes ao tráfego de rede.

As funções triangulares são empregadas geralmente na representação de números fuzzy, enquanto as trapezoidais representam intervalos fuzzy. Por sua vez, as funções trapezoidais podem ser projetadas a partir de abordagens orientadas a modelos e baseadas em conhecimento. Neste sentido, usar funções de pertinência trapezoidais provê ao usuário mais liberdade na modelagem e desenvolvimento do sistema (WU, 2012).

Assim, na abordagem FuzzyNetClass, a extensão do processo de inferência que atua sobre a base de regras aplica heurísticas baseadas na visão do especialista

e técnicas intervalares para computação dos dados intervalares, representando os graus de pertinência gerados por imagens das funções de pertinência.

O processo de fuzzificação baseado em conjuntos fuzzy valorados intervalarmente é realizado de acordo com a natureza e definição de um conjunto fuzzy (tipo-2) intervalar, associando a cada valor do universo um subintervalo em $[0, 1]$ como valor de entrada e não simplesmente um número em $[0, 1]$. Ou seja, tem-se a inserção no mecanismo de inferência tanto da incerteza relacionada às funções de pertinência de entrada valoradas intervalarmente, quanto da imprecisão das computações geradas pela execução das regras de inferência. E, cada entrada $\mathrm{x} \in \chi$, está associada um vetor bi-dimensional definido pela relação de pertinência, ou seja:

$$\mathbf{x} = (x_1, x_2) \in L([0, 1])^2 \text{ onde } x_1 = \underline{\mu}_{\tilde{A}}(\mathrm{x}) \text{ e } x_2 = \overline{\mu}_{\tilde{A}}(\mathrm{x}), \forall \mathrm{x} \in \chi.$$

Diferentes métodos de fuzzificação são apresentados na literatura, ou seja, após receber uma entrada *crisp*, o módulo pode retornar um conjunto fuzzy que pode ser unitário ou não-unitário, que pode ser um T1FS ou IT2FS (TAN; CHUA, 2007). Este conjunto fuzzificado ativará o módulo de inferência e a base de regras para produzir um conjunto fuzzy de saída. Todos os métodos possuem a mesma estrutura e compartilham a mesma base de regras.

No método de fuzzificação resultando em um conjunto T2FS não-unitário, tem-se um modelo matemático mais adequado para operar com dados imprecisos, e será considerado nesta Tese onde as funções de pertinência inferior e superior são modeladas por T1FS e representam números fuzzy (MENDEL; JOHN; LIU, 2006).

Base de Regras

Etapa constituída por regras que classificam as VL de acordo com os conjuntos fuzzy IvFS. De acordo com (TAN; CHUA, 2007), sejam IVFS $\tilde{A}_i$ que representam termos linguísticos dos antecedentes em T_{x_i}, onde o conjunto fuzzy de ativação são $\tilde{F}_i^l$, com FOU dada por $\left[\underline{F}_i^l, \overline{F}_i^l\right]$, onde $1 \leq i \leq p$, e o IVFS $\tilde{B}$ com termos linguísticos T_y, de IVFS $\tilde{G}^l$, com FOU dada por $\left[\underline{G}_i^l(y), \overline{G}_i^l(y)\right]$. Para simplificar, considere no método de fuzzificação (unitário) a expressão da $\tilde{R}_Z^l$ a l-ésima regra do tipo de Zadeh:

$$\tilde{R}_Z^l : \text{SE } x_1 \text{ é } \tilde{F}_1^l \text{ e } \ldots \text{ e } x_p \text{ é } \tilde{F}_p^l \text{ ENTÃO } y \text{ é } \tilde{G}^l.$$

Inferência

Nesta etapa tem-se a unidade de decisão lógica que realiza as operações de inferência entre os dados de entrada e as condições impostas na base de regras, gerando

a ação a ser realizada no sistema de inferência fuzzy tipo-2 intervalar.

Seja $\mathrm{x} \in \chi$ uma entrada *crisp*, e um sistema baseados em IVFS de Mandani. Para cada conjunto de ativação $F^l(\mathrm{x})$, pela aplicação da regra $\tilde{R}_Z^l$, obtém-se a correspondente expressão da FOU dada pelo intervalo:

$$FOU(F^l(\mathrm{x})) = \left[T_{i=1}^{p} \underline{F}_i^l(x_i), T_{i=1}^{p} \overline{F}_i^l(x_i)\right]$$

A regra $\tilde{R}_Z^l$ ao considerar a entrada dada pela expressão acima, retorna um conjunto de saída $\tilde{B}^l$, cuja FOU é dada pela expressão:

$$FOU(B^l)(y)) = \left[T_{i=1}^{p} \underline{F}_i^l(x_i) \vee \underline{G}^l(y), T_{i=1}^{p} \overline{F}_i^l(x_i) \vee \overline{G}^l(y)\right]$$

onde o operador $\vee$ indica a aplicação de uma t-norma definida sobre IVFS, frequentemente aplicada na t-norma do mínimo.

Em (TAN; CHUA, 2007), o próximo passo na inferência descreve a agregação (união) de todos os conjuntos de ativação das regras genéricas, contendo o IVFS $\tilde{B}$ de saída, com FOU determinada pela expressão:

$$FOU(\tilde{B}(Y)) = [\underline{B}(Y), \overline{B}(Y)] = \left[\underline{B}^1(y) \wedge \ldots \wedge \underline{B}^M(y), \overline{B}^1(y) \wedge \ldots \wedge \overline{B}^M(y)\right]$$

onde o operador $\wedge$ indica a aplicação de uma t-conorma definida sobre IVFS, frequentemente aplicada na t-norma do máximo.

Defuzzificação

Um sistema fuzzy valorado intervalarmente possui, pelo menos, um conjunto fuzzy valorado intervalarmente nos antecedentes ou consequentes de uma regra. Nesta Tese, considera-se SBRF baseados em IVFS onde o defuzzificador é composto de dois estágios, que são eles:

(i) **Redutor de Tipo**: responsável por reduzir um conjunto fuzzy valorado intervalarmente em conjunto fuzzy, ao buscar o melhor conjunto fuzzy que representa o IVFS deve satisfazer a seguinte premissa: quando toda a incerteza desaparecer, o resultado do sistema de inferência fuzzy tipo-2 reduz-se a um sistema de inferência (WU; NIE, 2011).

Em (TAN; CHUA, 2007), a redução de tipo de IVFS é o T1FS cujos elementos são os centroides e os conjuntos imersos no conjunto de entrada $\tilde{A}$. Denota-se o menor e maior valor do centroide nestes conjuntos por $c_l(\tilde{B})$ e $c_r(\tilde{B})$ os quais podem ser calculados pela expressão:

$$c_l(\tilde{B}) = \sum_{i=1}^{L} \frac{\sum_{i=1}^{L} y_i \overline{B}(y_i) + \sum_{i=1}^{L} y_i \underline{B}(y_i)}{\sum_{i=1}^{L} \overline{B}(y_i) + \sum_{i=1}^{L} \underline{B}(y_i)} \qquad c_r(\tilde{B}) = \sum_{i=1}^{L} \frac{\sum_{i=1}^{L} y_i \underline{B}(y_i) + \sum_{i=1}^{L} y_i \overline{B}(y_i)}{\sum_{i=1}^{L} \underline{B}(y_i) + \sum_{i=1}^{L} \overline{B}(y_i)}$$

sempre que $y_1 < \ldots < y_N$ e onde L e R indicam as trocas entre as funções de pertinência de $\tilde{B}$, inferior/superior e superior/inferior, respectivamente.

(ii) **Defuzzificador**: O sistema de inferência fuzzy tipo-2 usa a média dos pontos limites $\underline{Y}$ e $\overline{Y}$ da saída $B(x)$:

$$y = \frac{\underline{Y} + \overline{Y}}{2} = \frac{c_l(\tilde{B}) + cr(\tilde{B})}{2}, \tag{22}$$

onde os valores $\underline{Y}$ e $\overline{Y}$ são calculados via método iterativo de Karnik e Mendel (algoritmo KM)(MENDEL, 2013), ou obtidos através do uso de um método convencional, como o centroide, no valor final da inferência.

6.3 Extração de Dados

A extração dos dados para os resultados gerados na FuzzyNetClass passa pela seleção do tipo de saída de dados, contemplando três possibilidades:

- **Dados *Crisp***: expressão de saída mais simples, dada por um valor numérico, que pode ser interessante para posteriores interpretações e heurísticas, se for o caso de composição e/ou suporte a integração com outros sistemas computacionais. Estas saídas são passíveis de análise dos resultados via métricas e ordenação, mas com a desvantagem de não considerar a incerteza e imprecisão inerente às entradas e/ou geradas durante os processos do controlador fuzzy;

- **Termos Linguísticos**: expressão de saída provendo compreensão imediata e mais intuitiva dos dados processados, pois faz uma interpretação baseada em termos linguísticos da linguagem natural. No caso da abordagem FuzzyNetClass um dado de saída representado pelo termo linguístico “Vídeo Alto” (*HighVideo*) indica que o dado estava classificado com alto grau de pertinência para variável linguística “Video”, interpretando que está bem caracterizado como vídeo de acordo com os atributos selecionados e correspondentes variáveis linguísticas de entrada aplicadas no controlador fuzzy.

- **Dados Intervalares**: expressão dos dados de saída na forma de intervalos, explicita na saída intervalar a extensão e/ou propagação da informação incompleta das entradas e dos processos de fuzzificação, inferência e defuzzificação. Esta estruturação de dados é passível de uma análise dos resultados que considera

a incerteza dos dados de entrada como também a imprecisão inerente a tais processos. E ainda, pode ser estendida para contemplar métricas de análise de desempenho, como a entropia intervalar ou ainda incluir ordenação, por meio do uso de ordens admissíveis, para classificação e/ou hierarquia dos resultados gerados.

6.4 Considerações do Capítulo

Neste Capítulo foi apresentada a visão geral da arquitetura do FuzzNetClass. Foram descritos os módulos que compõe a arquitetura e a interação de cada etapa para a realização da classificação de tráfego de rede.

A concepção da FuzzyNetClass foi realizada, com a perspectiva de estender os métodos tradicionais baseados em lógica clássica e probabilística, incorporando metodologias de Inteligência Computacional, como aprendizagem de máquina, para modelagem de sistemas flexíveis mais realísticos e confiáveis.

Abordagens em T2FS, em especial em IVFS, estendem a abordagem em T1FS, provendo sistemas mais precisos de raciocínio, dedução e computação em que os objetos do discurso e análise estão associados a informação imperfeita. No caso, informação com um ou mais aspectos envolvendo aplicações com dados imprecisos, incertos, vagos, incompletos, parcialmente verdadeiros ou parcialmente possíveis (BUSTINCE et al., 2016).

Deste modo, a FuzzyNetClass foi desenvolvida para lidar com as incertezas e imprecisões referentes aos atributos identificados como relevantes para classificação de fluxos de rede do tipo *streaming* de vídeo e considera que as variáveis tratadas na abordagem são definidas por conjuntos fuzzy valorados intervalarmente.

7 FUZZYNETCLASS: MODELAGEM DAS ETAPAS OPERACIONAIS PARA INSERÇÃO DE DADOS

"Mas eu não vim até aqui,
Para desistir agora...".

Até o Fim
Engenheiros do Hawaii

Neste Capítulo são apresentados os procedimentos e ferramentas para a criação dos *Datasets* e seleção dos atributos principais para classificação do tráfego de rede. Para validar a proposta foram criados *Datasets* a partir de captura do tráfego de rede. A captura do tráfego de rede foi realizada e validada por meio de ferramentas de análise de rede e uma ferramenta para extração de atributos dos fluxos de rede.

A partir dos atributos extraídos dos fluxos foi realizada a seleção dos principais atributos, os quais foram aplicados na etapa de classificação da FuzzyNetClass. Foram utilizadas duas abordagens para seleção dos atributos. Uma das abordagens fez uso da ferramenta WEKA, com algoritmos de seleção de atributos, a outra abordagem fez uso da ferramenta KEEL com as avaliações dos resultados da acurácia e bases de regras de algoritmos de classificação.

7.1 Geração dos *Datasets* Empregados

Para a realização das avaliações da proposta são necessários *Datasets* classificados de forma correta e validados por especialistas. Na pesquisa realizada na literatura foram identificados *Datasets* diversos com finalidades diferentes do objetivo da FuzzyNetClass. O *Dataset* UNB ISCX Network Traffic (VPN-nonVPN) (GERARD DRAPPER GIL, 2022a) [1] disponibilizado pela Universidade de New Brunswick, localizada no Canadá, que é amplamente usado nas validações de trabalhos que tratam de classificação e identificação do tráfego de rede apresentou as características desejadas para este trabalho. Entretanto, foram identificados problemas na forma da

[1] `http://205.174.165.80/CICDataset/ISCX-VPN-NonVPN-2016/`

classificação do tráfego de rede.

Este *Dataset* possui categorias do tráfego de rede tais como *Browsing* (Navegação Web), *E-mail*, *VoIP* e *streaming* de vídeo. Por se tratar de um *Dataset* de 2016, os protocolos de *streaming* de vídeo atuais não estavam disponíveis amplamente, desta maneira tornaria o processo de avaliação incompatível com a proposta dessa Tese. Mesmo com esta prerrogativa, o *Dataset* ISCX foi solicitado aos autores e analisado. Ao analisar a forma de como foram categorizados os fluxos de rede foram identificadas falhas nos filtros aplicados. Por exemplo, na categoria de *streaming* de vídeo há fluxos com menos de 3 pacotes de rede e com uso de portas de comunicação incompatíveis com os protocolos usados para vídeo.

Na Tabela 5, podem ser visualizados fluxos que aparecem categorizados como *video*, com curta duração, observa-se que a duração do fluxo está representada na tabela em escala de tempo em segundos. No primeiro fluxo seriam 0,024951 segundos para duração (*Flow Duration*) e possui apenas um pacote enviado no sentido de *upload* (*Total Fwd Packet*) e apenas um pacote enviado no sentido de *download* (*Total Bwd Packets*).

Em um fluxo de *streaming* de vídeo o comportamento normal seria, no mínimo, de centenas de pacotes no sentido de *download*. Em destaque na Tabela 5 dois fluxos de *streaming* de vídeo onde os valores dos atributos são completamente distintos dos demais fluxos listados. Desta forma, houve o descarte do uso do *Dataset* devido as falhas na forma de como o mesmo foi construído e categorizado.

Outra alternativa encontrada na literatura pesquisada foi a construção de *Datasets* próprios. A grande parte dos *Datasets* não é disponibilizada publicamente e, geralmente, não há forma de contato direto para solicitação juntos aos autores. Aliado a isto, os *Datasets* criados para aplicações específicas, tais como na área de segurança, não se aplicam a proposta dessa Tese.

Outro fato importante para a criação do próprio *Dataset* são os atributos dos fluxos de rede disponibilizados. Não há padronização de quais atributos são extraídos dos fluxos de rede. A opção pelo uso da ferramenta CicFlowMeter para a extração dos atributos teve como objetivo a extração do maior número de atributos possível o que possibilita uma análise mais completa de cada fluxo de rede.

Para a extração dos atributos são necessários os arquivos das capturas em modo *RAW*. O arquivo em modo *RAW* disponibiliza na íntegra os dados dos fluxos e isto pode comprometer a privacidade de usuários e gerar brechas na segurança do ambiente de rede onde os dados foram capturados. Devido a isto, estes tipos de arquivos não são disponibilizados pelos autores dos trabalhos.

Com estas restrições apresentadas nos *Datasets* encontrados houve a opção da criação de *Datasets* próprios, em um ambiente de rede controlado e com o uso de capturas do tráfego de rede em ambientes reais. Para realizar as capturas dos fluxos

Tabela 5 – Exemplos de Fluxos com Problemas no *Dataset* UNB ISCX Network Traffic (VPN-nonVPN)

A	B	C	D	E	Label
0.024951	1	1	0.0	0.0	video
0.025132	1	1	0.0	0.0	video
0.18808	1	1	31.0	313.0	video
0.026112	1	1	0.0	0.0	video
0.025122	1	1	0.0	0.0	video
0.042841	1	1	27.0	291.0	video
4.457922	2	0	0.0	0.0	video
5.906169	2	0	0.0	0.0	video
104.818723	3	5	0.0	63.0	video
104.728554	3	5	0.0	63.0	video
0.997372	3	3	0.0	0.0	video
105.105754	6	4	2026.0	714.0	video
10.211556	8	8	2213.0	870.0	video
0.17722	10	9	920.0	5653.0	video
16.655428	12	9	376.0	6298.0	video
10.068057	13	12	4964.0	2729.0	video
119.998865	21	10	11134.0	2385.0	video
119.876061	**2239**	**4378**	**0.0**	**20725620.0**	**video**
119.971172	**2715**	**4857**	**1220.0**	**23789884.0**	**video**

- **(A)** *Flow Duration* (em segundos), • **(B)** *Total Fwd Packet*,
- **(C)** *Total Bwd packets*, • **(D)** *Total Length of Fwd Packet* (Bytes) e
- **(E)** Total Length of Bwd Packet (Bytes)

de rede de *streaming* de vídeo foi utilizado um ambiente virtualizado.

Para gerar os arquivos de captura no formato .pcap foi usada a ferramenta Tcpdump. Os fluxos foram capturados em um ambiente de acesso doméstico à Internet com um link de 240 Mbit/s com tecnologia GPON (*Gigabit Passive Optical Network*). Destas capturas foram classificados os fluxos de vídeo sob demanda e de Live *streaming* com o uso da ferramenta Wireshark e de *scripts* em Python baseados na biblioteca Pandas.

Para adicionar fluxos de rede de outros tipos foram inseridos arquivos de capturas coletados de várias fontes. Um dos ambientes utilizados para captura do tráfego refere-se a uma rede acadêmica, de universidade no sul do Brasil. As capturas foram realizadas em momentos distintos do dia e em dias da semana diferentes.

Outra fonte utilizada foi por meio de conjuntos de dados públicos (MOUSTAFA; SLAY, 2015; CHO; MITSUYA; KATO, 2000). Os fluxos de rede dos seguintes protocolos foram capturados: DNS, NTP, FTP, SSH, HTTP, HTTPS e QUIC. Para validar os fluxos de vídeo em HTTPS e QUIC foram realizadas análises na ferramenta Wireshark[2] por um especialista em redes.

7.2 Montagem dos Fluxos de Pacotes e Extração de Atributos

Para realizar a montagem dos fluxos de pacotes e a extração dos atributos relacionados foram analisadas ferramentas citadas nos trabalhos encontrados na literatura. Foram analisadas três ferramentas que possuem as funcionalidades desejadas para aplicação nesta Tese. As três ferramentas analisadas foram o NETMATE[3], o Scapy[4] e o CicFlowMeter[5]. Devido aos requisitos necessários para o uso na FuzzyNetClass optou-se pela ferramenta CicFlowMeter.

NETMATE A ferramenta NETMATE (*Network Measurement and Accounting Meter*) possui código-fonte aberto e realiza a exportação de atributos dos fluxos de rede. Entretanto, o projeto não está sendo mais mantido e não possui suporte ao protocolo IPv6. Desta forma, optou-se em não utilizar o NETMATE neste trabalho devido a necessidade da análise de fluxos em IPv6.

Scapy A ferramenta Scapy provê a manipulação de forma interativa de pacotes e fluxos. A partir de capturas de rede ou da leitura de arquivos com tráfego capturado é possível obter estatísticas dos fluxos. O formato dos arquivos é compatível com a biblioteca Libpcap. Para este trabalho, onde o objetivo é trabalhar com os

[2] https://www.wireshark.org/
[3] https://github.com/DanielArndt/netmate-flowcalc
[4] https://scapy.net/
[5] https://github.com/ahlashkari/CICFlowMeter

dados extraídos dos fluxos, o uso de uma ferramenta que demanda o desenvolvimento de todos os procedimentos e funcionalidades para a extração e geração de resultados aumentaria a complexidade de forma desnecessária.

CicFlowMeter A ferramenta CicFlowMeter é um analisador de fluxo do tráfego de rede. A escolha desta ferramenta deu-se à pela ampla adoção na comunidade acadêmica, possuir licença sem custos para uso e ter a capacidade de extração de atributos a partir dos fluxos de rede. Esta ferramenta pode ser utilizada para analisar fluxos bidirecionais, onde o primeiro pacote determina as direções para frente (origem para destino - *upstream*) e para trás (destino para origem - *downstream*).

A escolha da ferramenta CicFlowMeter, utilizada na versão 3.0, possui compatibilidade com a biblioteca Libcap, sendo esta biblioteca padrão para arquivos de captura em rede. As funcionalidades de extração de atributos dos fluxos permitem mais de 85 atributos estatísticos do tráfego de rede como duração do fluxo, número de pacotes, número de bytes, comprimento dos pacotes, entre outros, podem ser calculados separadamente em ambas direções, enviando e recebendo pacotes. Na Tabela 6 estão listados todos os atributos de fluxos e as descrições que a ferramenta disponibiliza para uso.

Procedimento para Extração de Atributos

A ferramenta CicFlowMeter recebe como entrada arquivos de captura salvos no formato da biblioteca Libpcap na opção *Network->Offline* ou realiza a captura do tráfego de rede na opção *Network->Realtime*. Nesta Tese a opção de utilização foi com o uso de arquivos de captura. Os arquivos de captura foram salvos por meio da ferramenta Tcpdump. A ferramenta Tcpdump faz uso do formato de arquivos da bilbioteca Libpcap e permite o uso de filtros para seleção dos fluxos de interesse, por exemplo, para fluxos com as portas 443 nos protocolos UDP ou TCP.

Ao submeter na ferramenta um arquivo ou uma pasta com arquivos de capturas são gerados arquivos separados no formato CSV (*Comma Separated Values*) com os valores dos atributos para cada fluxo detectado. Para definir um fluxo de rede a ferramenta utiliza duas abordagens. A primeira é com o uso de fluxos que utilizam o protocolo TCP. Neste caso o fluxo se encerra de acordo com a finalização por meio dos procedimentos de encerramento padrão, com a troca de segmentos FYN/ACK ou com segmento RST (*Reset*).

Entretanto, as capturas podem não conter todas as trocas de pacotes e a terminação dos mesmos. Neste caso é utilizada a segunda abordagem que se baseia no tempo limite do fluxo (*flow timeout*), parametrizado na ferramenta. Este recurso é

Tabela 6 – Atributos Extraídos dos Fluxos pela Ferramenta CicFlowMeter

Atributo	Descrição
Flow duration	Duração do fluxo em microsegundos
Total Fwd Packet	Total de pacotes em direção de upload
Total Bwd packets	Total de pacotes em direção de download
Total Length of Fwd Packet	Total do tamanho de pacotes em direção de upload
Total Length of Bwd Packet	Total do tamanho de pacotes em direção de download
Fwd Packet Length Min	Tamanho mínimo de pacote em direção de upload
Fwd Packet Length Max	Tamanho máximo de pacote em direção de upload
Fwd Packet Length Mean	Tamanho médio de pacote em direção de upload
Fwd Packet Length Std	Desvio padrão de tamanho de pacote em direção de upload
Bwd Packet Length Min	Tamanho mínimo de pacote em direção de download
Bwd Packet Length Max	Tamanho máximo de pacote em direção de download
Bwd Packet Length Mean	Tamanho médio de pacote em direção de download
Bwd Packet Length Std	Desvio padrão de tamanho de pacote em direção de download
Flow Bytes/s	Número de bytes do fluxo por segundo
Flow Packets/s	Número de pacotes do fluxo por segundo
Flow IAT Mean	Tempo médio entre dois pacotes enviados em um fluxo
Flow IAT Std	Desvio padrão entre dois pacotes enviados em um fluxo
Flow IAT Max	Tempo máximo entre dois pacotes enviados em um fluxo
Flow IAT Min	Tempo mínimo entre dois pacotes enviados em um fluxo
Fwd IAT Min	Tempo mínimo entre dois pacotes enviados em direção de upload
Fwd IAT Max	Tempo máximo entre dois pacotes enviados em direção de upload
Fwd IAT Mean	Tempo médio entre dois pacotes enviados em direção de upload
Fwd IAT Std	Desvio padrão entre dois pacotes enviados em direção de upload
Fwd IAT Total	Tempo total entre dois pacotes enviados em direção de upload
Bwd IAT Min	Tempo mínimo entre dois pacotes enviados em direção de download
Bwd IAT Max	Tempo máximo entre dois pacotes enviados em direção de download
Bwd IAT Mean	Tempo médio entre dois pacotes enviados em direção de download
Bwd IAT Std	Desvio padrão entre dois pacotes enviados em direção de download
Bwd IAT Total	Tempo total entre dois pacotes enviados em direção de download
Fwd PSH flags	Número de ocorrências da flag PUSH enviadas por pacotes no sentido de upload (valor 0 para UDP)
Bwd PSH Flags	Número de ocorrências da flag PUSH enviadas por pacotes no sentido de download (valor 0 para UDP)
Fwd URG Flags	Número de ocorrências da flag URG enviadas por pacotes no sentido de upload (valor 0 para UDP)
Bwd URG Flags	Número de ocorrências da flag URG enviadas por pacotes no sentido de download (valor 0 para UDP)
Fwd Header Length	Total de bytes usados pelos cabeçalhos no sentido de upload
Bwd Header Length	Total de bytes usados pelos cabeçalhos no sentido de download
FWD Packets/s	Número de pacotes por segundo no sentido de upload
Bwd Packets/s	Número de pacotes por segundo no sentido de download
Packet Length Min	Tamanho mínimo de pacote
Packet Length Max	Tamanho máximo de pacote
Packet Length Mean	Tamanho médio de pacote
Packet Length Std	Desvio padrão de tamanho de pacote
Packet Length Variance	Variância de tamanho de pacote
FIN Flag Count	Número de pacotes FIN
SYN Flag Count	Número de pacotes SYN
RST Flag Count	Número de pacotes RST
PSH Flag Count	Número de pacotes PUSH
ACK Flag Count	Número de pacotes ACK
URG Flag Count	Número de pacotes URG
CWR Flag Count	Número de pacotes CWR
ECE Flag Count	Número de pacotes ECE
Down/Up Ratio	Razão entre download e upload
Average Packet Size	Tamanho médio de pacotes
Fwd Segment Size Avg	Tamanho médio de pacote em direção de upload
Bwd Segment Size Avg	Tamanho médio de pacote em direção de download
Fwd Bytes/Bulk Avg	Número médio de bytes em rajadas em direção de upload
Fwd Packet/Bulk Avg	Número médio de pacotes em rajadas em direção de upload
Fwd Bulk Rate Avg	Número médio de rajadas em direção de upload
Bwd Bytes/Bulk Avg	Número médio de bytes em rajadas em direção de download
Bwd Packet/Bulk Avg	Número médio de pacotes em rajadas em direção de download
Bwd Bulk Rate Avg	Número médio de rajadas em direção de download
Subflow Fwd Packets	Número médio de pacotes em subfluxos na direção de upload
Subflow Fwd Bytes	Número médio de bytes em subfluxos na direção de upload
Subflow Bwd Packets	Número médio de pacotes em subfluxos na direção de download
Subflow Bwd Bytes	Número médio de bytes em subfluxos na direção de download
Fwd Init Win bytes	Número total de bytes enviados na janela inicial de transmissão em direção upload
Bwd Init Win bytes	Número total de bytes enviados na janela inicial de transmissão em direção download
Fwd Act Data Pkts	Número de pacotes TCP com ao menos 1 Byte de dados úteis na direção de upload
Fwd Seg Size Min	Tamanho mínimo de segmento na direção de upload
Active Min	Tempo mínimo que um fluxo esteve ativo antes de ser tornar inativo
Active Mean	Tempo médio que um fluxo esteve ativo antes de ser tornar inativo
Active Max	Tempo máximo que um fluxo esteve ativo antes de ser tornar inativo
Active Std	Desvio padrão do tempo no qual um fluxo esteve ativo antes de ser tornar inativo
Idle Min	Tempo mínimo que um fluxo esteve inativo antes de ser tornar ativo
Idle Mean	Tempo médio que um fluxo esteve inativo antes de ser tornar ativo
Idle Max	Tempo máximo que um fluxo esteve inativo antes de ser tornar ativo
Idle Std	Desvio padrão do tempo no qual um fluxo esteve inativo antes de ser tornar ativo

importante porque para fluxos que utilizam o protocolo UDP não há mecanismos de encerramento de troca de dados. O tempo de encerramento do fluxo pode ser definido de duas formas. Usando um tempo limite ou usando um tempo para detecção de inatividade. Por padrão estes tempos são de 120s como tempo e 5s para inatividade de troca de dados do fluxo e podem ser alterados de acordo com os parâmetros desejados pelo usuário.

O consumo de recursos e o tempo para a realização do processamento dos arquivos de captura foram baixos. Neste trabalho foi utilizado um ambiente virtualizado no Vmware Workstation Player 15[6], com 4GB de RAM, 3 cores de CPU e armazenamento de 180GB. O hospedeiro que executou a máquina virtual possui um processador i5-2400 3,1GHz de 4 cores, 8 GB de RAM e disco SSD de 1TB. Para realizar o processamento dos arquivos o tempo será proporcional a quantidade de pacotes e de fluxos identificados pela ferramenta.

Como um exemplo do tempo para processamento, para 374 arquivos de capturas com 4456 fluxos, na configuração citada, foram consumidos cerca de 4 minutos. Portanto, mesmo em um ambiente de recursos computacionais modestos a ferramenta Cicflowmeter obteve um bom desempenho na extração de atributos dos fluxos de rede.

Para realizar as conversões de formatos, analisar de forma estatística os dados, realizar a normalização dos valores dos atributos e demais procedimentos de adaptações de formatos de regras usou-se a linguagem Python com as bibliotecas Pandas [7] e NumPy [8] para criação de códigos. Os arquivos dos códigos utilizados nesta Tese estão disponíveis em repositório público [9].

7.3 Processo de Seleção de Atributos

Para realizar a seleção de atributos relevantes a classificação de *streaming* de vídeo foi desenvolvida uma metodologia que está ilustrada na Figura 17. Cada etapa da metodologia será discutida nas seções a seguir.

7.3.1 Captura de Tráfego

Na etapa de captura de tráfego foram capturados pacotes com a ferramenta Tcpdump, armazenando em arquivos no formato pcap. A partir dos arquivos de captura foram criados 4 *Datasets* nomeados A, B, C e D, concebidos com diferentes tipos de fluxos de rede para cada protocolo. A variação da quantidade de fluxos em cada *Dataset* teve como critério o volume de fluxos de *streaming* de vídeo que foram coletados

[6] https://https://docs.vmware.com/en/VMware-Workstation-Player/15/rn/player-15-release-notes.html

[7] https://pandas.pydata.org/

[8] https://numpy.org/

[9] https://github.com/emmonks/FuzzyNetClass/tree/main/Python

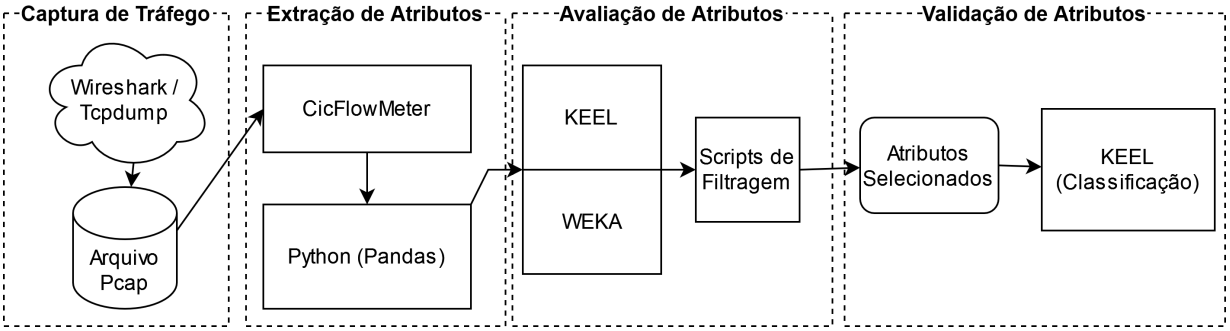


Figura 17 – Visão Geral da Metodologia Desenvolvida para Seleção de Atributos

e classificados de forma manual.

Os *Datasets* criados de forma não balanceada tentam reproduzir de forma mais próxima possível o comportamento da distribuição de fluxos em um ambiente real de rede, onde não há um padrão determinístico de tipos de protocolos. Cabe destacar que todos os *Datasets*[10] estão disponíveis para acesso público.

Na Tabela 7, estão descritos os 4 *Datasets* usados para as avaliações, com a quantidade de fluxos em cada uma das categorias do tráfego de rede classificados.

Tabela 7 – Descrições dos *Datasets* Nomeados A, B, C e D

Protocolo	***Dataset* A**	***Dataset* B**	***Dataset* C**	***Dataset* D**
VoD	232	232	361	361
Live	425	232	361	361
HTTP	600	232	361	1000
HTTPS	600	232	361	1000
QUIC	600	232	361	1000
DNS	600	232	361	1000
Outros	600	232	361	1000

7.3.2 Extração de Atributos

Na etapa de extração de atributos utilizou-se a ferramenta Cicflometer com arquivos de tráfego de rede já capturados. A partir dos arquivos de saída da ferramenta foram executados *scripts* para análise e normalização dos valores dos atributos. Esta etapa tem como objetivo preparar os atributos para serem submetidos a etapa de avaliação de atributos.

Na Figura 18, mostra a interface da ferramenta CicFlowMeter executar uma extração de dados de um arquivo de captura., onde pode-se observar as opções de configuração de tempo limite.

Para a criação dos *Datasets* A, B, C, e D, os arquivos capturados em formato pcap foram submetidos à ferramenta CicFlowMeter com os seguintes parâmetros: 1200s de tempo limite de fluxo e tempo de inatividade em 5s.

[10] `https://l.ufpel.edu.br/emmonks/`

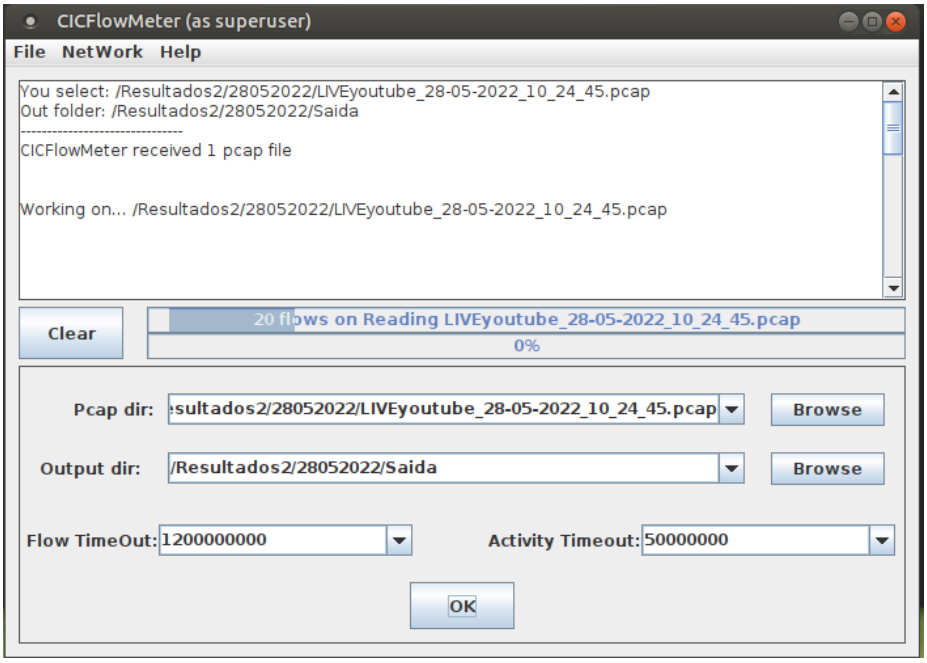


Figura 18 – Extração de Atributos em Sendo Processada na Ferramenta CicFlowMeter

A ferramenta gera arquivos no formato CSV para cada arquivo de captura. O tempo para geração dos arquivos de saída depende da quantidade de fluxos identificados na captura. Em casos de um volume grande de fluxos, foram particionados os arquivos de captura com o auxílio da ferramenta Wireshark.

Após a geração dos arquivos em formato CSV foram utilizados *scripts* em Python com a bilbioteca Pandas (TEAM, 2020) para realizar filtros e gerar estatísticas dos fluxos. Todos os atributos relacionados a identificação dos fluxos, tais como endereços IP e portas de comunicação, foram removidos. Os atributos com valores em zero também foram removidos. Para agrupar os arquivos CSV gerados em um único arquivo usou-se script desenvolvido em Linux Shell *script*.

Normalizações

Para realizar o processo de fuzzificação os valores dos atributos dos fluxos devem ser normalizados em uma escala de 0 a 10. Desta forma, foram realizadas análises dos conjuntos de dados e aplicadas as remoções dos *outliers*. O processo utilizou a remoção de 0,1% dos valores máximos e mínimos de cada atributo. Além disto, os valores normalizados obedeceram a escala de 0 a 10, sendo o valor 10 como limite máximo absoluto.

Para realizar a filtragem e processamento dos valores foi utilizado um script em Python com as bibliotecas Pandas e NumPy (VAN DER WALT; COLBERT; VAROQUAUX, 2011). Na Tabela 8, estão listados os valores de cada atributo usados para normalização.

Tabela 8 – Valores Utilizados para Normalização em cada Atributo

Atributo	Valor Referência
Fwd Packet Length Mean	1232 Bytes
Fwd Packet Length Std	504.741276 Bytes
Bwd Packet Length Mean	1329.456959 Bytes
Bwd Packet Length Std	527.643453 Bytes
Flow IAT Mean	15012.885500 μs
Flow IAT Std	3849188.047082 μs
Fwd IAT Mean	15012.010875 μs
Fwd IAT Std	4403924.347515 μs
Bwd IAT Mean	3057.140563 μs
Packet Length Mean	1290.164944 Bytes
Packet Length Std	564.249638 Bytes

7.3.3 Avaliação de Atributos

A seleção de atributos dos fluxos de rede é fundamental para a classificação do protocolo. Nesta etapa, foram utilizadas duas ferramentas: WEKA e KEEL.

A partir do arquivo de saída gerado pela ferramenta CicFlowMeter, o arquivo foi convertido para o formato .arff, utilizado pelo WEKA. Essa conversão requer a inclusão de um cabeçalho com informações sobre os campos, atributos, e uma indicação de rótulo, referente à categoria de tupla de atributos de cada linha.

Os atributos não numéricos, atributos de identificação de endereço IP, portas de comunicação e protocolo da camada de transporte foram removidos para permitir o uso do WEKA. Esses atributos não foram utilizados no processo de classificação e, portanto, um total de 38 atributos foram utilizados na análise com os algoritmos disponíveis na ferramenta. A Figura 19 mostra os procedimentos aplicados nas ferramentas WEKA e KEEL para selecionar os atributos mais relevantes para classificar o tráfego de *streaming* de vídeo.

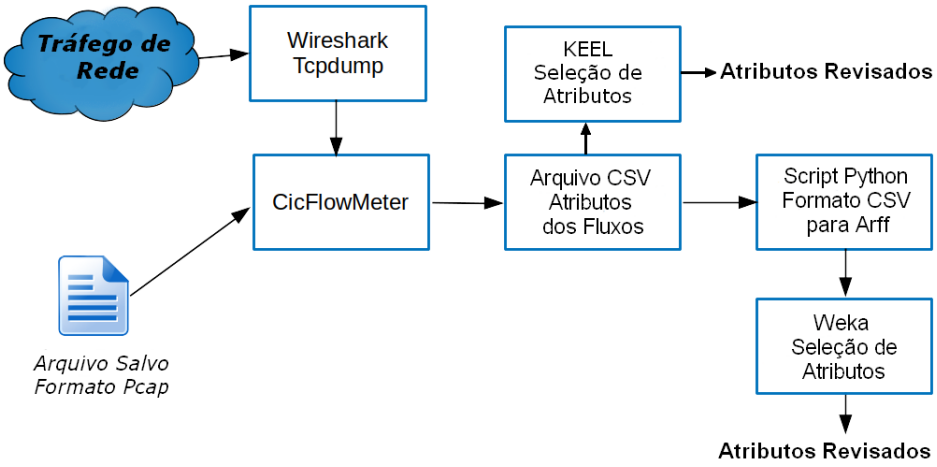


Figura 19 – Procedimentos para a Avaliação de Atributos

Na ferramenta WEKA foi usado o algoritmo de seleção de atributos CfsSubsetEval

(*Correlation-based Feature Subset Selection*) (HALL; SMITH, 1999). Este algoritmo foi utilizado para selecionar os atributos mais relevantes para a classificação do tráfego de rede, pois obteve melhores resultados nos trabalhos encontrados em na literatura (GNANAMBAL et al., 2018), (SHAFIQ; YU; WANG, 2017) e (MATHUR; RAHEJA; AHLAWAT, 2018), e nos experimentos realizados nesta Tese. No WEKA, o algoritmo de seleção de atributos CfsSubsetEval foi utilizado com três métodos de busca *BestFirst*, *Evolutionary Search* e *GreedyStepwis*.

Para realizar uma análise comparativa com os resultados obtidos do algoritmo *CfsSubsetEval* foram usados os algoritmos de seleção de atributos *InfoGainAttributeEval* e *ChiSquareAttributeEval* (HALL, 1998), ambos com o método de avaliação *Ranker*.

Na Tabela 9 são apresentados os resultados da seleção dos atributos para cada avaliador utilizado, com os respectivos métodos de busca. Foram selecionados os atributos que obtiveram maior índice de pontuação em cada um dos avaliadores. Não foram considerados os atributos relacionados aos valores mínimos e aos valores máximos, por exemplo, atributos tais como *Packet Length Min* (Tamanho Mínimo de Pacote) e *Packet Length Max* (Tamanho Máximo de Pacote).

Tabela 9 – Resultados do Avaliadores na Ferramenta WEKA

Avaliador	**Método de Busca**	**Atributos**
CfsSubsetEval	BestFirst	**Bwd_Packet_Length_Std**, Bwd_Header_Length, **Packet_Length_Std**, Bwd_Bytes/Bulk_Avg
CfsSubsetEval	Evolutionary Search	**Bwd_Packet_Length_Mean**, Fwd_IAT_Std, Fwd_Header_Length, Bwd_Header_Length, **Packet_Length_Std**, **Packet_Length_Variance**, Down/Up_Ratio, **Average_Packet_Size**, Bwd_Segment_Size_Avg, Bwd_Bulk_Rate_Avg
CfsSubsetEval	Greedy Stepwise	**Bwd_Packet_Length_Std**, Bwd_Header_Length, **Packet_Length_Std**, Bwd_Bytes/Bulk_Avg
InfoGainAttibuteEval	Ranker	Fwd_Header_Length, **Packet_Length_Std**, **Packet_Length_Variance**, **Average_Packet_Size**, **Packet_Length_Mean**, Down/Up_Ratio, Bwd_Segment_Size_Avg, **Bwd_Packet_Length_Mean**
ChiSquareAttributeEval	Ranker	Fwd_Header_Length, Packet_Length_Std, **Packet_Length_Variance**, **Average_Packet_Size**, **Packet_Length_Mean**, Down/Up_Ratio, Bwd_Segment_Size_Avg, **Bwd_Packet_Length_Mean**, Fwd_IAT_Std, Fwd_Act_Data_Pkts, **Bwd_Packet_Length_Std**

Os resultados dos avaliadores apresentaram repetições de atributos nos diversos métodos aplicados. Deve-se notar que em todos os algoritmos houve seleção de atributos relacionados aos tamanhos dos pacotes, em destaque na Tabela 9. Esta constatação indicou que atributos desta natureza seriam importantes no processo de classificação.

Com base nesses resultados, um especialista em rede realizou a análise dos atributos. A escolha dos atributos foi baseada em valores médios, desvio padrão ou para alguns casos particulares, os valores totais. Em relação aos valores máximos e mínimos, os mesmos possuem muita variação. A inconsistência de tempo e valores extremos ocorrem devido à variação dos recursos da rede que podem causar atrasos variáveis nas entregas de pacotes e retransmissões/perdas de pacotes.

Outra técnica utilizada foi a filtragem baseada na ferramenta KEEL. Aplicou-se essa abordagem para validar os atributos mais relevantes para a construção das regras. Inicialmente, foram utilizados 38 atributos extraídos dos fluxos de rede listados na Tabela 10. Os 38 atributos foram escolhidos a partir do não uso de atributos com valores mínimos, máximos, valores específicos para protocolos e para valores zerados. Os valores mínimos e máximos não se mostraram confiáveis devido a não representarem valores de atributos independentes aos dados gerados nos testes realizados com os algoritmos de seleção de atributos e de classificação.

Tabela 10 – Atributos Aplicados nos Algoritmos de Classificação

Atributo	**Descrição**
Active_Mean	AVG do tempo que um fluxo esteve ativo antes de ser tornar inativo
Active_Std	STD do tempo no qual um fluxo esteve ativo antes de ser tornar inativo
Average_Packet_Size	AVG do tamanho de pacotes
Bwd_Bulk_Rate_Avg	AVG do número médio de rajadas em direção de BWD
Bwd_BytesBulk_Avg	AVG de bytes em rajadas em direção de BWD
Bwd_Header_Length	Total de bytes usados pelos cabeçalhos no sentido de BWD
Bwd_IAT_Mean	AVG do tempo entre dois pacotes enviados em direção de BWD
Bwd_IAT_Std	STD entre dois pacotes enviados em direção de BWD
Bwd_IAT_Total	Tempo total entre dois pacotes enviados em direção de BWD
Bwd_PacketBulk_Avg	AVG do número de pacotes em rajadas em direção de BWD
Bwd_Packet_Length_Mean	AVG do tamanho de pacote em direção de BWD
Bwd_Packet_Length_Std	STD do tamanho de PKT em direção de BWD
Bwd_Packets/s	Número de PKT por segundo no sentido de BWD
Bwd_Segment_Size_Avg	AVG do tamanho de PKT em direção de BWD
DownUp_Ratio	Razão entre BWD e FWD
Flow_Bytes/s	Número de bytes do fluxo por segundo
Flow_Duration	Duração do fluxo em microssegundos
Flow_IAT_Mean	AVG do tempo entre dois PKT enviados em um fluxo
Flow_IAT_Std	STD entre dois PKT enviados em um fluxo
Flow_Packets/s	Número de PKT do fluxo por segundo
Fwd_Act_Data_Pkts	Contagem de PKT TCP sem carga em FWD
Fwd_Header_Length	Total de bytes usados pelos cabeçalhos no sentido de FWD
Fwd_IAT_Mean	AVG do tempo entre dois PKT enviados em direção de FWD
Fwd_IAT_Std	STD entre dois PKT enviados em direção de FWD
Fwd_IAT_Total	Tempo total entre dois PKT enviados em direção de FWD
Fwd_Packet_Length_Mean	AVG do tamanho de PKT em direção de FWD
Fwd_Packet_Length_Std	STD de tamanho de PKT em direção de FWD
Fwd_Packetss	Número de PKT por segundo no sentido de FWD
Fwd_Segment_Size_Avg	AVG do tamanho de PKT em direção de FWD
Idle_Mean	AVG do tempo que um fluxo esteve inativo antes de ser tornar ativo
Idle_Std	STD do tempo no qual um fluxo esteve inativo antes de ser tornar ativo
Packet_Length_Mean	AVG do tamanho de PKT
Packet_Length_Std	STD de tamanho de pacote
Packet_Length_Variance	VAR de tamanho de pacote
Total_Bwd_packets	Total PKTs em BWD
Total_Fwd_Packet	Total PKTs em FWD
Total_Length_of_Bwd_Packet	Volume total de PKT em BWD
Total_Length_of_Fwd_Packet	Volume total de PKT em FWD

Legenda: Desvio Padrão (STD), Média (AVG), Pacotes (PKT),
Direção de *Download* (BWD), Direção de *Upload* (FWD), Variância (VAR)

Dos resultados dos experimentos realizados com os algoritmos CHIRW, FARCHD, FURIA e IVTURS, B, foram extraídas regras com confiança maior que 0,8 para as *streaming* de vídeo nos formatos *VoD* e *Live*, e também extraídos os atributos com

maior quantidade de participação significativa nas composições das regras resultantes das execuções dos algoritmos de classificação.

Foi realizado um processo de filtragem com o uso de *scripts* em Python para extrair os atributos mais usados nas regras de cada algoritmo. Assim, foram obtidos 14 atributos da primeira rodada de filtragem.

Na segunda rodada de filtragem, o processo foi refeito gerando 11 atributos com maior ocorrência na composição das regras mais confiáveis, com valores acima de 0,8. A Tabela 11 contém os 11 atributos resultantes dos processos de filtragem e dos resultados processados nas ferramentas WEKA e KEEL.

Tabela 11 – Atributos Selecionados

Atributo	Descrição
Fwd Packet Length Mean	Tamanho médio de pacote em direção de upload
Fwd Packet Length Std	Desvio padrão de tamanho de pacote em direção de upload
Bwd Packet Length Mean	Tamanho médio de pacote em direção de download
Bwd Packet Length Std	Desvio padrão de tamanho de pacote em direção de download
Flow IAT Mean	Tempo médio entre dois pacotes enviados em um fluxo
Flow IAT Std	Desvio padrão entre dois pacotes enviados em um fluxo
Fwd IAT Mean	Tempo médio entre dois pacotes enviados em direção de upload
Fwd IAT Std	Desvio padrão entre dois pacotes enviados em direção de upload
Bwd IAT Mean	Tempo médio entre dois pacotes enviados em direção de download
Packet Length Mean	Tamanho médio de pacote
Packet Length Std	Desvio padrão de tamanho de pacote

7.3.4 Validação de Atributos

Para validar os resultados obtidos em um método que obteve o auxílio de um especialista na área para a montagem das regras foram utilizadas outras abordagens por meio da ferramenta KEEL. No estudo de caso 2, na Seção 8.2, está apresentado e discutido o processo de validação de atributos.

7.4 Considerações do Capítulo

Neste Capítulo foi discutida a modelagem operacional da abordagem FuzzyNetClass. O processo de geração dos *Datasets* empregados nesta Tese foi necessário devido as alternativas de *Datasets* encontrados na literatura e que são disponibilizados de forma pública, não atenderem os requisitos pretendidos para o tipo de tráfego de rede em *streaming* de vídeo. As ferramentas utilizadas para seleção e extração de atributos propiciaram a geração de *Datasets* confiáveis e validados para as avaliações dos estudos de caso.

As etapas apresentadas proveram meios para validar a abordagem e os atributos selecionados para a realização das avaliações. Por meio de procedimentos com o uso de algoritmos de aprendizagem de máquina para seleção de atributos obteve-se como resultado 11 atributos extraídos dos fluxos de rede. Os atributos selecionados neste Capítulo serão explorados nos estudos de caso.

8 FUZZYNETCLASS: ESTUDOS DE CASO

"Tudo está tão certo que parece
errado,
É onde não consigo me achar,
Luzes da verdade na realidade,
Sempre estão mudando de lugar".

Rolam as Pedras
Kiko Zambianchi

Neste Capítulo são apresentados e discutidos os estudos de casos baseados na abordagem FuzzyNetClass. No primeiro estudo de caso são apresentados os resultados da classificação explorando a abordagem fuzzy, com um número reduzido de atributos e regras. No segundo estudo de caso são apresentados os resultados da análise dos impactos dos atributos na classificação de *streaming* de vídeo. No terceiro estudo de caso são considerados os resultados da FuzzyNetClass explorando a classificação em uma abordagem híbrida, com o uso de um conjunto maior de atributos e com uma base de regras adaptadas por meio de algoritmos de aprendizagem de máquina.

As diferentes etapas que levaram aos resultados dos estudos de caso descritos neste Capítulo, bem como os diferentes códigos fonte da FuzzyNetClass, estão disponíveis em dois repositórios públicos construídos empregando a plataforma GitHub, (1) o repositório incluindo os códigos fonte referentes a abordagem com número reduzido de atributos [1] ; (2) o repositório contendo os códigos fontes da abordagem híbrida [2].

8.1 Estudo Caso 1: Classificação com Abordagem Fuzzy

Neste estudo de caso foi avaliada a acurácia da abordagem FuzzyNetClass quanto a classificação de *streaming* de vídeo. Como especialistas na área, foram envolvidos

[1] `https://github.com/brunomourapaz/FuzzyNetClass`
[2] `https://github.com/emmonks/FuzzyNetClass`

integrantes da CREI (Coordenação de Redes e Infraestrutura) da UFPEL. Esta equipe tem sete integrantes que atuam especificamente em gerenciamento de redes e tem como missão central manter operacional uma infraestrutura que atende 20 mil usuários, e cujas conexões de rede se distribuem em mais de 50 prédios.

8.1.1 Descrição do Estudo de Caso

Os especialistas da CREI/UFPEL participaram modelando as funções de pertinência, bem como selecionando os atributos a serem empregados nas mesmas. Neste estudo de caso, particularmente a seleção de atributos foi feita com auxílio de algoritmos de aprendizagem de máquina disponibilizados pela ferramenta WEKA, cuja relação está no Apêndice B.

Na construção dos *Datasets* empregados no estudo de caso foram capturados 300 fluxos de *streaming* de vídeo decorrentes da reprodução dos mesmos na plataforma Youtube, todos no formato VoD (*Video on Demand*). O tempo de captura de cada vídeo variou entre 2 a 4 minutos, permitindo o registro de eventuais alterações nos parâmetros operacionais da rede, os quais podem impactar, mas nem sempre o fazem, na qualidade da exibição à intermitência e variações nas condições da rede.

Para reproduzir os vídeos foi usado o navegador Firefox versão 105, configuração padrão, sendo executado em um sistema operacional Linux Ubuntu 20.04. Os vídeos foram capturados ao longo de três meses com sessões de captura em diferentes momentos do dia, durante a semana e finais de semana. A captura dos vídeos deu-se em ambiente de acesso à Internet com link de 240 Mbit/s com tecnologia GPON (*Gigabit Passive Optical Network*).

As capturas dos fluxos de rede foram realizadas com o emprego da ferramenta Tcpdump[3], a qual com o emprego de filtros facultou a supressão de fluxos que sabidamente não se tratavam de *streaming* de vídeo. Esta ferramenta foi selecionada tendo como critérios centrais sua maturidade, constante atualização e adoção perante as comunidades nacionais e internacionais que atuam na área.

Os fluxos de *streaming* de vídeo foram salvos em arquivos no formato pcap, os quais foram submetidos à ferramenta CicFlowMeter para a extração dos atributos. Para garantir os intervalos previstos de 2 a 4 minutos para a captura a CicFlowMeter foi configurada com os parâmetros para definição de fluxos de rede em 1200s de *timeout* e tempo de inatividade em 5s.

Nas capturas realizadas na plataforma Youtube o protocolo QUIC predominou no controle do transporte dos vídeos analisados entre cliente e servidor, havendo poucos casos onde o protocolo HTTPS foi utilizado. Ambos protocolos QUIC e HTTPS utilizam criptografia por padrão. Foram construídos dois *Datasets*, com tamanhos e características diferentes, um deles com 246 e outro com 54 fluxos de *streaming* de

[3] `https://www.tcpdump.org/`

Tabela 12 – Composição dos Fluxos por *Dataset*

Protocolo	Dataset 17102021
DNS	1025
FTP	1937
HTTP	25
SSH	5
Vídeo	246
Total de Fluxos	3346

Protocolo	Dataset 24102021
DNS	663
HTTP	2941
HTTPS	2941
NTP	63
Outros	6040
QUIC	304
Vídeo	54
Total de Fluxos	11304

vídeo (vide Tabela 12).

Com o intuito de conferir aos dados de entrada um perfil mais próximo das redes reais, arquivos adicionais foram combinados com os *Datasets*. Estes arquivos foram gerados a partir de duas fontes: (a) capturas feitas durante o desenvolvimento desta Tese em uma rede acadêmica de uma universidade no sul do Brasil; e, (b) de capturas disponibilizadas publicamente, feitas em instituições de distintas origens (MOUSTAFA; SLAY, 2015; CHO; MITSUYA; KATO, 2000).

Destes arquivos adicionais, foram extraídos fluxos de rede para os seguintes protocolos: DNS, NTP, FTP, SSH, HTTP, HTTPS e QUIC. Para validar os fluxos HTTPS e QUIC, comumente usados para *streaming* de vídeo, houve a utilização da ferramenta Wireshark[4] e o auxílio de um especialista em rede. A seleção destes protocolos ante as capturas a serem adicionadas aos *Datasets*, teve por objetivo evitar que fossem introduzidos como tráfegos adicionais, fluxos pertinentes *streaming* de vídeo, comprometendo o acompanhamento do processo como um todo (vide Tabela 12)

Os fluxos de rede adicionais, foram explorados como a seguir:

- no *Dataset* 17102021 foram agregados 3100 fluxos;
- no *Dataset* 24102021 foram agregados 11250 fluxos;

Para realizar a seleção dos atributos foi empregado o algoritmo CfsSubsetEval disponível na ferramenta WEKA. Sua aplicação sobre os *Datasets* retornou os atributos com maior relevância. Após análise dos especialistas da CREI/UFPEL, foram selecionados os atributos *Packet Length Mean*, *Fwd Packet Length Std* e *Backward IAT Total*. Os valores dos atributos escolhidos foram normalizados em uma escala de 0 a 10, sendo o 10 o valor máximo possível.

Neste estudo de caso, foi considerada a classificação explorando abordagem fuzzy, na qual são executadas todas as etapas do sistema de inferência fuzzy de valor intervalar. A saída fornece o nível do fluxo analisado, que tem como objetivo realizar a

[4] `https://www.wireshark.org/`

classificação do fluxo de rede referente ao tipo de *streaming* de vídeo: *Low*, *Average* e *High*.

Neste cenário, a FuzzyNetClass utilizou uma base de regras composta de 27 regras e 3 termos linguísticos. O processo de classificação foi realizado com o uso da biblioteca Juzzy (WAGNER, 2013).

A plataforma Juzzy foi selecionada porque apresenta alternativas para modelagem e implementação de sistemas fuzzy convencionais e fuzzy valorados intervalarmente, além disso, por ser um projeto de código-fonte aberto atualmente disponível para comunidade acadêmica, e que recentemente abordou o tratamento para conjuntos fuzzy valorados intervalarmente de intervalo restrito (D'ALTERIO et al., 2020) (vide Apêndice C).

O método de fuzzificação considerou funções de pertinência do tipo trapezoidal. O processo de inferência utilizado foi com base no método de Mamdani (MAMDANI, 1976), considerando uma base de regras com conectivos lógicos do tipo "AND" aplicando normas triangulares. E por fim, a etapa de defuzzificação utilizando o centro da área.

Esta situação é oportuna para uma melhor interpretabilidade pelo usuário, pois as funções de pertinência mais simples são formadas usando linhas retas. Devido à sua simplicidade de formulação e eficiência computacional, tanto as funções de pertinência trapezoidais quanto as funções triangulares têm sido usadas extensivamente, especialmente em aplicações que tratam dados capturados em tempo real.

Base de Dados e Funções de Pertinência

Os valores dos atributos selecionados são aplicados à escala padrão considerando o intervalo $[0, 10]$, estabelecendo o valor 10 como limite para os valores acima dele. Assim, para *Packet Length Mean*, foi usado a Eq. (23), *Fwd Packet Length Std* a Eq. (24) e *Backward Iat Total* Eq. (25), na obtenção dos graus de pertinência:

$$PLM = (nf_i(PLM)/MaxPLM * 10) \quad (23)$$

$$PLS = (nf_i(PLS)/MaxPLS * 10) \quad (24)$$

$$BIAT = (nf_i(BIAT)/MaxBIAT) * 10 \quad (25)$$

considerando os seguintes parâmetros para cada fluxo de rede:

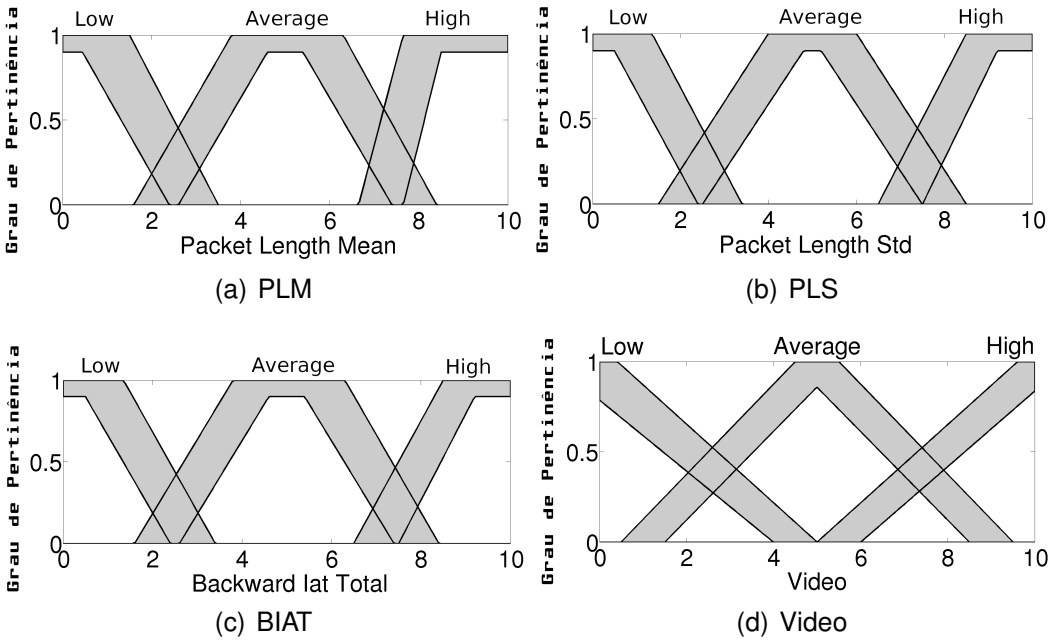
- nf_i representa um fluxo de rede capturado;
- PLM é um atributo médio de tamanho de pacote;
- PLS é um atributo de desvio padrão de comprimento de pacote no sentido de envio;

- $BIAT$ considera o tempo total de chegadas de pacotes no sentido de recebimento;
- $\max PLM$ é o valor total do atributo médio de tamanho de pacote mais alto identificado;
- $\max PLS$ é o valor total do maior desvio padrão do tamanho do pacote no sentido de envio identificado;
- $\max BIAT$ é o valor total do atributo de tempo total entre chegadas de recebimentos de pacotes mais alto identificado.

Os termos linguísticos que definem os conjuntos para variável *Packet Length Mean* (PLM) são os seguintes: “Low” (PLML), “Average” (PLMR) e “High” (PLMH - melhor caso). Denota-se $PLM = a$ e $a \in [0, 10]$. Na Figura 20(a) são representadas as correspondentes funções de pertinência.

O atributo *Fwd Packet Length Std* (PLS) é usado como entrada e obtido pela leitura do fluxo de rede analisado. Os termos para os conjuntos definidos para esta variável são: “Low” (PLSL), “Average” (PLSR - melhor caso) e “High” (PLSH). Denota-se $PLS = b$ e $b \in [0, 10]$. Estas funções de pertinência são apresentadas na Figura 20(b).

Na modelagem dos conjuntos fuzzy para BIAT (*Backward Iat Total*), foram criados os seguintes termos: “Low” (BIATL - melhor caso), “ Average” (BIATR) e “High” (BIATH). Denota-se $BIAT = c$ e $c \in [0, 10]$. Estas funções de pertinência são vistas na Figura 20(c).



(a) PLM (b) PLS

(c) BIAT (d) Video

Figura 20 – Funções de Pertinência dos Atributos de Entrada (PLM, PLS, BIAT) e Atributo de Saída (Classificação de Vídeo)

Fuzzificação

Nessa etapa, ocorre o mapeamento dos valores de entrada, já normalizados para o intervalo de $[0, 10]$, para o domínio fuzzy, como apresenta a Figura 21.

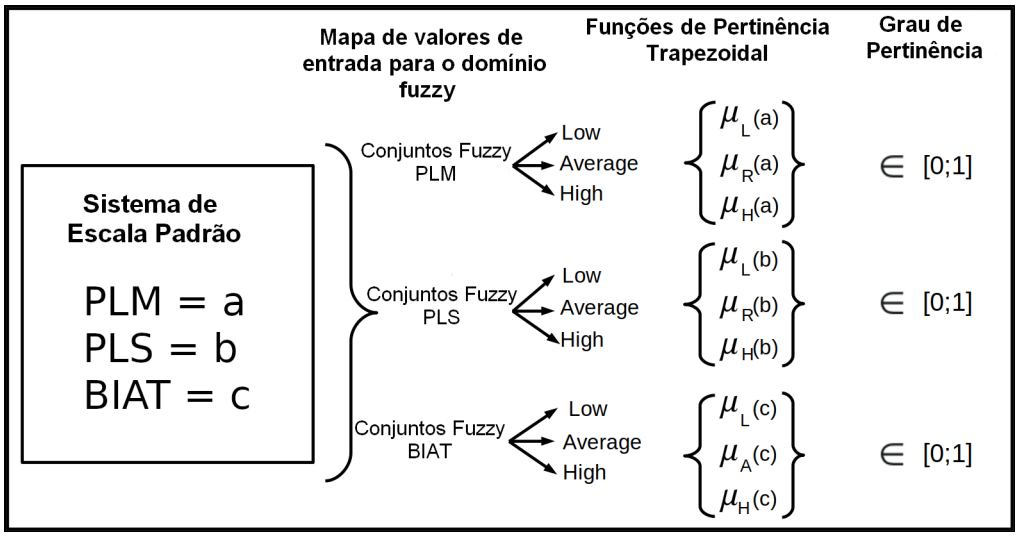


Figura 21 – Processo de Fuzzificação dos Atributos de Entrada: PLM, PLS e BIAT

Base de Regras

A Base de Regras da FuzzyNetClass foi desenvolvida com intuito de ser compreensível e editável descrevendo de maneira consistente a estratégia de controle considerando três fatores: (i) as variáveis linguísticas nomeiam os conjuntos fuzzy, tornando a modelagem do sistema mais próxima do mundo real; (ii) são utilizadas conexões lógicas do tipo “AND” para criar a relação entre as variáveis de entrada; (iii) as implicações são do tipo $Modus\ Ponens$ (modo afirmativo): “Se ((x é A) e (y é B)) então (z é C).”

Na Tabela 13 está apresentada a base de regras aplicada para classificação de fluxos de *streaming* de vídeo.

Inferência

No processo de Inferência, tem-se operações entre conjuntos, combinação dos antecedentes das regras e implicações modeladas através do operador *Modus Ponens Generalizado*, ocorrendo em três etapas:

(i) Aplicação da Operação Fuzzy: nesta etapa, ocorre a aplicação dos operadores fuzzy sendo que a entrada consta de três valores resultantes da fuzzificação. Como as regras são formadas pelo operador fuzzy “AND”, a aplicação utiliza o método MIN (mínimo) sobre os dois valores retornados da fuzzificação;

Tabela 13 – Base de Regras da FuzzyNetClass

	PLM		PLS		BIAT		VÍDEO
	PLML		PLSL		BIATL		Average Video
	PLML		PLSL		BIATR		Low Video
	PLML		PLSL		BIATH		Low Video
	PLML		PLSR		BIATL		Average Video
	PLML		PLSR		BIATR		Low Video
	PLML		PLSR		BIATH		Low Video
	PLML		PLSH		BIATL		Low Video
	PLML		PLSH		BIATR		Low Video
	PLML		PLSH		BIATH		Low Video
	PLMR		PLSL		BIATL		Average Video
	PLMR		PLSL		BIATR		Low Video
	PLMR		PLSL		BIATH		Low Video
IF	PLMR	**AND**	PLSR	**AND**	BIATL	**THEN**	Average Video
	PLMR		PLSR		BIATR		Low Video
	PLMR		PLSR		BIATH		Low Video
	PLMR		PLSH		BIATL		Low Video
	PLMR		PLSH		BIATR		Low Video
	PLMR		PLSH		BIATH		Low Video
	PLMH		PLSL		BIATL		Average Video
	PLMH		PLSL		BIATR		Average Video
	PLMH		PLSL		BIATH		Low Video
	PLMH		PLSR		BIATR		Average Video
	PLMH		PLSR		BIATH		Low Video
	PLMH		PLSR		BIATL		High Video
	PLMH		PLSH		BIATR		Low Video
	PLMH		PLSH		BIATH		Low Video
	PLMH		PLSH		BIATL		Average Video

(ii) Aplicação do Método de Implicação Fuzzy: realizada pela combinação entre o valor obtido na aplicação do operador fuzzy e os valores do conjunto fuzzy de saída da regra, utilizando o método MIN (mínimo) sobre estas combinações;

(iii) Aplicação do Método de Agregação Fuzzy: considera a composição dos resultados fuzzy da saída de cada regra, utilizando o método MAX (máximo), assim criando uma única região fuzzy para ser analisada pelo próximo processo do módulo.

Defuzzificação

Nessa etapa, ocorre a transformação da região resultado da Inferência em um valor discreto que representa o fluxo de rede analisado. Essa transformação foi realizada através do emprego do método centro da área. Esse método calcula o centroide da área composta pela saída fuzzy do sistema de Inferência, o qual considera a união de todas as especificações da base de regras. A expressão analítica do centroide é dada pela equação:

$$u = \frac{\sum_{i=1}^{N} u_i \mu_{OUT}(u_i)}{\sum_{i=1}^{N} \mu_{OUT}(u_i)} \tag{26}$$

8.1.2 Discussão dos Resultados

Na Tabela 14 são apresentados os resultados da execução da abordagem FuzzyNetClass, sendo destacado os percentuais para a quantidade de fluxos classificados em cada grupo, neste caso quanto ao grau de pertinência para cada conjunto de saída.

Tabela 14 – Classificação de *Streaming* de Vídeo - Resultados Resumidos

Dataset	**TFV**	**Vídeo**						**Acurácia**	
		High		**Average**		**Low**		**C**	**CoS**
		C	**CoS**	**C**	**CoS**	**C**	**CoS**	**Average/ HighVideo**	**Average/HighVideo**
17102021	246	108	116	51	116	87	32	64.63%	94.31%
24102021	54	34	37	8	8	13	9	77.78%	83.33%

• **(TFV)** Total de Fluxos de Vídeo, • **(CoS)** Centro dos Conjuntos • **(C)** Centroide

Quanto a acurácia, no caso do *Dataset* 17102021, houve 64,63%, com o uso do centroide para a redução e 94,31%, com o uso do centro dos conjuntos. Por sua vez, no caso do *Dataset* 24102021, foi obtido o percentual de 77,78% para aplicação que usa do centroide, e 83,33%, para uso do centro dos conjuntos. Neste estudo de caso, destacou-se o redutor CoS. Os fluxos de *streaming* de vídeo foram na sua maioria classificados nas faixas que correspondem aos IVFS com variável de saída *video* e termos linguísticos *Average* ou *High*.

Entretanto, a análise dos fluxos de *streaming* de vídeo classificados na faixa que correspondem aos IVFS com variável de saída *Average* ainda se mostra significativa, em ambos redutores de tipos. Assim, a abordagem proposta demanda novos delineamentos, que busquem simultaneamente, decremento no número de elementos do IVFS referente ao termo *Average* e incremento, quando refere-se ao termo *High*.

Os valores médios para intervalos com limites inferiores e superiores para os conjuntos definidos, a partir dos termos linguísticos da variável de saída, para cada um dos *Datasets*, estão apresentados nos gráficos da Figura 22.

Nestes gráficos, verifica-se que a variável *Vídeo* obteve bons resultados para os limites inferiores e superiores da variável linguística *HighVideo*. E, como esperado, o mesmo não ocorreu com outros tipos de fluxo de rede, que este estudo considerou como DNS, FTP, HTTP, SSH para *Dataset* 17102021, e DNS, HTTP, HTTPS, NTP, outros e QUIC para *Dataset* 24102021.

Estes valores médios gerados, ou sejam, limitantes inferiores e superiores para os intervalos de saída, mostram que os elementos dos demais tipos de fluxos não tem pertinência significativa no conjunto fuzzy modelado como *HighVideo*, em ambos os casos dos *Datasets* 17102021 e 24102021.

Estes valores, para os demais tipos fluxos, exceto *Vídeo*, obtiveram um alto grau de pertinência para as variáveis linguísticas *UpperAverageVideo* e *LowerAverageVi-*

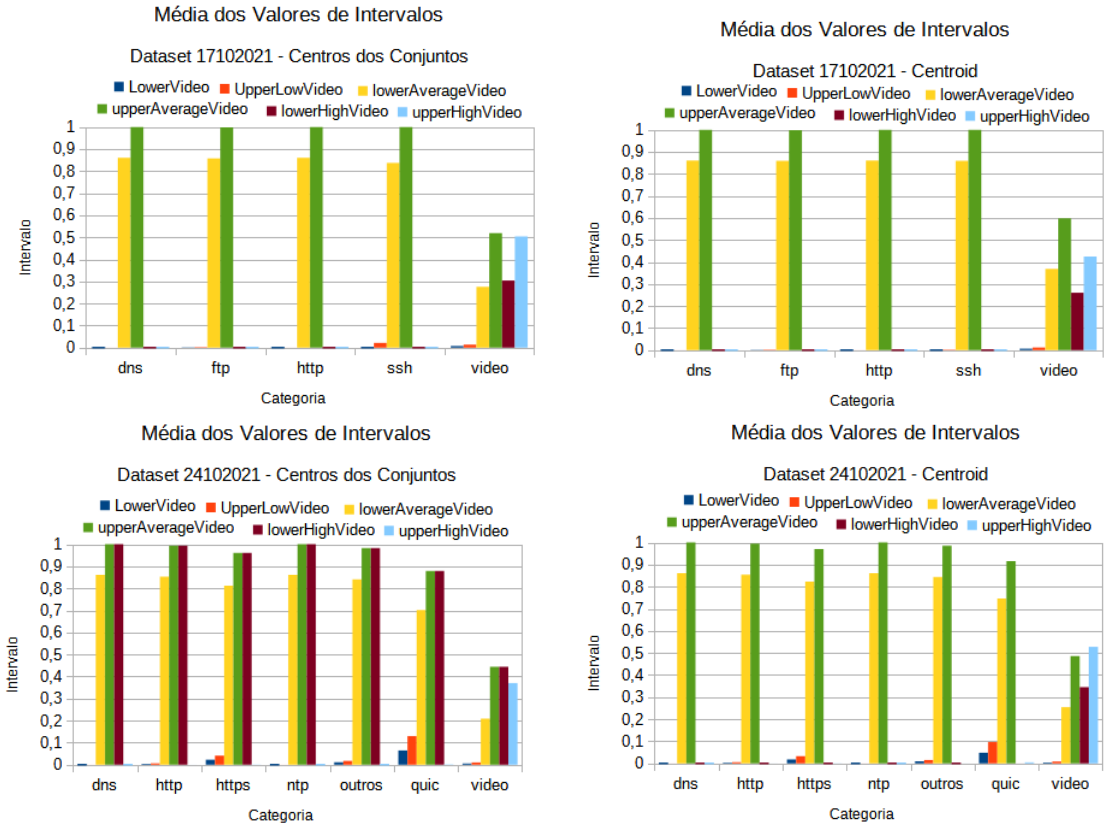


Figura 22 – Valores Médios dos Limites Inferiores e Superiores para as Variáveis considerando nos Datasets 17102021 e 24102021 Redutores C e CoS

deo. Desta forma, mostram relevante indecisão na classificação, e apontam para uma extensão da abordagem FuzzyNetClass, considerando novas variáveis e por conseguinte nova base de regras. Estas observações deverão ser consideradas nos novos estudos de caso, considerando a integração da aprendizagem de máquina também na geração das funções de pertinência e na construção da base de regras.

Interpretação dos Resultados via Gráficos de Dispersão

Na Figura 23 tem-se os gráficos de dispersão para os pontos médios de cada intervalo de saída, para todos os conjuntos fuzzy gerados pelos dois redutores: (C) Centroide à esquerda; e, (CoS) Centro dos Conjuntos à direita.

Os dois gráficos na parte superior referem-se aos resultados obtidos pela aplicação do *Dataset* 17102021, enquanto que os dois inferiores, ao *Dataset* 24102021.

Observa-se dispersão mais acentuada em todos os gráficos na variável linguística *Low*, quando se comparado com as outras classes de fluxo que não são do tipo *Vídeo*. Isso está de acordo com a proposta de intensificar a classificação de apenas fluxos de vídeo nos termos de *Average* e *High*.

Deve-se ressaltar ainda que, para os gráficos referentes a redução via centroide,

observa-se que ocorreram na área destinada à variável *Vídeo* mais pontos azuis, indicando pertinência *High*, e verdes, pertinência *Average*, do que pontos vermelhos o que demostra um melhor desempenho deste redutor. Ou seja, quanto menor a quantificação de pontos vermelhos na área restrita à classe *Vídeo*, melhor será a classificação e organização da informação para esta variável.

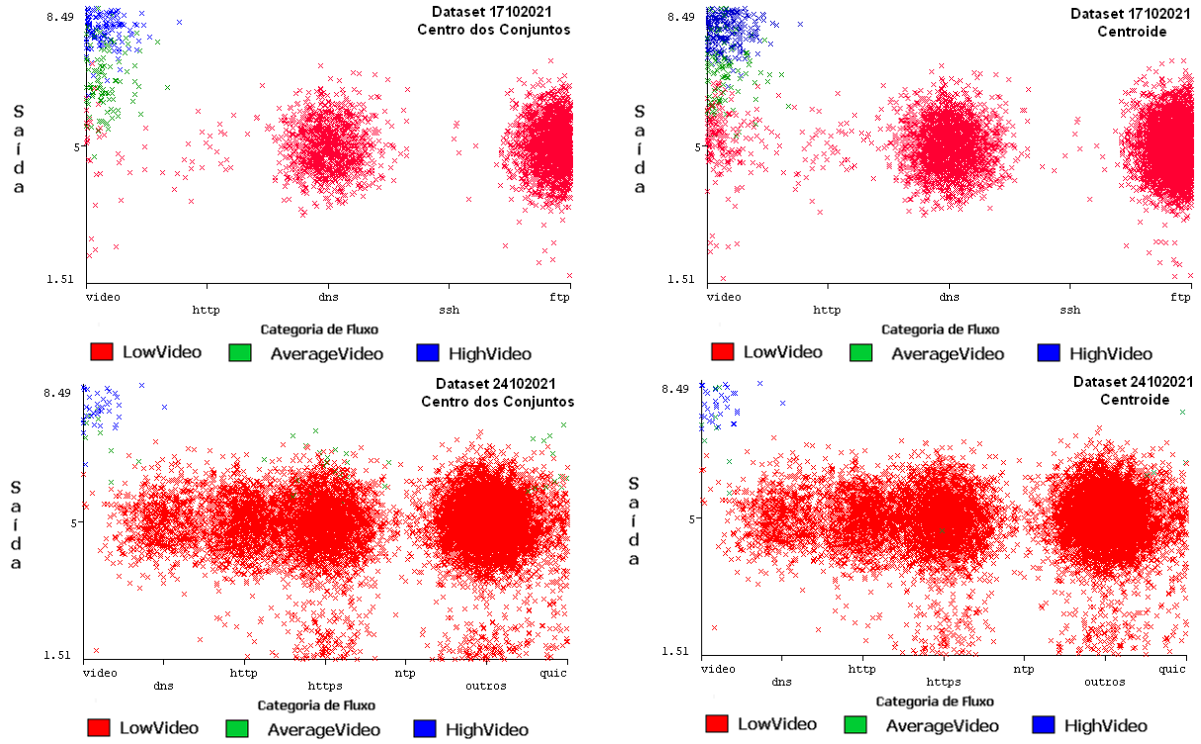


Figura 23 – Dispersão dos Valores Pontuais para os *Datasets* 17102021 e 24102021 com o Uso dos Redutores C e CoS

Interpretação dos Resultados via Ferramenta KEEL

Para validação do cenário referente a este estudo de caso, foram utilizados métodos automatizados por meio da ferramenta KEEL e com o uso de algoritmos baseados em ambas áreas de pesquisa, abordagens em lógica fuzzy e aprendizagem de máquina. Foram aplicados os *Datasets* (A) 17102021 e 24102021(B) em 8 algoritmos e analisados os resultados para acurácia e F1 Score. A escolha dos 8 algoritmos teve como critério o uso de algoritmos baseados em lógica fuzzy e em aprendizagem de máquina.

Nos gráficos das Figura 24 estão os resultados das execuções na ferramenta KEEL e o desempenho dos atributos escolhidos para a abordagem FuzzyNetClass. Foram comparados os resultados desta abordagem com 4 algoritmos baseados em lógica fuzzy disponíveis na ferramenta KEEL que são o CHI, FURIA, FARCHD e IVTURS.

Os resultados por *Dataset* são distintos para os algoritmos baseados em lógica

fuzzy e para o FuzzNetClass. No caso do *Dataset* 24102021 os algoritmos FARCHD e CHI não conseguiram classificar nenhum *streaming* de vídeo e no Dataset 17102021 obtiveram acurácia alta. O algoritmo FURIA obteve o melhor resultado de acurácia em ambos *Datasets*. A acurácia da abordagem FuzzyNetClass manteve-se com nível similar nos dois *Datasets* testados.

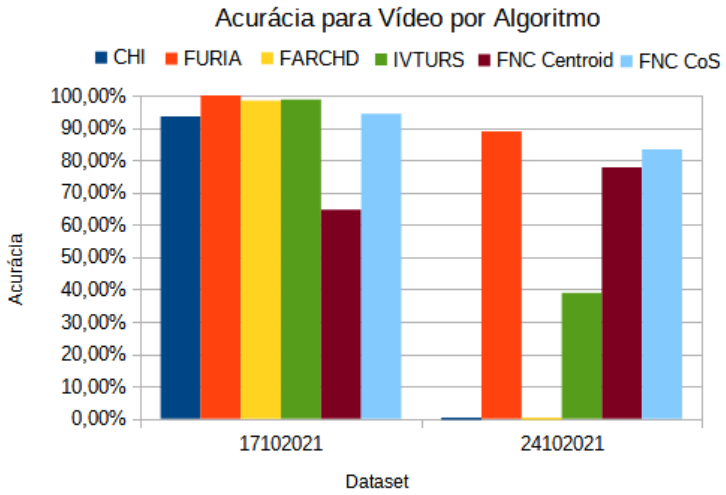


Figura 24 – Acurácia para *Streaming* de Vídeo por Algoritmo, Gerado na Ferramenta KEEL e na Abordagem FuzzyNetClass

A Figura 25 tem-se a quantificação de regras geradas com o uso de algoritmos fuzzy na ferramenta KEEL e a quantidade de regras aplicadas na abordagem FuzzyNetClass. Observa-se que a quantidade de regras geradas nos algoritmos fuzzy são próximas da abordagem FuzzyNetClass, com exceção do algoritmo FURIA que gerou 246 regras no *Dataset* 24102021 e 46 regras no *Dataset* 17102021. Assim, observou-se que a quantidade de regras geradas no FURIA resulta em um nível de acurácia maior do que os demais algoritmos.

Esta análise corrobora para o desenvolvimento de um classificador híbrido para a abordagem FuzzyNetClass, usufruindo de um conjunto maior de regras, a partir de maior número atributos, visando melhores resultados.

Interpretação dos Resultados via Matrizes de Confusão

Nas Tabelas 15, 16, 17 e 18 são listadas as matrizes de confusão para cada um dos *Datasets* com o uso de algoritmos de classificação fuzzy e de aprendizagem de máquina executados na ferramenta KEEL. Pode-se observar que os *Datasets* apresentaram resultados diferentes de acordo com a utilização do tipo de algoritmo, algo previsto devido as imprecisões e incertezas dos fluxos de rede.

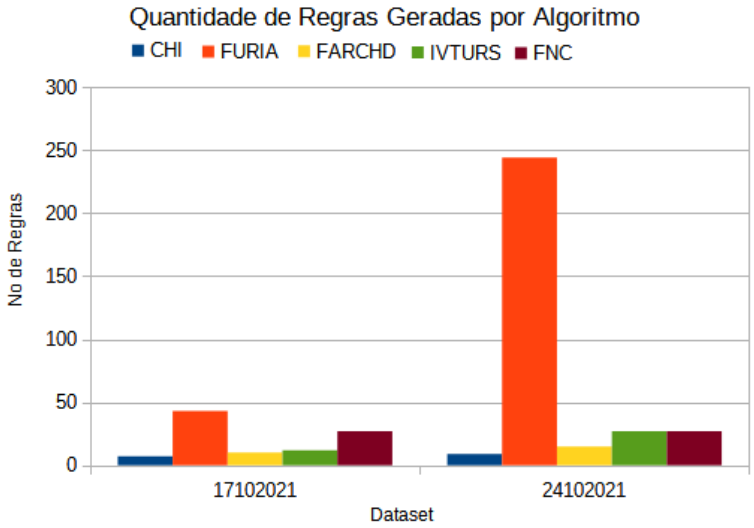


Figura 25 – Quantidade de Regras por Algoritmo Gerado na Ferramenta KEEL e de Forma Manual

Tabela 15 – Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores Fuzzy para os Fluxos Contidos no *Dataset* 17102021

Chi-RW						**FURIA**					
	vídeo	**http**	**dns**	**ssh**	**ftp**		**vídeo**	**http**	**dns**	**ssh**	**ftp**
vídeo	**230**	0	0	0	16	**vídeo**	**244**	2	0	0	0
http	1	0	0	0	24	**http**	3	20	2	0	0
dns	0	0	4	0	1020	**dns**	0	0	1025	0	0
ssh	0	0	0	0	5	**ssh**	0	0	1	4	0
ftp	0	0	1	0	1936	**ftp**	0	0	1	0	1936
FARCHD						**IVTURS**					
	vídeo	**http**	**dns**	**ssh**	**ftp**		**vídeo**	**http**	**dns**	**ssh**	**ftp**
vídeo	**244**	0	0	0	2	**vídeo**	**245**	0	0	0	1
http	5	0	0	0	20	**http**	12	0	0	0	13
dns	1	0	0	0	1024	**dns**	3	0	0	0	1022
ssh	0	0	0	0	5	**ssh**	0	0	0	0	5
ftp	0	0	0	0	1937	**ftp**	1	0	0	0	1936

Tabela 16 – Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores em Aprendizagem de Máquina para os Fluxos Contidos no *Dataset* 17102021.

SVM					
	vídeo	**http**	**dns**	**ssh**	**ftp**
vídeo	**246**	0	0	0	0
http	0	22	3	0	0
dns	0	1	1024	0	0
ssh	0	0	1	0	4
ftp	0	0	1	0	1936

C45					
	vídeo	**http**	**dns**	**ssh**	**ftp**
vídeo	**246**	0	0	0	0
http	1	20	4	0	0
dns	0	4	1021	0	0
ssh	0	2	1	0	2
ftp	0	0	2	0	1935

KNN					
	vídeo	**http**	**dns**	**ssh**	**ftp**
vídeo	**242**	4	0	0	0
http	3	19	2	0	1
dns	0	0	1025	0	0
ssh	0	0	1	4	0
ftp	0	0	1	1	1935

GANN					
	vídeo	**http**	**dns**	**ssh**	**ftp**
vídeo	**242**	4	0	0	0
http	5	16	1	0	3
dns	0	1	1022	0	2
ssh	1	0	0	1	3
ftp	0	0	1	1	1935

Tabela 17 – Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores em Aprendizagem de Máquina para os Fluxos Contidos no *Dataset* 24102021

SVM							
	video	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
video	**41**	0	0	12	0	0	1
dns	0	613	0	6	0	44	0
http	0	0	638	108	0	529	27
https	0	29	225	1312	0	1354	21
ntp	0	0	0	0	0	63	0
outros	0	287	36	192	0	5523	2
quic	0	0	1	145	0	12	146

C45							
	video	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
video	**44**	0	1	6	0	1	2
dns	0	639	0	10	0	13	1
http	0	0	972	159	0	163	8
https	8	24	80	2444	0	368	17
ntp	0	0	0	0	63	0	0
outros	1	9	14	93	1	5916	6
quic	1	0	2	85	0	2	214

KNN							
	video	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
video	12	0	5	29	0	8	0
dns	0	610	2	34	2	14	1
http	6	0	995	280	0	14	7
https	38	33	516	2164	10	101	79
ntp	0	0	0	57	6	0	0
outros	3	251	19	4432	0	1331	4
quic	0	4	8	127	0	12	153

GANN							
	video	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
video	**0**	0	9	45	0	0	0
dns	3	182	75	6	0	396	1
http	0	77	517	123	0	585	0
https	2	35	151	1656	2	1085	10
ntp	0	0	2	0	0	61	0
outros	1	10	30	368	0	5631	0
quic	0	1	5	166	0	131	1

Tabela 18 – Matrizes de Confusão Obtidas por Meio de Múltiplos Algoritmos Classificadores Fuzzy para os Fluxos Contidos no *Dataset* 24102021.

Chi-RW							
	vídeo	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
vídeo	0	0	0	43	0	11	0
dns	0	0	0	0	0	663	0
http	0	0	0	43	0	1258	0
https	0	0	0	902	0	2039	0
ntp	0	0	0	0	0	63	0
outros	0	0	0	262	0	5778	0
quic	0	0	0	138	0	166	0

FURIA							
	vídeo	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
vídeo	42	0	1	9	0	1	1
dns	0	629	0	7	0	26	1
http	2	0	1011	192	0	90	7
https	7	29	101	2371	0	410	23
ntp	0	0	0	0	63	0	0
outros	0	14	22	111	0	5887	6
quic	1	0	0	105	0	2	196

FARCHD							
	vídeo	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
vídeo	0	0	0	54	0	0	0
dns	0	0	0	30	0	633	0
http	0	0	0	781	0	521	0
https	0	0	0	2011	0	930	0
ntp	0	0	0	5	0	58	0
outros	0	0	0	355	0	5685	0
quic	0	0	0	259	0	45	0

IVTURS							
	vídeo	**dns**	**http**	**https**	**ntp**	**outros**	**quic**
vídeo	0	0	0	54	0	0	0
dns	0	0	0	17	0	646	0
http	0	0	0	612	0	690	0
https	0	0	0	1973	0	968	0
ntp	0	0	0	5	0	58	0
outros	0	0	0	472	0	5568	0
quic	0	0	0	173	0	131	0

Validação dos Resultados via Entropia Intervalar

Nesta seção é apresentada a interpretação dos resultados empregando como métrica a entropia intervalar definida pela Eq.(21), quando $p = 1$ e agregação $\mathbb{M}$ dada pela extensão intervalar para média aritmética.

Para esta análise, considera-se que a saída de uma função intervalar pode exibir mais ou menos incerteza do que suas entradas. Esta incerteza foi preservada pelo diâmetro dos intervalos, refletindo a falta de conhecimento preciso sobre o grau de pertinência do elemento ao conjunto fuzzy modelado nas entradas, incerteza também propagada durante o processo nas suas computações.

Na Tabela 19 são apresentados os resultados obtidos com o emprego da métrica de entropia intervalar para as variáveis de entrada.

Tabela 19 – Resultados da Métrica de Entropia

Variáveis de Entrada							
Dataset	**Variável**	**Low**		**Average**		**High**	
		Inf	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**
17102021	PacketLengthMean	0.09729000	0.09724900	0.00161660	0.00864620	5.3145e-18	0.01611000
	PacketLengthStd	0.00355660	0.11025000	0.00927300	0.03279900	0	0.00143550
	BackwardIatTotal	0.00101320	0.10987000	0.00014442	0.00056497	1.0286e-19	0.09265000
24102021	**Variável**	**Low**		**Average**		**High**	
		Inf	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**
	PacketLengthMean	0.01100700	0.16651000	0.00875740	0.04522600	1.5823e-18	0.00965060
	PacketLengthStd	0.01169300	0.13957000	0.01566700	0.05588100	0	0.03206600
	BackwardIatTotal	1.1102e-16	0.10000000	0	0	0	0

Verificam-se que, para ambos *Datasets*, o cálculo da entropia intervalar resultam :

(i) em intervalos com reduzido diâmetro, mostrando controle da modelagem imprecisão, na aferição do especialista;

(ii) intervalos com valores extremos muito próximos de zeros (onde o maior extremo superior recebe o valor 0,16651000) interpretando que a entropia intervalar é baixa. E, portanto, provendo interpretação flexível para organização destas informações, as quais estão baseadas na “mancha” de incerteza que define os conjuntos fuzzy das variáveis de entrada.

Entretanto, a modelagem mostra-se pouco realística, pois a modelagem de algumas variáveis retorna entropia zerada. Em especial, a variável *BackwardIatTotal*, que representa o intervalo de tempo total calculado pela variação do tempo de chegada entre dois ou mais pacotes consecutivos no fluxo da rede.

No próximo estudo de caso, com base nesta análise da abordagem FuzzyNetClass, o atributo associado a variáveis com entropia zerada será substituído por outros que levam a melhores resultados nesta análise. E ainda, novas regras deverão consequente serem propostas.

A análise da entropia na extração dos dados de saída da abordagem FuzzyNetClass está baseada nos resultados da Tabela 20, incluindo os valores referentes ao cálculo do diâmetro (W) dos termos linguísticos da variável de saída.

Tabela 20 – Resultados da Métrica de Entropia na Saída

Variável de saída										
Dataset	**RT**	**Low**			**Average**			**High**		
		Inf	**Sup**	W	**Inf**	**Sup**	W	**Inf**	**Sup**	W
17102021	C	0.0017565	0.0027411	0.0010	0.0099868	0.1565400	0.1466	0.0235030	0.0359840	0.0125
	CoS	0.0018194	0.0030468	0.0012	0.0158330	0.1635800	0.1477	0.0273060	0.0424910	0.0152
24102021	C	0.0166840	0.0253350	0.0087	0.0090564	0.1533600	0.1443	0.0018777	0.0030364	0.0012
	CoS	0.0163460	0.0265660	0.0102	0.0096736	0.1546300	0.1450	0.0023659	0.0039699	0.0016

• **(RT)** Redutor • **(CoS)** Centro dos Conjuntos • **(C)** Centroide • (W) Diâmetro

Esta análise também considerou dois *Datasets* e dois operadores de redução de tipo (RT) nas execuções da FuzzyNetClass para obtenção de resultados, que são eles: (i) Centroide (C) e (ii) Centro dos Conjuntos (CoS). Em todos os casos analisados, tem-se intervalos com diâmetro reduzido, e cada valor para entropia, referente aos extremos superiores das funções de pertinência para as VL da variável de saída, se manteve baixo, menor ou igual a 0.1635800, no caso *AverageVideo* e redutor CoS.

Com o uso de três atributos para classificação de *streaming* de vídeo, observou-se neste estudo de caso resultados promissores para a classificação explorando somente lógica fuzzy, na qual a seleção de atributos foi auxiliada por aprendizagem de máquina.

Com o objetivo de aumentar o número de atributos utilizados na classificação e por consequência o tamanho da base de regras, os próximos estudos de caso ampliam a discussão referente ao emprego de aprendizagem de máquina nas etapas de seleção de atributos e inferência fuzzy.

8.2 Estudo de Caso 2: Impacto dos Atributos na Classificação

O objetivo central deste estudo de caso foi a avaliação do impacto dos 11 atributos selecionados no Capítulo 7 para a classificação de *streaming* de vídeo.

8.2.1 Descrição do Estudo de Caso

Neste estudo de caso para a análise do impacto destes atributos foram realizadas avaliações com o uso de 8 algoritmos para classificação, sendo 4 com abordagem fuzzy e 4 com abordagem em aprendizagem de máquina. Na Tabela 21 estão listados todos os algoritmos utilizados neste estudo de caso.

Tabela 21 – Algoritmos Selecionados para o Estudo de Caso

Algoritmo	**Abordagem**
CHI-RW	Lógica Fuzzy Tipo 1
FARC-HD	Lógica Fuzzy Tipo 1
FURIA	Lógica Fuzzy Tipo 1
IVTURS	Lógica Fuzzy Valorada Intervalarmente
SVM	Aprendizagem de Máquina
C45	Aprendizagem de Máquina
KNN	Aprendizagem de Máquina
GANN	Aprendizagem de Máquina

Estes classificadores foram selecionados considerando sua aceitação pela comunidade técnico-científica, seu alinhamento com as demandas de processamento da abordagem FuzzyNetClass, bem como pelo fato de serem softwares disponibilizados por meio de códigos-fonte abertos. Uma descrição dos mesmos pode ser encontrada no Apêndice B.

Para cada algoritmo foram analisados os resultados de sua execução configurados com os 11 atributos selecionados para classificação de fluxos de rede em *streaming* de vídeo em formato de *VoD* ou em formato de *Live*. As classificações foram realizadas para os *Datasets* A, B, C e D, já descritos na Tabela 7, na seção 7.3.1.

Para realizar a análise comparativa do impacto dos atributos foi utilizada a ferramenta KEEL. Esta ferramenta foi selecionada por disponibilizar todos os algoritmos necessários para realizar os procedimentos previstos neste estudo de caso.

Nesta Figura 26, é apresentada uma visão da interface empregada pela ferramenta KEEL, onde tem-se instanciada uma execução que contempla o emprego do algoritmo FURIA, e os respectivos da dos de entrada e saída. Uma descrição da ferramenta KEEL está disponível no Apêndice C.

8.2.2 Discussão dos Resultados

Os resultados foram analisados a partir das métricas de acurácia, F1 Score, quantidade de regras geradas e tempo de execução para o processamento da classificação para cada algoritmo.

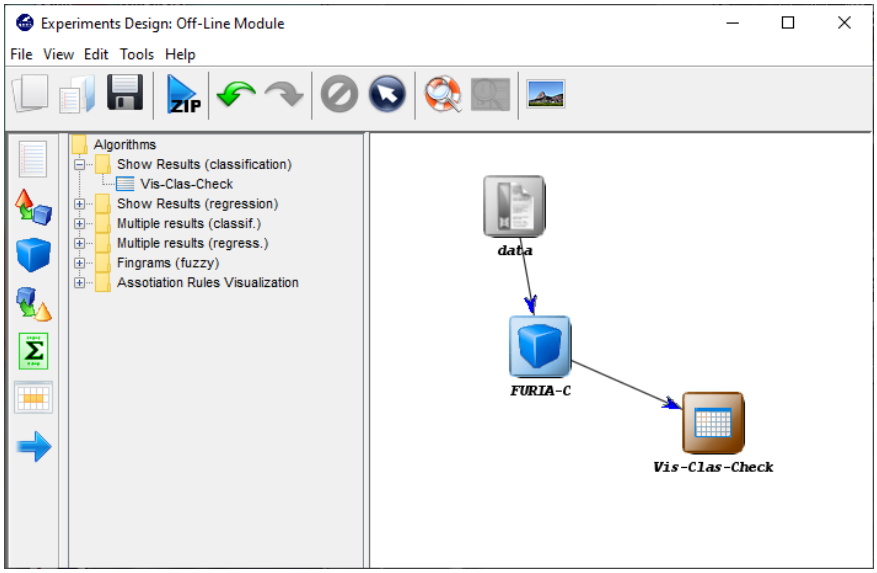


Figura 26 – Interface da Ferramenta KEEL para Experimento com Algoritmo FURIA

Acurácia dos Algoritmos de Classificação

Nos gráficos da Figura 27 estão representados os valores de acurácia geral, para *VoD* e para *Live* com algoritmos fuzzy. O algoritmo FURIA possui o melhor desempenho em todos os *Datasets* avaliados. Pode-se observar que a acurácia obtida na classificação de fluxos em *VoD* e *Live* são melhores do que a acurácia geral. Este desempenho deve-se ao ajuste de atributos realizados previamente que tiveram como foco a classificação destes tipos de protocolos. Porém, o desempenho geral ficou próximo dos 93% com o uso do algoritmo FURIA.

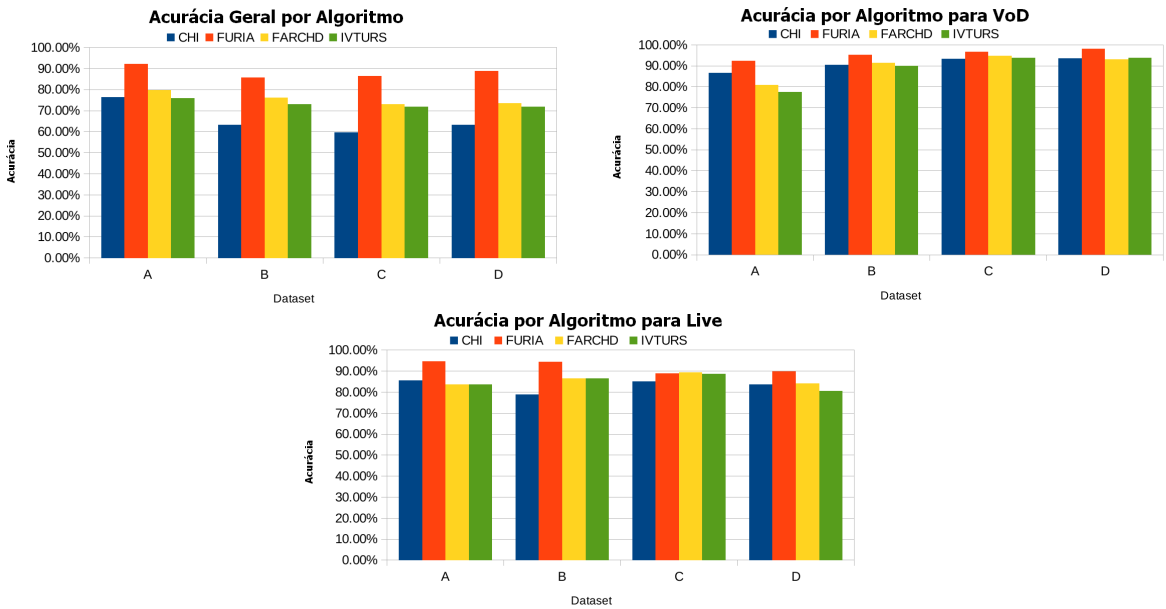


Figura 27 – Acurácia por Algoritmo em Abordagem Fuzzy

Nos gráficos da Figura 28 estão representados os valores de acurácia geral, para

VoD e para *Live* com algoritmos em aprendizagem de máquina. O algoritmo C.45 possui o melhor desempenho em todos os *Datasets* avaliados, com o SVM com resultados bem próximos. Pode-se observar que a acurácia para classificação de fluxos em *VoD* e *Live* possuem melhor acurácia do que a acurácia geral, resultado similar a abordagem em lógica fuzzy. A acurácia média dos algoritmos C.45 e SVM ficaram acima dos 94%, chegando a próximo de 100% para o algoritmo SVM no *Dataset* D.

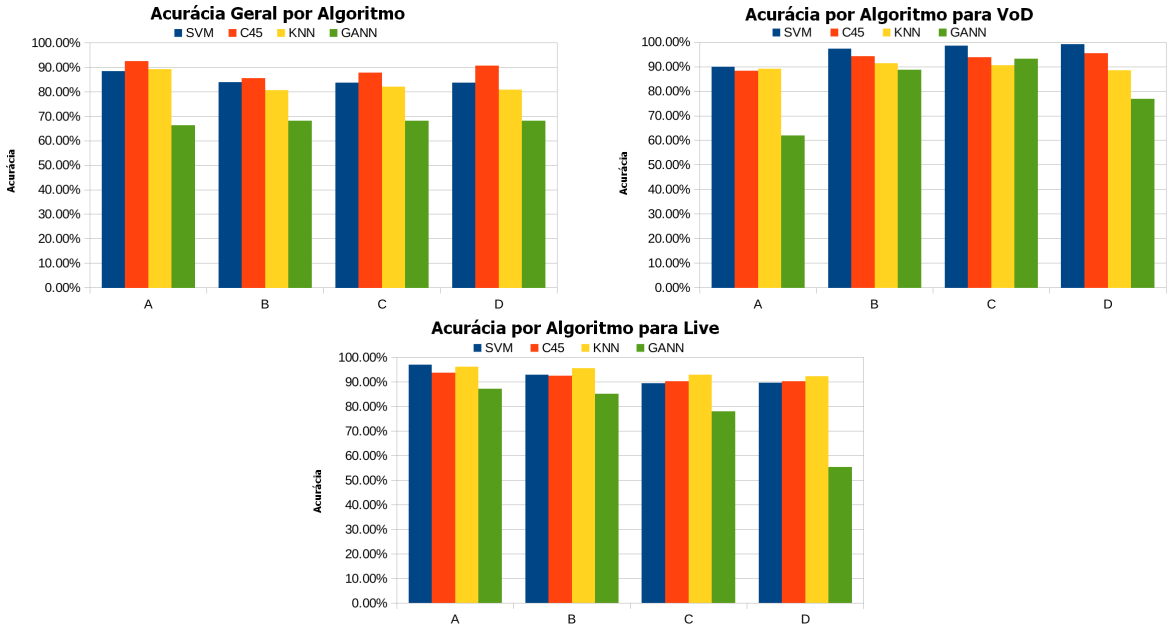


Figura 28 – Acurácia por Algoritmo em Aprendizagem de Máquina

A métrica F1 Score é a média harmônica entre precisão (*precision*) e revocação (*recall*) (MIRANDA RIOS et al., 2021). Como essa medida é uma média, ela fornece uma estimativa mais precisa da eficiência do classificador do que meramente precisão ou *recall*. Portanto, essas métricas são usadas para avaliar o desempenho das abordagens propostas para classificar os fluxos dos protocolos de rede considerados nesta Tese.

Nos gráficos da Figura 29 estão representados os valores de F1 Score, para *VoD* e *Live* nos 8 algoritmos analisados. Na Figuras 29(a) e 29(c) estão os resultados do F1 Score para os algoritmos em lógica fuzzy. O algoritmo FURIA obteve melhor desempenho, principalmente, na classificação dos fluxos em *VoD*. Os fluxos do tipo *Live* obtiveram excelente F1 Score com o algoritmo FURIA.

Nas Figuras 29(b) e 29(d) são apresentados os resultados do F1 Score para os algoritmos em aprendizagem de máquina, onde os algoritmos SVM, C45 e KNN obtiveram resultados consistentes tanto para fluxos em *VoD* quanto para fluxos em formato *Live*.

Quantidade de Regras Geradas

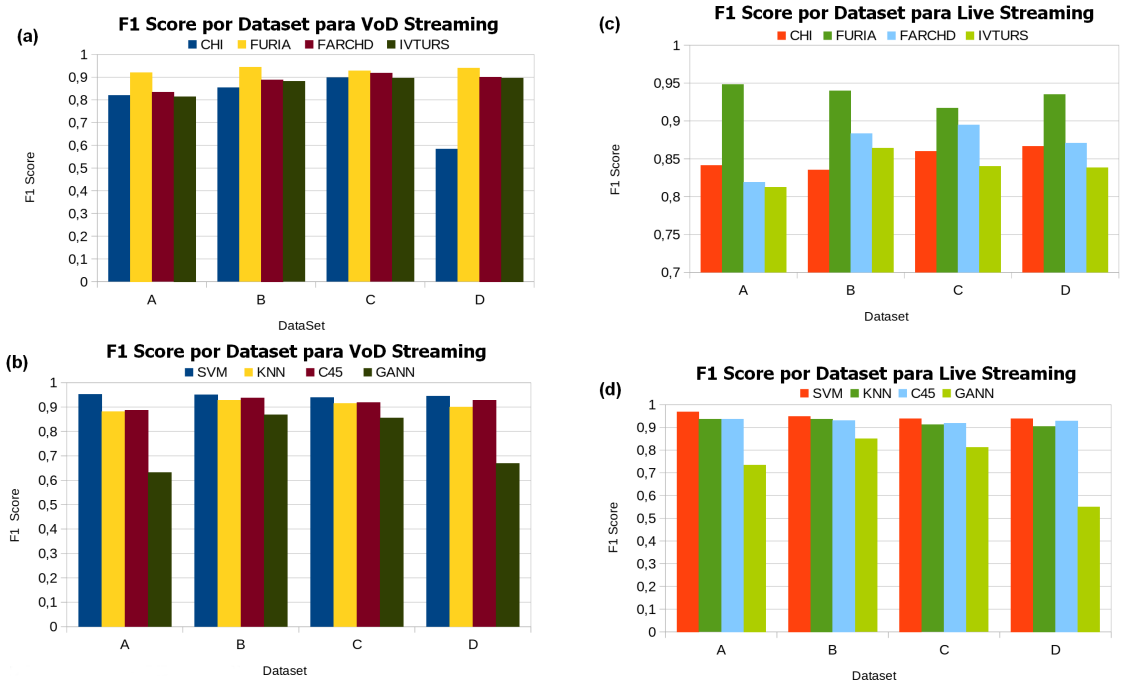


Figura 29 – F1 Score, Algoritmos Fuzzy para VoD (a) e Live (c), Algoritmos em Aprendizagem de Máquina para VoD (b) e Live (d)

Nos gráficos das Figura 30 estão comparadas as quantidades de regras para classificação de todos os tipos de protocolos e para os tipos *VoD* e *Live* em separado. O algoritmo FURIA possui a maior quantidade de regras para as classificações de fluxos de *VoD* e *Live*. Este incremento na quantidade de regras resulta na melhoria de acurácia na classificação. Para a classificação geral dos *Datasets* o algoritmo CHI apresenta o maior número de regras, mas não na melhor acurácia, que pertence ao algortimo FURIA. O processo de escolha das regras obedece um critério de confiança ou peso, onde as regras que possuem maior valor destes quesitos apresentam melhor acurácia. As regras contabilizadas para as classificações de *VoD* e *Live* foram selecionadas com confiança maior do que 0,8, valor baseado de acordo com a qualificação da saída na ferramenta KEEL.

Tempo de Execução por Algoritmo de Classificação

Os tempos das execuções de cada algoritmo são apresentados na Tabela 22. Considerando o método mais rápido, com melhor acurácia, destacam-se os algoritmos FURIA, SVM e C45. O algoritmo FURIA gera resultados rápidos e com as melhores acurácias nos testes realizados. Os algoritmos SVM e C45 apresentaram ótimas acurácias com tempos de execuções muito próximos um do outro.

Por outro lado, as execuções mais lentas foram obtidas pelos algoritmos IVTURS e GANN. O algoritmo IVTURS obteve o maior tempo de execução, muito acima dos

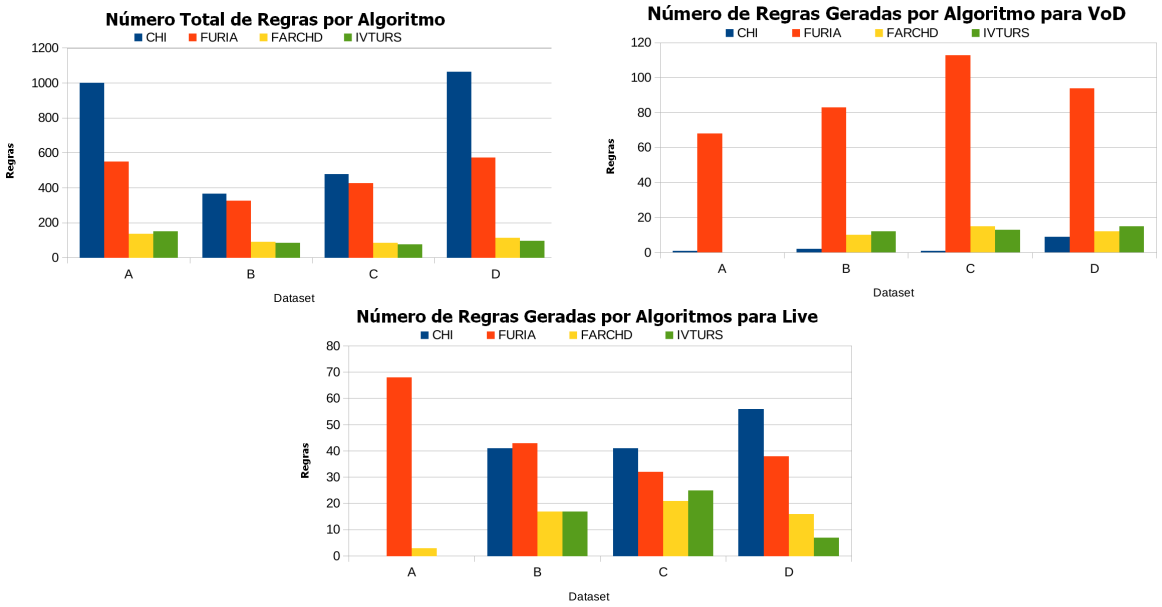


Figura 30 – Quantidade de Regras Geradas por Algoritmo Geral, para VoD e para Live

demais. O motivo pode estar relacionado a ser o único algoritmo baseado em lógica fuzzy tipo-2 intervalar, a qual gera maior complexidade na execução. O GANN explora algoritmos genéticos e obteve o segundo maior tempo para gerar a classificação. Entretanto, o tempo de execução maior não resultou em uma maior acurácia por parte dos algoritmos mais lentos.

Como esperado, percebe-se que o tamanho do *Dataset* influencia no tempo consumido na execução. O *Dataset* D possui a maior quantidade de dados e provocou os tempos maiores de execução.

Tabela 22 – Tempo de Execução por Algoritmo(s)

Dataset	**CHI**	**FURIA**	**FARCHD**	**IVTURS**	**SVM**	**C45**	**KNN**	**GANN**
A	13	100	17	68700	17	24	28	10140
B	8	35	10	23460	10	14	12	4140
C	8	36	12	41640	12	20	13	4140
D	15	48	17	111600	17	23	11	6060

Matrizes de Confusão das Classificações Realizadas por *Dataset*

As matrizes de confusão para cada um dos *Datasets* utilizados são apresentadas nas Tabelas 23, 24, 25 e 26. Os erros de classificação ocorridos para os protocolos *VoD* e *Live* foram relacionados a inversão de *VoD* para *Live* ou de *Live* para *VoD*. Em outros casos aconteceu a classificação de um *streaming* de vídeo, *VoD* ou *Live*, para o protocolo QUIC ou HTTPS. Os protocolos QUIC e HTTPS são os protocolos utilizados pelos serviços de *streaming* de vídeo. Portanto, os erros de classificação ocorridos são aceitáveis e dentro do esperado.

Os protocolos HTTP e HTTPS foram os que obtiveram maior número de erros de classificação, excluindo a categoria *Outros* onde isto já era esperado por conter diversos outros protocolos. Isto é devido a grande variedade de serviços que utilizam como base os protocolos HTTP e HTTPS e aumentam consideravelmente as variações dos atributos dos fluxos. Tais incertezas relacionadas as variações dos atributos podem ser tratadas por meio do uso de abordagens em lógica fuzzy valorada intervarlamente.

Tabela 23 – Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* A

Chi-RW							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**201**	30	0	1	0	0	0
Live	56	**364**	4	1	0	0	0
https	0	21	336	57	70	62	16
quic	1	20	24	413	9	64	55
http	0	2	30	10	515	29	8
dns	0	0	19	4	27	546	2
outros	0	3	155	4	5	6	419

FURIA							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**215**	17	0	0	0	0	0
Live	20	**403**	1	1	0	0	0
https	0	3	500	44	34	5	14
quic	0	1	33	557	4	2	3
http	0	0	40	2	552	3	3
dns	0	0	6	2	0	588	4
outros	0	1	22	9	7	1	560

FARCHD							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**209**	22	0	1	0	0	0
Live	57	**356**	10	2	0	0	0
https	1	39	278	72	86	12	112
quic	2	21	39	439	18	2	79
http	0	2	38	9	523	0	28
dns	0	1	7	6	5	579	2
outros	0	3	29	11	13	9	535

IVTURS							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**207**	23	0	2	0	0	0
Live	59	**356**	7	3	0	0	0
https	2	41	240	89	75	39	114
quic	7	22	36	426	8	7	94
http	1	3	45	16	487	14	34
dns	0	0	11	6	24	555	4
outros	0	6	33	31	8	14	508

Tabela 24 – Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* B

Chi-RW							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**210**	22	0	0	0	0	0
Live	49	**183**	0	0	0	0	0
https	0	1	117	22	20	6	16
quic	1	0	23	92	2	67	1
http	0	0	8	0	84	5	17
dns	0	0	1	0	1	228	1
outros	0	0	27	0	19	1	116

FURIA							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**221**	10	0	1	0	0	0
Live	12	**219**	0	1	0	0	0
https	1	4	136	15	50	4	22
quic	1	1	5	221	0	1	3
http	1	0	15	0	203	1	12
dns	0	0	1	0	0	228	3
outros	0	0	25	10	29	2	166

FARCHD							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**212**	19	0	1	0	0	0
Live	31	**201**	0	0	0	0	0
https	1	2	120	25	50	1	33
quic	1	0	22	162	22	0	25
http	0	0	7	7	193	0	25
dns	0	0	2	1	2	225	2
outros	0	1	20	32	51	1	127

IVTURS							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**201**	31	0	0	0	0	0
Live	21	**209**	0	2	0	0	0
https	0	3	131	24	45	0	29
quic	0	0	33	138	39	7	15
http	0	3	15	2	161	1	50
dns	0	0	2	4	4	218	4
outros	2	5	34	20	42	0	129

Tabela 25 – Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* C

Chi-RW							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**337**	23	0	0	0	0	0
Live	54	**307**	0	0	0	0	0
https	0	11	46	173	25	43	14
quic	0	3	12	182	5	107	32
http	0	4	13	63	146	29	28
dns	0	0	0	0	13	344	3
outros	0	5	9	73	37	12	149

FURIA							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**349**	12	0	0	0	0	0
Live	40	**321**	0	0	0	0	0
https	0	3	238	28	47	5	40
quic	1	0	9	339	4	3	5
http	1	1	18	5	319	3	14
dns	0	0	1	1	0	357	2
outros	0	2	24	17	50	4	264

FARCHD							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**342**	19	0	0	0	0	0
Live	38	**323**	0	0	0	0	0
https	0	11	62	68	39	3	178
quic	3	1	12	281	14	2	48
http	1	5	13	33	248	0	61
dns	0	0	4	1	6	348	2
outros	0	2	13	56	44	4	242

IVTURS							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**339**	22	0	0	0	0	0
Live	41	**320**	0	0	0	0	0
https	7	26	55	59	47	11	156
quic	2	2	12	309	14	3	19
http	4	15	9	16	211	2	104
dns	0	1	1	2	14	342	1
outros	3	15	7	31	59	3	243

Tabela 26 – Matrizes de Confusão Obtidas por Meio de Múltiplos Classificadores para os Fluxos Contidos no *Dataset* D

Chi-RW							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**338**	23	0	0	0	0	0
Live	59	**302**	0	0	0	0	0
https	0	2	409	313	50	74	7
quic	1	3	95	535	8	261	93
http	0	1	23	107	695	45	19
dns	0	0	4	0	7	982	3
outros	0	5	58	139	20	65	362

FURIA							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**354**	5	0	0	2	0	0
Live	36	**325**	0	0	0	0	0
https	1	1	754	69	101	14	60
quic	1	1	26	964	3	1	4
http	0	1	70	10	872	1	46
dns	0	0	1	3	0	996	0
outros	0	1	101	23	51	5	819

FARCHD							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**336**	24	0	1	0	0	0
Live	44	**304**	0	4	9	0	0
https	0	2	416	219	110	10	243
quic	1	2	96	750	89	10	52
http	3	2	62	59	712	1	161
dns	0	0	6	7	6	980	1
outros	1	3	87	137	47	8	717

IVTURS							
	VoD	**Live**	**https**	**quic**	**http**	**dns**	**outros**
VoD	**339**	16	0	1	5	0	0
Live	56	**291**	0	0	14	0	0
https	0	3	732	150	88	18	9
quic	0	3	81	873	33	8	2
http	0	13	286	19	640	2	40
dns	0	0	23	6	9	962	0
outros	0	7	540	83	70	27	273

8.3 Estudo de Caso 3: Classificação com Abordagem Híbrida

Esta seção trata de aspectos referentes a concepção, a prototipação e as avaliações referentes ao terceiro estudo de caso, onde foi contemplada a classificação híbrida concebida para a abordagem FuzzyNetClass.

Para isto, foi também foi empregada a lógica fuzzy valorada intervalarmente para o procedimento de classificação, de forma integrada com algoritmos de aprendizagem de máquina para seleção de atributos, e ainda para definição de regras e funções de pertinência das variáveis, ambas tratadas no processo de classificação.

8.3.1 Descrição do Estudo de Caso

Neste estudo de caso, tal como no primeiro, também foi empregada da plataforma Juzzy versão 2. O código da linguagem Java na qual é implementada a Juzzy, foi executado em um Linux Ubuntu 20.04, com o Java *runtime* na versão 11.0.17, usando um processador Intel i9, com 20 núcleos de processamento e 32 GB de memória RAM.

No que diz respeito a extração de atributos empregando os algoritmos de seleção descritos na Seção 7.3, com a validação por especialistas da CREI/UFPEL, foram selecionados um total de 11 atributos. Este número maior de atributos que o empregado no estudo de caso 1, teve por objetivo buscar a qualificação das métricas de acurácia e entropia, contribuindo assim para uma maior confiabilidade dos resultados a serem obtidos.

Para dar suporte a classificação híbrida prevista para a FuzzyNetClass foi explorado o algoritmo IVTURS, disponibilizado também na Ferramenta KEEL, a partir do qual foi possível analisar os limites a serem empregados nas funções de pertinência do sistema fuzzy valorado intervalarmente utilizado. Desta análise, foram ajustados os limites das funções de pertinência superiores e inferiores para sistema fuzzy valorado intervalarmente.

Por sua vez, na etapa de Inferência da FuzzyNetClass foi considerada uma análise baseada no algoritmo FURIA, da qual resulta um conjunto de regras e o respectivo grau de confiança de cada um delas. A motivação para emprego do FURIA neste caso, foi consequência dos testes realizados durante o desenvolvimento desta Tese, a partir dos quais foi possível identificar que o algoritmo FURIA obteve um desempenho superior aos demais algoritmos testados. Os especialistas da CREI/UFPEL, tendo por base as indicações feitas pelo FURIA quando ao grau de confiança de cada regra, realizaram a seleção da regras a serem empregadas.

Na defuzzificação de um sistema fuzzy valorado intervalarmente consideram-se duas principais etapas: (i) Defuzzificador, (ii) Redutor de Tipo-1. Nesse estudo de caso, coerente com as escolhas com o primeiro estudo de caso, foi considerada a plataforma Juzzy para a prototipação da FuzzyNetClass, para a função de defuzzifi-

cador foi utilizada a metodologia centroide, por sua vez como redutor de tipo-1 foram empregadas as duas metodologias centroide e centro dos conjuntos, sendo inclusive feitas comparações neste sentido.

Na etapa de extração de dados foram utilizadas saídas com valores *crisp* pontuais, graus de pertinência obtidos dos conjuntos fuzzy valorados intervalarmente e termos linguísticos.

Os algoritmos de classificação considerados para comparações geram como saída um termo textual para cada fluxo de rede analisado, o qual é comparado com o termo textual do *Dataset* de entrada, e usado para calcular a acurácia.

O sistema de inferência fuzzy considerou funções de pertinência do tipo trapezoidal. O processo de inferência utilizado foi com base no método de Mamdani (MAMDANI, 1976), considerando uma base de regras com conectivos lógicos do tipo “AND” aplicando normas triangulares.

Para a obtenção dos termos textuais *LowVideo*, *AverageVideo* e *HighVideo* associados a variável de saída *Vídeo* da FuzzyNetClass, os valores foram agregados empregando a média aritmética dos graus de pertinência inferiores e superiores de cada conjunto fuzzy valorado intervalarmente projetado na saída.

Com base nas etapas de fuzzificação, aplicou-se a média aritmética do valor *crisp* obtido para variável de saída *Vídeo*, quando da execução da etapa de redução de tipo. Após agregação dos IVFS considerados, identifica-se o IVFS que atingiu maior média.

Neste sentido, a partir da agregação dos graus de pertinência do IVFS, tem-se sempre as seguintes possibilidades de termos linguísticos (TL) para a variável de saída *Vídeo*:

(i) Termo *LowVideo*, caso o IVFS modelando a TL *Low* para variável *Vídeo* apresenta o maior valor das agregações;

(ii) Termo *AverageVideo*, caso a média aritmética obtida do IVFS associado ao TL *Average* seja maior que as médias dos IVFS para TL*Low* e TL*High*; e ainda

(iii) Termo *HighVideo* caso a agregação do IVFS para TL *High* seja maior que as agregações correspondentes dos IVFS com TL*Low* e TL *Average*.

Desta forma foram organizadas as avaliações da acurácia da variável de saída no FuzzyNetClass, as quais são comparadas com os algoritmos de aprendizagem de máquina selecionados na validação.

Todos os *Datasets* considerados neste estudo de caso incluem fluxos de *streaming* de vídeo *VoD* e *Live* de forma separada. Neste estudo de caso, todos estes fluxos foram intencionalmente categorizados como *Vídeo*.

Por outro lado, devido a similaridade dos tipos de fluxo *VoD* e *Live* foram considerados o termo linguístico *HighVideo* para análise dos resultados de acurácia, comparação entre redutores e na análise de entropia intervalar.

Base de Dados e Definição das Funções de Pertinência

Para definir as configurações dos pontos limites das funções de pertinência foi utilizado o algoritmo de classificação IVTURS junto a ferramenta KEEL. O IVTURS oferece uma abordagem que utiliza a IVFL e fornece como saída além da acurácia, configurações dos pontos limites (*lowers*, *uppers* e *peaks*) para funções de pertinências triangulares associadas às variáveis consideradas no processo de classificação. Em sua totalidade os pontos de *lowers*, *uppers* e *peaks* gerados com IVTURS foram iguais para todas as variáveis de entrada.

O IVTURS utiliza como método de aprendizagem de máquina o FARC-HD (ALCALÁ-FDEZ; ALCALÁ; HERRERA, 2011) para geração e *tuning* das funções de pertinência. A etapa de aprendizado é realizada considerando funções de pertinência triangulares obtidas através de um particionamento linear do domínio de cada variável de entrada.

Também é utilizado no IVTURS, na perspectiva da aprendizagem de máquina, um algoritmo genético com dois objetivos: (i) ajustar os valores de parâmetros usados na construção das funções de equivalência restrita valoradas intervalarmente com intuito de melhorar o método de raciocínio fuzzy, e (ii) realizar o processo de seleção de regras obtendo um conjunto de regras compacto e cooperativo. O processo de redução de regras usa uma abordagem considerando algoritmo genético de codificação simples com base no modelo CHC (ESHELMAN, 1990).

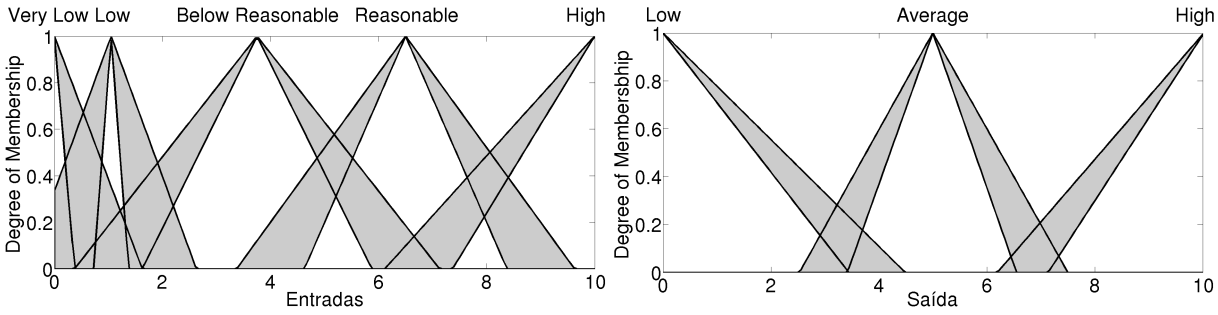
A partir de testes preliminares para análise da acurácia, realizando variações nas quantidades dos termos linguísticos, neste estudo de caso que foca na classificação híbrida da FuzzyNetClass, foram incrementados de 03 para 05 os termos linguísticos de todas as variáveis de entrada.

Na Figura 31(a), tem-se o modelo gerado para as variáveis de entrada, considerando os seguintes termos linguísticos: *Very Low*, *Low*, *Below Reasonable*, *Reasonable* e *High*.

E ainda, Figura 31(b), tem-se exposto o modelo considerado para a variável de saída, neste caso a variável de saída também foi modelada considerando três conjuntos fuzzy valorados intervalarmente (*Low, Average e High*) empregando funções de pertinência triangulares.

Na Tabela 27 são apresentados os pontos de configuração das funções de pertinência de todas as variáveis de entrada obtidas por meio das execuções do algoritmo IVTURS. Como o algoritmo IVTURS gera como saída uma classificação binária, foi necessário realizar a modelagem da variável de saída para FuzzyNetClass a partir de testes realizados e com a visão de um profissional especialista no gerenciamento de redes de computadores.

Os parâmetros que indicam os limites, que estão abaixo de zero ou acima de 10,



(a) Entradas (b) Saída

Figura 31 – Modelo das Variáveis de Entrada e Saída

foram considerados para prover uma descrição gráfica das funções de pertinência (completam os pontos que definem os triângulos inferiores e superiores) dos conjuntos fuzzy valorados intervalarmente, para as entradas e saídas, na modelagem considerada. Entretanto, como mostram as áreas escuras nos triângulos nas Figura 31(a) e 31(b), as imagens geradas no processo de fuzzificação são referentes apenas ao intervalo de domínio da aplicação, no caso $[0, 10]$.

Tabela 27 – Configuração das Variáveis

Tipo de Variáveis	**Termos Linguísticos**	**Limites**					
		Superiores			**Inferiores**		
		início	**pico**	**final**	**início**	**pico**	**final**
Entrada	Very Low	-1.5	0	1.5	-0.5	0	0.5
	Low	0	1	2.5	0.8	1	1.8
	Below Reasonable	0.5	3.8	7.2	1.8	3.8	7.2
	Reasonable	3.4	6.8	9.8	4.5	6.8	8.5
	High	6.2	10	12.2	7.2	10	11.2
Saída	*Low*	-2	0	4.5	-1	0	3.5
	Average	2.5	5	8.5	3.5	5	7.5
	High	4.5	10	12	5.5	10	11

De mesma forma como exposto anteriormente na Seção 8.1, neste estudo de caso os valores obtidos das variáveis consideradas foram normalizados para o intervalo de $[0, 10]$. Esta etapa de definições das configurações das funções de pertinência integra-se ao processo de fuzzificação, onde ocorre o mapeamento de subconjuntos de números reais, em geral discretizados, para o domínio fuzzy.

A fuzzificação indica que há atribuições de valores linguísticos, descrições vagas ou qualitativas, definidas por funções de pertinência associadas às variáveis de entrada. E ainda, pode indicar uma espécie de *pré-processamento* de categorias ou classes dos sinais de entrada, reduzindo o número de valores a serem processados. Sendo assim, uma menor quantidade de valores processados significa menos complexidade das computações.

Base de Regras

As regras que compõem o sistema de inferência fuzzy podem ser obtidas por especialistas em forma de sentenças linguísticas, e se constituem em um aspecto fundamental no desempenho de um sistema de inferência fuzzy.

Neste estudo de caso foram consideradas configurações obtidas através de execuções do algoritmo FURIA. A partir da execução do mesmo foram consideradas 140 regras distintas, com a ajuda de especialistas, as quais foram integradas a FuzzyNetClass. Na Tabela 28 estão listadas as quantidades de regras para cada termo linguístico de saída *LowVideo*, *AverageVideo* e *HighVideo*. A quantidade de regras para o termo *HighVideo* foi mais predominante devido aos experimentos para validação de resultados, onde foram ajustados os parâmetros para redução de falsos positivos.

Tabela 28 – Número de Regras Geradas pelo Algoritmo FURIA para Cada Termo Linguístico de Saída

Termo	**Qte de Regras**
LowVideo	11
AverageVideo	15
HighVideo	114

O processo de obtenção das regras resultantes do algoritmo FURIA foi realizado por meio de *scripts* desenvolvidos em Python e Shell Script. Para realizar a filtragem e a pré-formatação dos arquivos de saída gerados pela ferramenta KEEL quando da execução do algoritmo FURIA foram utilizados os utilitários sed[5], awk[6] e grep[7] em um Shell Script executado em ambiente Linux.

O algoritmo FURIA possibilita aprendizagem de regras fuzzy e estende o algoritmo RIPPER (COHEN, 1995a), sucessor do algoritmo IREP (FÜRNKRANZ; WIDMER, 1994) para indução das regras a serem geradas.

Enquanto procedimento interno da sua funcionalidade que promove a otimização das regras que disponibiliza, o algoritmo FURIA, contempla uma etapa interativa a qual busca identificar as regras mais confiáveis, por um procedimento de exclusão (*prunning strategy*). Neste sentido, o algoritmo FURIA produz na saída, para cada uma das regras, um indicador de confiança (CF - *Confidence*), o qual pode auxiliar os especialistas na escolha para das mais oportunas quando da concepção da base de regras.

Um *script* em Python[8] realizou a varredura nos arquivos de saída da ferramenta

[5] https://www.gnu.org/software/sed/
[6] https://www.gnu.org/software/gawk/
[7] https://www.gnu.org/software/grep
[8] https://github.com/emmonks/FuzzyNetClass/blob/main/Python/monta_campos.py

KEEL, extraindo as regras com maior confiabilidade, com grau de confiança superior a 0,9. Na Figura 32 é mostrado um exemplo de regra gerada na execução do algoritmo FURIA, a qual é formada por duas variáveis de entrada e a classe de saída *Live* com grau de confiança 0,96.

```
(NormBwd_Packet_Length_Mean >= 9.438311(-> 9.403099)) and
(NormBwd_Packet_Length_Mean <= 9.517414(-> 9.528182)) =>
Label=Live (CF = 0.96)
```

Figura 32 – Exemplo de Regra Gerada pelo Algoritmo FURIA

A partir da extração por meio do *script*, as regras são convertidas para uso no formato utilizado na FuzzyNetClass e avaliadas por especialistas. Na conversão, os valores escalares das regras originais do FURIA são traduzidos para os termos linguísticos de variáveis de entrada (antecedentes) e de saída (consequentes) que estão listadas na Tabela 27.

8.3.2 Discussão dos Resultados

Nesta seção, consideram-se as avaliações realizadas com a classificação em abordagem híbrida do FuzzyNetClass. Nas avaliações foram aplicados os *Datasets* A, B, C e D descritos no capítulo 7. Os resultados foram analisados por métricas de acurácia e observados os resultados por tipo de redutor, centroide e centro dos conjuntos;

Na Tabela 29, os resultados de acurácia para cada um dos *Datasets* e tipo de redutor são reportados.

Tabela 29 – Tabela Acurácia Resumida - Estudo de Caso 3

Ds	Fluxos	HighVideo		AverageVideo		LowVideo		Falsos		Ac (High)		Ac (High+Average)	
		C	CoS	C	CoS	C	CoS	C	CoS	C	CoS	C	CoS
A	657	655	657	2	0	0	0	145	381	99.70%	100.00%	100.00%	100.00%
B	464	463	464	1	0	0	0	31	69	99.78%	100.00%	100.00%	100.00%
C	722	618	722	104	0	0	0	44	112	85.60%	100.00%	100.00%	100.00%
D	722	618	722	104	0	0	0	168	355	85.60%	100.00%	100.00%	100.00%

• **(Ds)** Dataset • **(Fluxos)** Total de fluxos de vídeo • **(Falsos)** Falsos positivos • **(C)** Centroid • **(CoS)** Centro dos Conjuntos • **(Ac)** Acurácia

Observa-se que o resultado da acurácia é dependente do tipo de redutor e da forma como são agrupados os tempos linguísticos de saída. Por exemplo nos casos dos *Datasets* C e D as acurácias para o redutor centroide apresentaram resultados piores, 85,60% em ambos os casos, em relação ao redutor centro dos conjuntos que obteve 100%.

Entretanto, ao se realizar o agrupamento do termos *HighVideo* e *AverageVideo* os resultados ficam iguais para centroide e centro dos conjuntos. Porém, ao analisar a

quantidade de falso positivos, fluxos classificados como *HighVideo* de forma equivocada, o redutor centro de conjuntos apresentou resultados piores do que o centroide.

Portanto, nos testes realizados a acurácia para cada um dos *Datasets* avaliados resultou em valores acima dos 85,6% para *streaming* de vídeo, mesmo no pior caso que foi com o uso do redutor centroide, sem o agrupamento do termos de saída *HighVideo* e *AverageVideo*.

Os valores médios dos intervalos inferiores e superiores para as saídas com termos linguísticos para cada um dos *Datasets* estão apresentados nos gráficos da Figura 33.

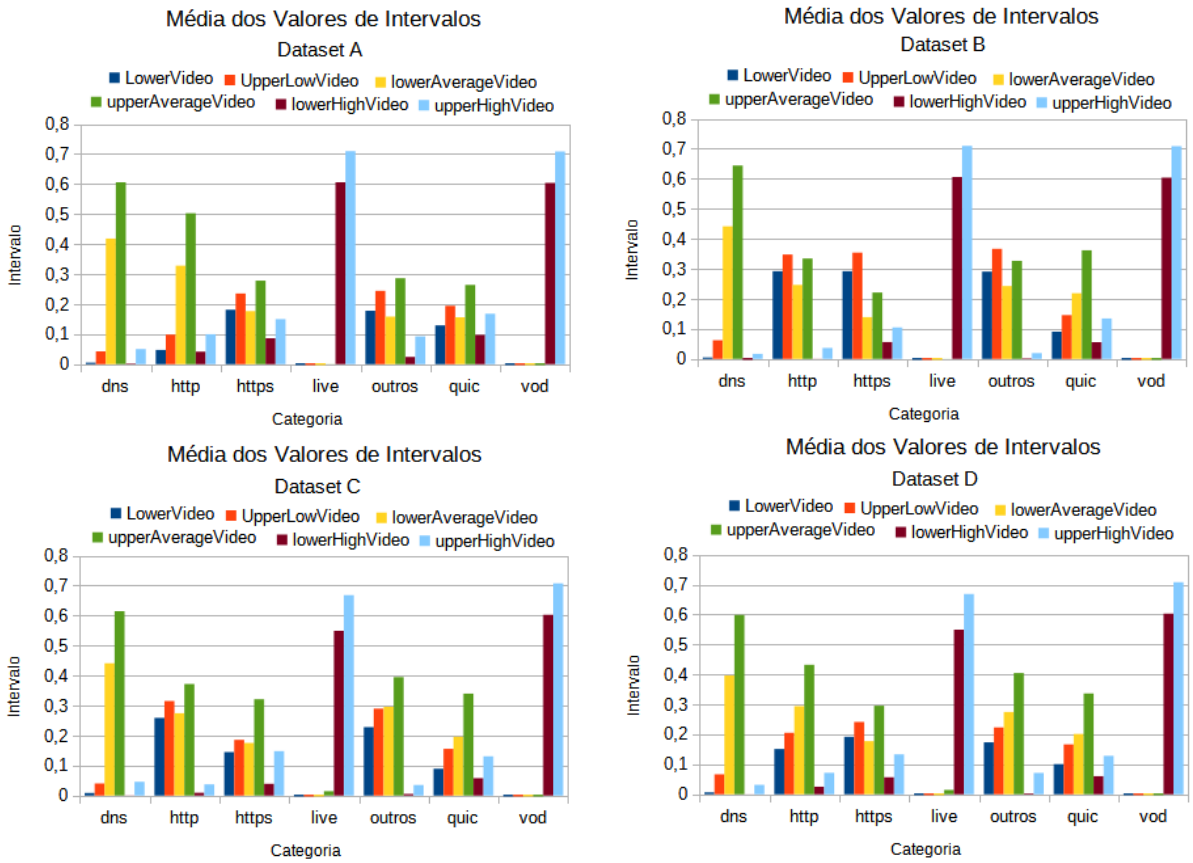


Figura 33 – Valores Médios dos Intervalos Inferiores e Superiores para os Termos Linguísticos

Para esta análise foram separados os fluxos de *streaming* de vídeo em categorias de *VoD* e *Live*. As categorias de *VoD* e *Live* ficaram de forma predominante nos intervalos que correspondem ao termo *HighVideo*. Estes gráficos mostram que tanto a categoria *VoD* quanto a categoria *Live* obtiveram bons resultados para os limites inferiores e superiores da variável linguística *HighVideo*. E, como esperado, o mesmo não ocorreu com outros tipos de fluxo de rede, que este estudo considerou como DNS, HTTP, HTTPS. E ainda, QUIC e outros existentes em todos os *Datasets* avaliados.

Estes valores médios gerados, ou seja, limitantes inferiores e superiores para os intervalos de saída, mostram que os elementos dos demais tipos de fluxos não tem

pertinência significativa no conjunto fuzzy modelado como *HighVideo*, em todos os *Datasets* e para os dois tipos de redutores avaliados. E, como esperado, estes valores para os demais tipos fluxos, exceto dos tipos *VoD* e *Live*, ficaram com predominância nos graus de pertinência para as variáveis linguísticas *LowerVideo*, *UpperLowVideo*, *UpperAverageVideo* e *LowAverageVideo*.

Para dimensionar o tamanho da FOU (*Foot Print of Uncertainty*) foram empregados valores de saída intervalares obtidos a partir dos resultados das execuções da FuzzyNetClass, onde foram analisadas as categorias separadas de *VoD* e *Live*.

Analisando os gráficos da Figura 33 é possível observar que os protocolos DNS, HTTPS e QUIC obtiveram maior FOU quando da classificação da FuzzyNetClass, isso indica que os dados de entrada produziram maiores incertezas e imprecisões quando das avaliações destes protocolos, para serem considerados *Vídeo* na FuzzyNetClass.

O dados produzidos pelos protocolos DNS e HTTP por vezes, atingiriam considerável avaliação para o IVFS *Average*, entretanto apresentaram maior área de incerteza modelada na FOU. Vale salientar ainda que, o protocolo HTTP apesar de ser diferente de *VoD* e *Live* produziram dados que se aproximaram das categorias consideradas para *streaming* de vídeo.

Já observando os dados para os protocolos *VoD* e *Live*, deve-se destacar que ambos além de obterem um grau de pertinência elevado, também alcançaram uma menor FOU. Tal observação reporta a relevância dos atributos elencados pelas etapas de seleção e validação para realizar a classificação na FuzzyNetClass.

A geração de uma menor área na FOU traduz um maior controle para modelagem da incerteza nos dados de entrada e menor imprecisão quando da propagação destas informações incertas durante a execução da FuzzyNetClass.

A abordagem híbrida FuzzNetClass, aplicando algoritmos de aprendizagem de máquina também na etapa de classificação melhora o desempenho e provê uma interpretação dos resultados por diferentes estratégias, o que é relevante para o gerenciamento de informações considerando a modelagem das incertezas e imprecisões na análise deste processo.

Interpretação dos Resultados via Gráficos de Dispersão

Nas Figuras 34, 35, 36 e 37 são apresentados os resultados da classificação de *streaming* de vídeo para os redutores centroide e centro do conjuntos, para os *Datasets* A, B, C e D, respectivamente. Os tipos *VoD* e *Live* foram agrupados com uma única categoria *Vídeo*.

Observa-se dispersão mais acentuada em todos os gráficos na variável linguística *Average*, quando se comparado com as outras classes de fluxo que não são do tipo *Vídeo*. Isso está de acordo com a proposta de intensificar a classificação de apenas

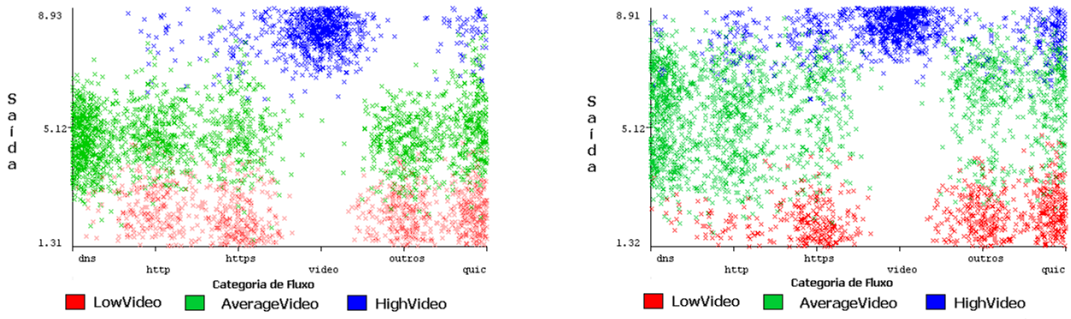


Figura 34 – Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* A

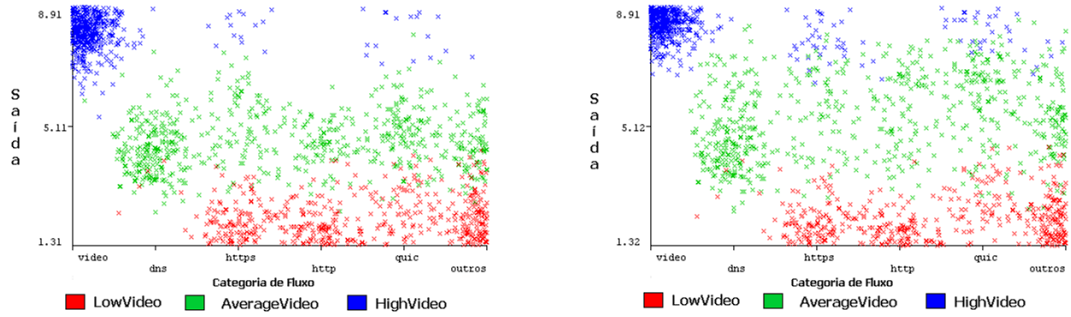


Figura 35 – Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* B

fluxos de vídeo no termo *High*.

Deve-se ressaltar ainda que, para os gráficos referentes a redução via centroide e via centro dos conjuntos obtiveram resultados muito próximos. Entretanto, é possível observar que o redutor centro dos conjuntos apresentou maior número de falsos positivos, representados por pontos azuis em categorias diferentes do tipo *Vídeo*.

Este comportamento teve maior incidência nos testes realizados com os *Datasets* A e D os quais possuem maior quantidade de fluxos de outras categorias que não a de *streaming* de vídeo. O que indica uma maior chance de dispersão com o aumento na quantidade de fluxos de ruído inseridos nos *Datasets*. Em relação aos resultados obtidos no estudo de caso com classificação com abordagem fuzzy, com um número menor de atributos, estes resultados apresentaram menores índices de dispersão para *Vídeo*.

Validação dos Resultados via Entropia Intervalar

Nesta seção é apresentada a interpretação dos resultados também empregando, neste estudo de caso, como métrica a entropia intervalar definida pela Eq.(21), onde $p = 1$ e agregação $\mathbb{M}$ é a extensão intervalar para média aritmética.

Como modelado no primeiro estudo de caso, foi considerado que a saída de uma

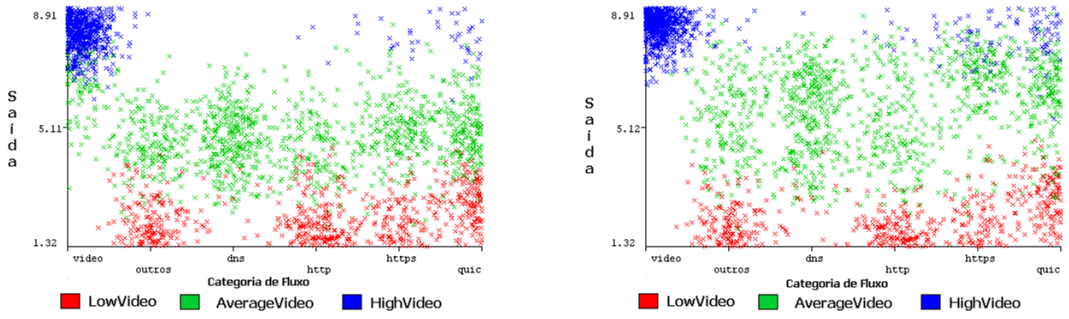


Figura 36 – Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* C

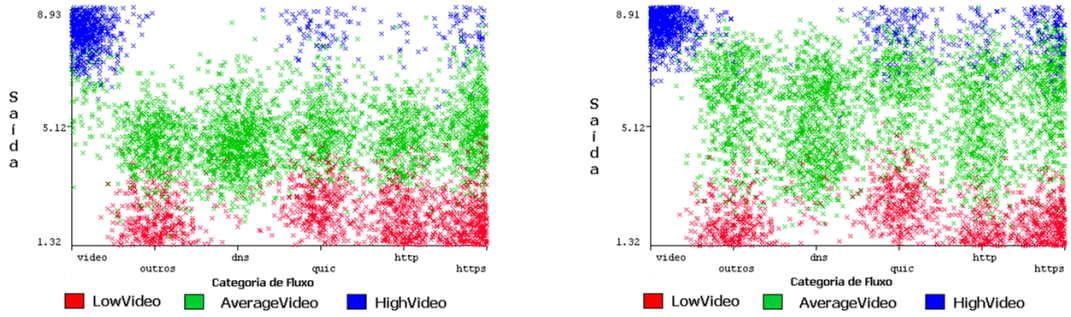


Figura 37 – Análise Comparativa entre Redutores Centroide e Centro dos Conjuntos para o *Dataset* D

função intervalar modela as incertezas inerentes aos atributos de entrada. E, a esta modelagem integra-se a imprecisão, quando da interpretação do diâmetro dos intervalos, refletindo além da falta de conhecimento preciso dos especialistas sobre o grau de pertinência do elemento ao conjunto fuzzy modelado nas entradas, também as imprecisões geradas nas execuções, as quais são propagadas durante suas computações.

Na Tabela 30 são apresentados os resultados obtidos com o emprego da métrica de entropia intervalar para as variáveis de entrada.

Em relação ao primeiro estudo de caso, verifica-se de imediato que a média da entropia intervalar apresenta:

- intervalos cujos extremos apresentam valores reduzidos, mostrando controle das incertezas da variável de saída;
- intervalos com diâmetro reduzido, mostrando controle na propagação das incertezas modeladas nas variáveis de entrada; e ainda,
- não apresentam zeros, o que mostra uma modelagem mais realistas para análise do fluxo de rede focando numa análise para tipo *streaming* de vídeo.

Além disso, tem-se uma expansão dos atributos considerados, mas diâmetros menores para muitas variáveis de entrada, incluindo as indexadas por (9) e (10) (e já

Tabela 30 – Resultados da Métrica de Entropia para as Variáveis de Entrada

Variáveis de Entrada											
DS	**Variável**	**Very Low**		**Low**		**Below Reasonable**		**Reasonable**		**High**	
		Inf	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**
A	(1)	0.157920	0.449610	0.034725	0.308860	0.029413	0.079628	0.007447	0.015418	0.004347	0.006210
	(2)	0.018176	0.047308	0.040522	0.099121	0.081588	0.161510	0.060591	0.103660	0.018612	0.030130
	(3)	0.015186	0.058783	0.020905	0.071396	0.096218	0.163610	0.020391	0.041062	0.025114	0.031512
	(4)	0.005280	0.016440	0.014690	0.051330	0.081099	0.152490	0.055245	0.088269	0.013808	0.019290
	(5)	0.042628	0.114260	0.013118	0.088708	0.041409	0.081291	0.014615	0.026372	0.005076	0.008288
	(6)	0.023421	0.074807	0.011844	0.060690	0.016364	0.043315	0.023886	0.042740	0.022691	0.031502
	(7)	0.051894	0.131790	0.008962	0.087683	0.033915	0.067279	0.013749	0.023578	0.006257	0.009265
	(8)	0.022275	0.063183	0.028533	0.075256	0.017599	0.050802	0.021051	0.032524	0.013561	0.022590
	(9)	0.047577	0.194210	0.045825	0.199410	0.049848	0.119440	0.072013	0.122260	0.071894	0.096299
	(10)	0.024103	0.064531	0.025779	0.069399	0.108010	0.167650	0.035041	0.090740	0.041950	0.054746
	(11)	0.027291	0.065816	0.012674	0.052625	0.009872	0.025627	0.005013	0.008558	0.002506	0.003723
	Variável	**Very Low**		**Low**		**Below Reasonable**		**Reasonable**		**High**	
		Inf	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**
B	(1)	0.118060	0.362050	0.052955	0.299440	0.024978	0.097558	0.004137	0.009045	0.000881	0.001496
	(2)	0.038198	0.084295	0.011263	0.053737	0.094433	0.161800	0.093251	0.152880	0.026651	0.043897
	(3)	0.015699	0.060073	0.017039	0.056375	0.064690	0.106770	0.015830	0.039010	0.029011	0.036028
	(4)	0.003579	0.012773	0.023543	0.076716	0.116020	0.201390	0.036364	0.073705	0.014710	0.019684
	(5)	0.024562	0.080791	0.012271	0.064171	0.024988	0.048560	0.016920	0.028585	0.009169	0.014120
	(6)	0.028510	0.078653	0.011692	0.068447	0.017275	0.039274	0.013231	0.022711	0.011471	0.015232
	(7)	0.047948	0.121160	0.012763	0.076800	0.018085	0.043007	0.010126	0.017596	0.004931	0.007729
	(8)	0.030641	0.078181	0.011668	0.060944	0.013755	0.033589	0.009712	0.016234	0.004317	0.008719
	(9)	0.048345	0.144480	0.075947	0.184720	0.036608	0.102080	0.066516	0.118350	0.084682	0.112970
	(10)	0.029121	0.066632	0.004562	0.044515	0.120001	0.182810	0.061448	0.149880	0.052921	0.070140
	(11)	0.019983	0.04152	0.000662	0.018614	0.005606	0.013516	0.007228	0.011417	0.003887	0.007114
	Variável	**Very Low**		**Low**		**Below Reasonable**		**Reasonable**		**High**	
		Inf	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**
C	(1)	0.115610	0.352200	0.051085	0.277050	0.027600	0.096987	0.015081	0.024385	0.005241	0.008699
	(2)	0.031690	0.069811	0.014278	0.054463	0.085459	0.154340	0.084362	0.136560	0.027355	0.045690
	(3)	0.020706	0.098465	0.022931	0.101390	0.065075	0.117160	0.015924	0.037408	0.021672	0.027052
	(4)	0.004292	0.015121	0.031367	0.090160	0.109760	0.201430	0.036907	0.075491	0.017678	0.023216
	(5)	0.038819	0.112550	0.010542	0.074501	0.021251	0.041872	0.012743	0.024792	0.011695	0.016815
	(6)	0.028874	0.079905	0.009644	0.059042	0.014059	0.031331	0.011705	0.020212	0.009378	0.012855
	(7)	0.053767	0.133290	0.014589	0.083399	0.015107	0.039317	0.010812	0.019570	0.006218	0.009667
	(8)	0.029216	0.072831	0.009136	0.051152	0.013373	0.029910	0.007181	0.011629	0.001834	0.004928
	(9)	0.052721	0.172070	0.048926	0.164470	0.043012	0.101500	0.055264	0.099361	0.065967	0.087358
	(10)	0.026687	0.069142	0.008759	0.061558	0.121910	0.190500	0.056729	0.133710	0.056130	0.071955
	(11)	0.024230	0.050801	0.003115	0.026614	0.005722	0.016172	0.007892	0.011837	0.003199	0.005623
	Variável	**Very Low**		**Low**		**Below Reasonable**		**Reasonable**		**High**	
		Inf	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**	**Inf**	**Sup**
D	(1)	0.149300	0.391590	0.029986	0.252460	0.029198	0.081537	0.011619	0.020307	0.003408	0.005619
	(2)	0.039148	0.085834	0.030559	0.092061	0.059846	0.119770	0.046495	0.081302	0.026330	0.038416
	(3)	0.020341	0.088339	0.033868	0.114200	0.084416	0.158460	0.020708	0.043187	0.017520	0.022333
	(4)	0.003162	0.010894	0.015610	0.052001	0.066315	0.131540	0.043718	0.076313	0.019932	0.025970
	(5)	0.042778	0.118910	0.011250	0.078160	0.028811	0.053978	0.016377	0.030626	0.009830	0.014370
	(6)	0.030150	0.088202	0.012352	0.067650	0.019599	0.045020	0.014130	0.024886	0.010510	0.014391
	(7)	0.035703	0.093261	0.013120	0.065381	0.022861	0.048542	0.015861	0.027583	0.008433	0.012630
	(8)	0.030006	0.080519	0.019853	0.071104	0.016886	0.041610	0.010680	0.018649	0.004891	0.008789
	(9)	0.055166	0.203160	0.053829	0.207440	0.054933	0.126650	0.036348	0.064028	0.032047	0.043093
	(10)	0.033028	0.083510	0.019155	0.088115	0.096535	0.167360	0.043608	0.103580	0.047397	0.061416
	(11)	0.013160	0.029888	0.007163	0.026106	0.007574	0.021099	0.007603	0.011449	0.003027	0.005855

DS (Dataset) (1) FwdPacketLengthMean (2) FwdPacketLengthStd (3) BwdPacketLengthMean (4) BwdPacketLengthStd (5) FlowIATMean (6) FlowIATStd (7) FwdIATMean (8) FwdIATStd (9) PacketLengthMean (10) PacketLengthStd (11) BwdIATMean

consideradas no primeiro estudo de caso), principalmente quando consideradas as variáveis de saída *Average* and *High* .

Para o conjunto de *Datasets* A, tem-se o maior valor (0.194210) da média para entropia intervalar, para cota superior do atributo *FwdPacketLengthMean* modelando a variável linguística *VeryLow*, e o menor valor (0.002506) para cota inferior do atributo *BwdIATMean* ao considerar a variável linguística *High*.

No caso do conjunto *Datasets* B, tem-se o maior valor (0.161800) da média para entropia intervalar, para cota superior do atributo *FwdPacketLengthStd* modelando a variável linguística *BelowReasonable*, e o menor valor (0.00088036) para cota inferior do atributo *FwdPacketLengthMean* ao considerar a variável linguística *High*.

Para o caso do conjunto *Datasets* C, tem-se o maior valor (0.190500) da média para entropia intervalar, para cota superior do atributo *PacketLengthStd* modelando a

Tabela 31 – Resultados da Métrica de Entropia na Saída

Centro dos Conjuntos									
	Low			**Average**			**High**		
Datasets	**Inf**	**Sup**	W	**Inf**	**Sup**	W	**Inf**	**Sup**	W
A	0.12679	0.17175	0.0450	0.24895	0.36386	0.1149	0.16499	0.23431	0.0693
B	0.15592	0.19964	0.0437	0.19309	0.27871	0.0856	0.18874	0.24657	0.0578
C	0.12445	0.16110	0.0367	0.20518	0.30160	0.0964	0.19642	0.26901	0.0726
D	0.13114	0.17994	0.0488	0.25369	0.38113	0.1274	0.11144	0.17623	0.0648
Centroide									
	Low			**Average**			**High**		
Datasets	**Inf**	**Sup**	W	**Inf**	**Sup**	W	**Inf**	**Sup**	W
A	0.15611	0.20669	0.0506	0.12730	0.20007	0.0728	0.14623	0.17969	0.0335
B	0.17242	0.22416	0.0517	0.15687	0.23112	0.0743	0.21178	0.25527	0.0435
C	0.14254	0.18682	0.0443	0.14830	0.22647	0.0782	0.18103	0.22645	0.0454
D	0.15897	0.21227	0.0533	0.17857	0.26513	0.0866	0.09202	0.11750	0.0255

- W Diâmetro

variável linguística *BelowReasonable*, e o menor valor (0.0042919) para cota inferior do atributo *BwdPacketLengthStd* ao considerar a variável linguística *VeryLow*.

E ainda, no caso do conjunto *Datasets* D, tem-se o maior valor (0.149300) da média para entropia intervalar, para cota inferior do atributo *FwdPacketLengthMean* modelando a variável linguística *VeryLow*, e o menor valor (0.0030265) para cota inferior do atributo *BwdIATMean* ao considerar a variável linguística *High*.

E, na Tabela 31 são apresentados os resultados obtidos com o emprego da métrica de entropia intervalar para a variável de saída considerando na validação os redutores centros dos conjuntos (CoS) e centroide (C). E tem-se ainda os resultados do cálculo do diâmetro referentes aos termos linguísticos da variável de saída.

Para os IVFS associados aos termos linguísticos *Average* e *High* o valor agregado para a cota **Sup** da entropia intervalar referente à variável de saída *Vídeo* em ambos os termos linguísticos *Average* e *High* foi menor para avaliação baseada na defuzzificação via C do que via CoS. Também foi menor o diâmetro nestes casos. Tal análise nos mostra que para as VL *Average* e *High* tem-se menor desorganização de informação e menor imprecisão.

Portanto, para este estudo, baseado na abordagem Híbrida do FuzzyNetClass, tem-se que as saídas nas classes *Average* e *High* mostram menor cota superior (e diâmetro) para entropia intervalar, e portanto, maior qualificação para a classificação das informações que incluem fluxos de *streaming* de vídeo *VoD* e *Live*.

8.4 Considerações do Capítulo

Neste Capítulo foram apresentados e discutidos três estudos de caso para avaliação da abordagem FuzzyNetClass. Os mesmos foram concebidos com a premissa de explorar aspectos centrais relacionados a concepção da abordagem FuzzyNetClass. Deste modo os dois tipos de classificadores foram contemplados bem como a sele-

ção dos atributos a serem utilizados. Para tanto, ferramentas, algoritmos e métricas de avaliação são discutidas considerando os comportamentos obtidos ao longo dos esforços realizados para de classificação de *streaming* de vídeo.

No primeiro estudo de caso, denominado classificação com abordagem fuzzy, onde as definições dos parâmetros foram baseadas no suporte de especialistas. Mesmo com um número reduzido de atributos e de regras, a FuzzyNetClass obteve valores de acurácia similares aos resultados disponíveis nos trabalhos relacionados. Tomando como exemplo, destacamos o trabalho de Parfenov et al. (2020), o qual contempla o emprego de 550 regras para realizar classificação de tráfego de rede malicioso e obteve acurácia próxima a 90% para algumas classes de ataques. Neste primeiro estudo de caso, com uma quantidade de 27 regras e 3 atributos, obteve-se acurácia similar.

No segundo estudo de caso foi avaliado o impacto da seleção de atributos na classificação de *streaming* de vídeo. Para esta avaliação, foram realizados testes explorando em algoritmos de aprendizagem de máquina, lógica fuzzy e suporte de especialistas em gerenciamento de redes. Foram analisadas as métricas de acurácia, F1 Score, o número de regras geradas, o tempo de execução e as matrizes de confusão decorrentes. Os resultados obtidos apontaram para uma excelente acurácia e um baixo número de falsos positivos indicados pelo F1 Score.

Os diferentes algoritmos empregados levaram a resultados similares para as diferentes métricas. Particularmente para a classificação de *streaming* em *VoD* e para *Live*, destacaram-se os algoritmos FURIA, SVM e C45 os quais apresentaram resultados um pouco melhores.

Em trabalhos disponíveis na literatura, foram extraídos atributos dos fluxos de rede para uso em classificação do tráfego de rede em geral ou especificamente para tráfego de ataques em rede (FENG et al., 2020), (DUCANGE et al., 2017). Nestes trabalhos, foram obtidas diferentes acurácias, sendo de 92% a máxima atingida para as classes de protocolos com maior acerto, o que é bastante próximo dos resultados alcançados pala FuzzyNetClass neste estudo de caso.

No terceiro estudo de caso, denominado classificação com abordagem híbrida, foram realizadas avaliações da abordagem FuzzyNetClass, envolvendo as etapas de fuzzificação, base de regras e inferência geradas com o uso de técnicas em aprendizagem de máquina.

Neste caso, foram selecionados um total de 11 atributos e uma base de regras composta por 140 entradas distintas, o que culminou com resultados superiores aqueles do primeiro estudo de caso. Por sua vez, com relação trabalhos similares encontrados na literatura, por exemplo (BANIHASHEMI; AKTHARKAVAN, 2022), (LABAYEN GUEMBE et al., 2020), a abordagem FuzzyNetClass obteve níveis de acurácia similares e/ou superiores.

Oportuno registrar que devido a diversidade entre os *Datasets* utilizados, bem como os diferentes objetivos de classificação assumidos, o que impacta nos atributos de classificação utilizados, comparações diretas não se mostram oportunas. Entretanto, embora os trabalhos encontrados na literatura não tratem da classificação de *streaming* de vídeo com uso de protocolos atuais, todos tem em comum com a FuzzyNetClass a premissa de classificar fluxos de rede.

9 CONSIDERAÇÕES FINAIS

"Eu quero ver o pôr do sol,
Lindo como ele só".

Lilás
Djavan

Este Capítulo é dedicado a registrar as principais conclusões decorrentes da concepção da abordagem *FuzzyNetClass*, sendo também apresentadas as publicações realizadas ao longo do desenvolvimento desta Tese. Também são caracterizadas alternativas para a continuidade dos trabalhos, tendo por base os esforços de estudo e pesquisa realizados.

9.1 Principais Conclusões

O emprego das redes de computadores nas mais diferentes atividades da sociedade informatizada, vem promovendo uma escala de uso que exige esforços de gerenciamento cada vez mais assertivos. Neste sentido, o mercado vem apontando a classificação do tráfego como uma das demandas estratégicas para administração das redes da atualidade, nas quais as aplicações *online* vem exigindo níveis cada vez maiores de confiabilidade e garantia de padrão de comportamento das diferentes funcionalidades.

Considerando que o tráfego associado ao *streaming* de vídeo já atingia em 2022 cerca de 70% do volume total de dados que trafegava na Internet, sua classificação vem cada vez mais sendo indispensável para as rotinas de trabalho das equipes de gerência. Por sua vez, aliado a este grande volume de dados, os protocolos criptografados introduzem uma complexidade adicional para a classificação do tráfego de redes, fazendo com que os métodos clássicos tenham seu uso cada vez mais comprometido.

Importante ressaltar, que as análises do tráfego de rede realizadas durante a concepção da FuzzyNetClass, caracterizaram o fato de ser empregado na implementação

das funcionalidades associadas ao uso do *streaming* de vídeo, os mesmos protocolos que também são utilizados em outros serviços, tais como a navegação de páginas ou o *download* de arquivos. Esta situação vem a contribuir também, com o aumento da complexidade para classificação de tráfego.

Em decorrência da análise dos diferentes trabalhos relacionados, foi identificada nas propostas uma tendência da adoção de abordagens explorando aprendizagem de máquina, enquanto forma de lidar com os desafios associados ao emprego de criptografia em um cenário que as diferentes aplicações fazem uso de protocolos similares. Por sua vez, em um número significativamente menor, foram encontrados trabalhos empregando lógica fuzzy, entretanto os mesmos empregaram *Datasets* com perfil de tráfego mais conservador, sem explorar o perfil dos protocolos adaptativos, mais atuais, para transmissão do *streaming* de vídeo. Estes trabalhos, também não tinham na sua proposta a exploração de uma abordagem híbrida, cujos resultados atingidos tivessem uma apropriação de sua construção e significado, mais confortável por parte dos seus usuários.

Tendo por base, a discussão dos trabalhos relacionados, acrescida do crescente emprego de criptografia nos fluxos de rede, decorre a motivação central da pesquisa realizada nesta Tese, que promove a associação do reconhecimento de padrões provido pela aprendizagem de máquina, com o tratamento da incerteza inerente aos procedimentos de classificação que utilizem a lógica fuzzy valorada intervalarmente. Assim sendo, a abordagem FuzzyNetClass tem um compromisso com a busca de melhores acurácias, preservando a perspectiva de interpretabilidade que a lógica fuzzy pode propiciar. A combinação destes dois aspectos é central para qualificação do suporte a tomada de decisão por parte das equipes de gerenciamento de redes na atualidade.

Para tanto, foi concebida para a abordagem FuzzNetClass uma arquitetura híbrida para classificação de tráfego de rede, mais precisamente para classificação de tráfego em *streaming* de vídeo. A perspectiva híbrida compreende o emprego de aprendizagem de máquina na seleção de atributos a serem empregados na classificação, bem como na etapa de inferência empregada no procedimento de classificação, pela criação e otimização da Base de Regras utilizada.

Os estudos de caso realizados com a abordagem FuzzNetClass obtiveram resultados consistentes considerando o problema de pesquisa proposto nesta Tese. O primeiro estudo de caso, particularmente, apresentou o comportamento da abordagem FuzzyNetClass empregando uma etapa de classificação não híbrida. Os atributos foram escolhidos com auxílio da aprendizagem de máquina, tendo sido elencados atributos para a Base de Regras os quais foram validados com auxílio de especialistas, resultando em três atributos. Mesmo com este este número de atributos os resultados mostraram-se promissores.

O impacto da seleção de um número maior de atributos pertinentes aos fluxos de

rede e também o emprego de um conjunto maior de *Datasets*, foi o foco considerado no segundo estudo de caso. Ao todo foram selecionados e validados 11 atributos dos fluxos de rede empregados, os quais propiciaram ótimos resultados de acurácia. Os níveis de acurácia obtidos indicaram a importância da seleção de atributos e particularmente no caso da FuzzyNetClass, o significado do emprego de aprendizagem de máquina para auxiliar o especialista nesta atividade.

No terceiro estudo de caso, por sua vez, foram avaliados os resultados da FuzzyNetClass com o emprego da abordagem híbrida para classificação. Foi empregado um número maior de atributos, bem como, um conjunto maior de *Datasets* que no primeiro estudo de caso, fazendo com que os resultados obtidos ficassem sensivelmente melhores. Esta melhora, foi decorrente do emprego de uma Base de Regras substancialmente maior, gerada com auxílio de aprendizagem de máquina a qual indicou o grau de confiança para as diferentes regras, permitindo ao especialista fazer uma escolha das regras a serem efetivamente utilizadas de modo confortável. Os indicadores atingidos, apontaram que que a sinergia entre especialistas e aprendizagem de máquina, caracterizou que a premissa híbrida da abordagem FuzzyNetClass, foi central para a melhoria dos resultados produzidos.

Os resultados dos estudos de casos 1 e 3 foram validados com o uso da métrica Entropia. Estas validações proveram subsídios para medir a confiabilidade e modelar a incerteza, sendo uma ferramenta importante para auxiliar na análise dos resultados obtidos.

Oportuno ressaltar, que os parâmetros empregados para a operação da arquitetura da abordagem FuzzyNetClass foram direcionados para a classificação de *streaming* de video. Entretanto, a arquitetura faculta que sejam ajustados os parâmetros direcionados a outros tipos de tráfego de rede a serem classificados.

Para tanto, estes ajustes paramétricos devem considerar a opinião de um especialista nos diferentes procedimentos da arquitetura, os quais implicam em etapas recursivas, que afetam a especificação dos parâmetros a serem empregados, tanto na seleção de atributos, como na etapa de inferência.

9.2 Publicações Realizadas

A seguir estão relacionadas as publicações ocorridas no decorrer do desenvolvimento dessa Tese. Foram elencadas somente aquelas cuja temática era central para a mesma e assim, de diferentes maneiras, contribuíram para o avanço dos esforços de estudo e pesquisa. Outrossim, estas publicações promoveram a discussão dos principais aspectos atinentes à pesquisa junto à comunidade técnico-científica:

- **MONKS, E. M.**, Moura, B., Schneider, G., Yamin, A., Reiser, R., and Santos, H. (2022). Abordagem Fuzzy Valorada Intervalarmente para Classificação de

Tráfego de Streaming de Vídeo. In CSBC 2022 - SEMISH 2022, UFF Niterói - RJ.

- **MONKS, E. M.**, Moura, B., Schneider, G., Yamin, A., Reiser, R., and Santos, H. (2022). Towards Interval-Valued Fuzzy Approach to Video Streaming Traffic Classification. In 31th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2022.

- **MONKS, E. M. et al. Strategy for Network Flows Attributes Selection to Video Streaming Traffic Classification. Expert Systems with Applications, 2023** (em etapa de revisão).

- FERNANDES, P. ; **MONKS, E. M.** ; DILLI, R. ; YAMIN, A. C. ; REISER, R. . Classificação de Recursos na IoT: Automação de Regras Fuzzy na Seleção de Preferências do Cliente. In: Simpósio Brasileiro de Automação Inteligente - SBAI, 2021, Rio Grande - RS - Brasil. SBAI 2021, 2021.

- SCHNEIDER, G. ; MOURA, B. ; **MONKS, E. M.** ; YAMIN, A. C. ; SANTOS, H. ; REISER, R. . Aplicando Análise Consensual Fuzzy para Tomada de Decisão na Alocacão de Recursos em Nuvem Computacionais. In: IV Workshop-Escola de Informática Teórica, 2021, Bagé - RS - Brasil. VI Workshop-Escola de Informática Teórica 17 a 19 de novembro de 2021, 2021.

- SCHNEIDER, G. ; MOURA, B. ; FREITAS, B. ; **MONKS, E. M**. ; YAMIN, A. C. ; SANTOS, H. ; REISER, R. . Empregando Medidas de Consenso Fuzzy para Gerenciamento de Recursos em Nuvens Computacionais. In: VI Congresso Brasileiro de Sistemas Fuzzy, 2021, São José do Rio Preto. Recentes Avanços em Sistemas Fuzzy, 2021.

- Dilli, R., Fernandes, P., **MONKS, E. M**, Reiser, R., and Yamin, A. (2022). Classificação de recursos na iot: Automação de regras fuzzy na seleção de preferências do cliente. In CSBC 2022 - SBCUP 2022.

- Schneider, G. B., Moura, B., **MONKS, E. M.**, Santos, H. S., Yamin, A. C., and Reiser, R. (2022). Int-flbcc: Exploring fuzzy consensus measures via penalty functions. In Ciucci, D., Couso, I., Medina, J., Slezak, D., Petturiti, D., Bouchon-Meunier, B., and Yager, R. R., editors, Information Processing and Management of Uncertainty in Knowledge-Based Systems - 19th International Conference, IPMU 2022, Milan, Italy, July 11-15, 2022, Proceedings, Part I, volume 1601 of Communications in Computer and Information Science, pages 434–447. Springer.

9.3 Continuidade da Pesquisa

Dentre as diversas alternativas para a continuidade da pesquisa desenvolvida nesta Tese, destacam-se as seguintes frentes de trabalho:

- realizar novos estudos explorando outros *Datasets* na perspectiva de estender o classificador da abordagem híbrida FuzzyNetClass. Esta ampliação irá considerar a classificação de outros tipos significativos de tráfego nas redes de computadores, em especial os diferentes tipos de vídeos;
- estender a modelagem da abordagem híbrida FuzzyNetClass considerando o emprego de Ordens Admissíveis, não somente quando da análise da entropia, mas também no reticulado, onde dados e operadores serão definidos de forma compatível com a ordem total considerada;
- adicionar a FuzzyNetClass uma etapa dinâmica para geração de regras, explorando uma sinergia entre *frameworks* para processamento fuzzy e técnicas de aprendizado de máquina por reforço;
- desenvolver novos estudos de caso, ampliando o escopo de aplicação da abordagem FuzzyNetClass, bem como avaliando as potenciais contribuições das novas frentes de trabalho indicadas para continuidade dos esforços de pesquisa.

As possibilidades indicadas nesta seção para avanços da pesquisa, foram elencadas tendo como um dos critérios centrais o emprego da abordagem FuzzyNetClass como uma ferramenta efetiva por parte de equipes que atuam no gerenciamento do tráfego de redes.

REFERÊNCIAS

ABDULLAH, S. A.; AL-HASHMI, A. S. TiSEFE: Time Series Evolving Fuzzy Engine for Network Traffic Classification. **Int. J. Commun. Netw.**, USA, v.10, n.1, p.116–124, 2018.

AHMAD, I. et al. Machine learning meets communication networks: Current trends and future challenges. **IEEE Access**, USA, v.8, p.223418–223460, 2020.

AL-OBEIDAT, F.; EL-ALFY, E.-S. Hybrid multicriteria fuzzy classification of network traffic patterns, anomalies, and protocols. **Pers Ubiquitous Comput**, USA, v.23, n.5, p.777–791, 2019.

ALCALÁ-FDEZ, J.; ALCALÁ, R.; HERRERA, F. A Fuzzy Association Rule-Based Classification Model for High-Dimensional Problems with Genetic Rule Selection and Lateral Tuning. **IEEE Transactions on Fuzzy Systems**, USA, v.19, n.5, p.857–872, 2011.

ALCALÁ-FDEZ, J. et al. KEEL: a software tool to assess evolutionary algorithms for data mining problems. **Soft Computing**, USA, v.13, n.3, p.307–318, 2009.

ALEXA, A. Disponível em: `<https://www.alexa.com/>`, Alexa Site. Acesso em dezembro de 2022.

ASIAIN, M. J. et al. Negations with respect to admissible orders in the interval-valued fuzzy set theory. **IEEE Transactions on Fuzzy Systems**, USA, v.26, n.2, p.556–568, 2017.

ASIAIN, M. J. et al. Negations With Respect to Admissible Orders in the Interval-Valued Fuzzy Set Theory. **IEEE Trans. Fuzzy Syst.**, USA, v.26, n.2, p.556–568, 2018.

ASMUSS, J.; LAUKS, G. Network traffic classification for anomaly detection fuzzy clustering based approach. In: ICNC-FSKD), 2015., 2015. **Anais...** USA, 2015. p.313–318.

BACZYŃSKI, M. Residual implications revisited. Notes on the Smets–Magrez theorem. **Fuzzy Sets and Systems**, USA, v.145, n.2, p.267–277, 2004.

BACZYŃSKI, M.; JAYARAM, B. **An Introduction to Fuzzy Implications**. Heidelberg: Springer, 2008.

BALASUBRAMANIAM, J. Yagers new class of implications J f and some classical tautologies. **Information Sciences**, USA, v.177, n.3, p.930–946, 2007.

BANIHASHEMI, S. B.; AKTHARKAVAN, E. Encrypted Network Traffic Classification Using Deep Learning Method. In: INTERNATIONAL CONFERENCE ON WEB RESEARCH (ICWR), 2022., 2022. **Anais...** USA, 2022. p.1–8.

BARRENECHEA, E.; BUSTINCE, H.; DE BAETS, B.; LOPEZ-MOLINA, C. Construction of interval-valued fuzzy relations with application to the generation of fuzzy edge images. **IEEE Transactions on fuzzy systems**, USA, v.19, n.5, p.819–830, 2011.

BARRENECHEA, E. et al. Construction of interval-valued fuzzy preference relations from ignorance functions and fuzzy preference relations. Application to decision making. **Knowledge-Based Systems**, USA, v.58, p.33–44, 2014.

BARROS, L. C. de; BASSANEZI, R. C. **Tópicos de lógica Fuzzy e Biomatemática**. Campinas: Unicamp-Imecc, 2006.

BEDREGAL, B. C. On interval fuzzy negations. **Fuzzy Sets and Systems**, USA, v.161, n.17, p.2290–2313, 2010.

BEDREGAL, B. R. C. On interval fuzzy negations. **Fuzzy Sets Syst.**, USA, v.161, n.17, p.2290–2313, 2010.

BEDREGAL, B. R. C.; MEZZOMO, I.; REISER, R. H. S. n-Dimensional Fuzzy Negations. **IEEE Trans. Fuzzy Syst.**, USA, v.26, n.6, p.3660–3672, 2018.

BELIAKOV, G.; PRADERA, A.; CALVO, T. **Aggregation Functions**: A Guide for Practitioners. Heidelberg: Springer, 2009. (Studies in Fuzziness and Soft Computing).

BENTALEB, A. et al. A survey on bitrate adaptation schemes for streaming media over HTTP. **IEEE Commun. Surv. Tutor.**, USA, v.21, n.1, p.562–585, 2018.

BENTKOWSKA, U.; KRÓL, A. Preservation of fuzzy relation properties based on fuzzy conjunctions and disjunctions during aggregation process. **Fuzzy Sets and Systems**, USA, 2015.

BENTKOWSKA, U. et al. Decision making with an interval-valued fuzzy preference relation and admissible orders. **Applied Soft Computing**, USA, v.35, p.792–801, 2015.

BUJLOW, T.; CARELA-ESPAÑOL, V.; BARLET-ROS, P. Independent Comparison of Popular DPI Tools for Traffic Classification. **Computer Networks**, USA, v.76, p.75–89, 2015.

BURILLO, P.; BUSTINCE, H. Construction theorems for intuitionistic fuzzy sets. **Fuzzy Sets and Systems**, USA, v.84, n.3, p.271–281, 1996.

BUSTINCE, H.; BARRENECHEA, E.; MOHEDANO, V. Intuicionistic Fuzzy Implication Operators - An Expression and Main Properties. **Uncertainty, Fuzziness and Knowledge-Based Systems**, USA, v.12, p.387–406, 2004.

BUSTINCE, H.; BURILLO, P.; SORIA, F. Automorphisms, negations and implication operators. **Fuzzy Sets and Systems**, USA, v.134, n.2, p.209–229, 2003.

BUSTINCE, H.; FERNÁNDEZ, J.; KOLESÁROVÁ, A.; MESIAR, R. Generation of linear orders for intervals by means of aggregation functions. **Fuzzy Sets and Systems**, USA, v.220, p.69–77, 2013.

BUSTINCE, H. et al. A New Approach to Interval-Valued Choquet Integrals and the Problem of Ordering in Interval-Valued Fuzzy Set Applications. **IEEE Trans. Fuzzy Syst.**, USA, v.21, n.6, p.1150–1162, 2013.

BUSTINCE, H. et al. A Historical Account of Types of Fuzzy Sets and Their Relationships. **IEEE Transactions on Fuzzy Systems**, USA, v.24, n.1, p.179–194, 2016.

BUSTINCE, H. et al. Similarity between interval-valued fuzzy sets taking into account the width of the intervals and admissible orders. **Fuzzy Sets and Systems**, USA, v.390, p.23 – 47, 2020. Similarity, Orders, Metrics.

CALLADO, A. C. et al. A Survey on Internet Traffic Identification. **IEEE Communications Surveys and Tutorials**, USA, v.11, n.3, p.37–52, 2009.

CALVO, T.; MAYOR, G.; MESIAR, R. (Ed.). **Aggregation Operators**: New Trends and Applications. DEU: Physica-Verlag GmbH, 2002.

CAO, J.; DRABECK, L.; HE, R. Statistical Network Behavior Based Threat Detection. In: IEEE CONFERENCE ON COMPUTER COMMUNICATIONS WORKSHOPS (INFOCOM WKSHPS), 2017. **Anais...** USA, 2017. p.420–425.

CARREON-ORTIZ, H.; VALDEZ, F.; CASTILLO, O. Fuzzy Flower Pollination Algorithm (FFPA): Comparative Study of Type-1 (T1FLS) and Interval Type-2 Fuzzy Logic System (IT2FLS) in Optimization Parameter Adaptation. **Computación y Sistemas**, USA, v.26, n.2, 2022.

CASINO, F.; CHOO, K.-K. R.; PATSAKIS, C. HEDGE: Efficient Traffic Classification of Encrypted and Compressed Packets. **IEEE Transactions on Information Forensics and Security**, USA, p.1–1, 2019.

CASTILLO, O.; MELIN, P. Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review. **Information Sciences**, USA, v.205, p.1–19, 2012.

CASTILLO, O.; MELIN, P. A review on the design and optimization of interval type-2 fuzzy controllers. **Applied Soft Computing**, USA, v.12, n.4, p.1267–1278, 2012.

CASTILLO, O.; MELIN, P.; PEDRYCZ, W. Design of interval type-2 fuzzy models through optimal granularity allocation. **Applied Soft Computing**, USA, v.11, n.8, p.5590–5601, 2011.

CASTRO, J. R.; CASTILLO, O.; MELIN, P. An interval type-2 fuzzy logic toolbox for control applications. In: IEEE INTERNATIONAL FUZZY SYSTEMS CONFERENCE, 2007., 2007. **Anais...** UK, 2007. p.1–6.

CHEN, S.; BARMAN, D. Adaptive weighted fuzzy interpolative reasoning based on representative values and similarity measures of interval type-2 fuzzy sets. **Inf. Sci.**, USA, v.478, p.167–185, 2019.

CHEN, S.; YU, S. Multiattribute decision making based on novel score function and the power operator of interval-valued intuitionistic fuzzy values. **Inf. Sci.**, USA, v.606, p.763–785, 2022.

CHO, K.; MITSUYA, K.; KATO, A. Traffic Data Repository at the WIDE Project. In: FREENIX TRACK: 2000 USENIX ATC, JUNE 18-23, 2000, SAN DIEGO, CA, USA, 2000. **Proceedings...** USENIX, 2000. p.263–270.

CHOI, H. M.; MUN, G. S.; AHN, J. Y. A medical diagnosis based on interval-valued fuzzy sets. **Biomedical Engineering: Applications, Basis and Communications**, USA, v.24, n.04, p.349–354, 2012.

CHROMEDRIVER. Disponível em: <https://chromedriver.chromium.org/>, ChromeDriver Site. Acesso em dezembro de 2022.

CLAFFY, K. C. **Internet Traffic Characterization**. 1994. Tese (Doutorado em Ciência da Computação) — University of California, San Diego, Department of Computer Science.

CLAFFY, K. C.; POLYZOS, G. C.; BRAUN, H.-W. Traffic characteristics of the T1 NSFNET backbone. In: IEEE INFOCOM'93 THE CONFERENCE ON COMPUTER COMMUNICATIONS, PROCEEDINGS, 1993. **Anais...** USA, 1993. p.885–892.

CLAISE, B.; SADASIVAN, G.; VALLURI, V.; DJERNAES, M. RFC 3954: Cisco systems NetFlow services export version 9. **IETF http://www. ietf. org/rfc/rfc3954. txt**, USA, 2004.

COHEN, W. W. Fast Effective Rule Induction. In: MACHINE LEARNING, PROCEEDINGS OF THE TWELFTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, TAHOE CITY, CALIFORNIA, USA, JULY 9-12, 1995, 1995, USA. **Anais...** Morgan Kaufmann, 1995. p.115–123.

COHEN, W. W. Fast effective rule induction. In: **Machine learning proceedings**. USA: Elsevier, 1995. p.115–123.

CORDÓN, O.; JESUS, M. del; HERRERA, F. A proposal on reasoning methods in fuzzy rule-based classification systems. **International Journal of Approximate**, USA, v.20, n.1, p.21–45, 1999.

CORTES, C.; VAPNIK, V. Support vector networks. **Machine Learning**, USA, v.20, p.273–297, 1995.

COSTA, L. et al. Interval Extension of the Generalized Atanassov's Intuitionistic Fuzzy Index using Admissible Orders. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, FUZZ-IEEE 2019, NEW ORLEANS, LA, USA, JUNE 23-26, 2019, 2019., 2019. **Anais...** IEEE, 2019. p.1–6.

COVER, T.; HART, P. Nearest Neighbor Pattern Classification. **IEEE Transactions on Information Theory**, USA, v.13, p.21–27, 1967.

CRUZ ASMUS, T. da et al. Towards interval uncertainty propagation control in bivariate aggregation processes and the introduction of width-limited interval-valued overlap functions. **Fuzzy Sets Syst.**, USA, v.441, p.130–168, 2022.

DAO, N.-N. et al. A Contemporary Survey on Live Video Streaming from a Computation-Driven Perspective. **ACM Computing Surveys (CSUR)**, USA, 2022.

DESCHRIJVER, G. Uninorms which are neither conjunctive nor disjunctive in interval-valued fuzzy set theory. **Inf. Sci.**, USA, v.244, p.48–59, 2013.

DESCHRIJVER, G.; KERRE, E. Implicators based on binary aggregation operators in interval-valued fuzzy set theory. **Fuzzy Sets and Systems**, USA, v.153, n.2, p.229–248, 2005.

DHOTE, Y.; AGRAWAL, S.; DEEN, A. J. A Survey on Feature Selection Techniques for Internet Traffic Classification. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND COMMUNICATION NETWORKS (CICN), 2015., 2015. **Anais...** USA, 2015. p.1375–1380.

DIAS, K. L.; PONGELUPE, M. A.; CAMINHAS, W. M.; ERRICO, L. de. An innovative approach for real-time network traffic classification. **Computer Networks**, USA, v.158, p.143–157, 2019.

DRAPER-GIL, G.; LASHKARI, A. H.; MAMUN, M. S. I.; GHORBANI, A. A. Characterization of Encrypted and VPN Traffic Using Time-Related. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS SECURITY AND PRIVACY (ICISSP), 2., 2016. **Proceedings...** Italy, 2016. p.407–414.

DUBOIS, D.; PRADE, H. **Fundamentals of Fuzzy Sets**. Boston: Kluwer Academic Publishers, 2000.

DUCANGE, P. et al. A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classifiers. In: FUZZ-IEEE, 2017., 2017. **Anais...** Italy, 2017. p.1–6.

D'ALCONZO, A. et al. A survey on big data for network traffic monitoring and analysis. **IEEE Transactions on Network and Service Management**, USA, v.16, n.3, p.800–813, 2019.

D'ALCONZO, A. et al. A Survey on Big Data for Network Traffic Monitoring and Analysis. **IEEE Transactions on Network and Service Management**, USA, v.16, n.3, p.800–813, 2019.

D'ALTERIO, P.; GARIBALDI, J. M.; JOHN, R. I.; WAGNER, C. Juzzy Constrained: Software for Constrained Interval Type-2 Fuzzy Sets and Systems in Java. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (FUZZ-IEEE), 2020., 2020. **Anais...** UK, 2020. p.1–8.

ESHELMAN, L. J. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In: FIRST WORKSHOP ON FOUNDATIONS OF GENETIC ALGORITHMS. BLOOMINGTON CAMPUS, INDIANA, USA, JULY 15-18 1990, 1990. **Proceedings...** Morgan Kaufmann, 1990. p.265–283.

FENG, Y.; KANG, Y.; ZHANG, H.; ZHANG, W. FAFS: A Fuzzy Association Feature Selection Method for Network Malicious Traffic Detection. **KSII Transactions on Internet and Information Systems (TIIS)**, KOR, v.14, n.1, p.240–259, 2020.

FINSTERBUSCH, M. et al. A Survey of Payload-Based Traffic Classification Approaches. **IEEE Communications Surveys & Tutorials**, USA, v.16, n.2, p.1135–1156, 2013.

FODOR, J. C. On fuzzy implication operators. **Fuzzy Sets and Systems**, USA, v.42, n.3, p.293–300, 1991.

FONT, J. M.; HÁJEK, P. On Łukasiewicz's four-valued modal logic. **Studia Logica**, USA, v.70, n.2, p.157–182, 2002.

FÜRNKRANZ, J.; WIDMER, G. Incremental Reduced Error Pruning. In: MACHINE LEARNING, PROCEEDINGS OF THE ELEVENTH INTERNATIONAL CONFERENCE, RUTGERS UNIVERSITY, NEW BRUNSWICK, NJ, USA, JULY 10-13, 1994, 1994. **Anais...** Morgan Kaufmann, 1994. p.70–77.

FÜRNKRANZ, J.; WIDMER, G. Incremental reduced error pruning. In: **Machine Learning Proceedings 1994**. Heidelberg: Elsevier, 1994. p.70–77.

GEHRKE, M.; WALKER, C.; WALKER, E. Some Comments on Interval Valued Fuzzy Sets. **Int. Journal of Intelligent Systems**, USA, v.11, n.10, p.751–759, 1996.

GERARD DRAPPER GIL, e. a. Disponível em: <https://www.unb.ca/cic/datasets/vpn.html>, VPN-nonVPN dataset (ISCXVPN2016) Site. Acesso em dezembro de 2022.

GERARD DRAPPER GIL, e. a. Disponível em: <https://github.com/ahlashkari/ISCXFlowMeter>, ISCXFlowMeter Site. Acesso em dezembro de 2022.

GNANAMBAL, S.; THANGARAJ, M.; MEENATCHI, V.; GAYATHRI, V. Classification algorithms with attribute selection: an evaluation study using WEKA. **International Journal of Advanced Networking and Applications**, USA, v.9, n.6, p.3640–3644, 2018.

GORZALCZANY, M. B. Interval-valued fuzzy inference involving uncertain (inconsistent) conditional propositions. **Fuzzy Sets and Systems**, USA, v.29, n.2, p.235–240, 1989.

HALL, M. A. Correlation-based feature subset selection for machine learning. **Thesis submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy at the University of Waikato**, USA, 1998.

HALL, M. A.; SMITH, L. A. Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper. In: TWELFTH INTERNATIONAL FLORIDA ARTIFICIAL INTELLIGENCE RESEARCH SOCIETY CONFERENCE, MAY 1-5, 1999, ORLANDO, FLORIDA, USA, 1999. **Proceedings...** AAAI Press, 1999. p.235–239.

HAO, J.; HO, T. K. Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. **Journal of Educational and Behavioral Statistics**, USA, v.44, n.3, p.348–361, jun 2019.

HERNÁNDEZ, P.; CUBILLO, S.; TORRES-BLANC, C. A Complementary Study on General Interval Type-2 Fuzzy Sets. **IEEE Trans. Fuzzy Syst.**, [S.l.], v.30, n.11, p.5034–5043, 2022.

HüHN, J.; HüLLERMEIER, E. FURIA: an algorithm for unordered fuzzy rule induction. **Data Mining and Knowledge Discovery**, USA, v.19, n.3, p.293–319, 2009.

IANA. Disponível em: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>, Service Name and Transport Protocol Port Number Registry. acesso em julho de 2019.

IGLESIAS, F.; MILOSEVIC, J.; ZSEBY, T. Fuzzy classification boundaries against adversarial network attacks. **Fuzzy Sets and Systems**, USA, v.368, p.20–35, 2019.

JAMMEH, E. A. et al. Interval type-2 fuzzy logic congestion control for video streaming across IP networks. **IEEE Trans Fuzzy Syst.**, USA, v.17, n.5, p.1123–1142, 2009.

JOY, C. Disponível em: <https://github.com/cisco/joy>, Joy Site. Acesso em dezembro de 2022.

JURIO, A. et al. Image magnification using interval information. **IEEE Transactions on Image Processing**, USA, v.20, n.11, p.3112–3123, 2011.

KAHRAMAN, C.; ÖZTAYşI, B.; ONAR, S. Çevik. A Comprehensive Literature Review of 50 Years of Fuzzy Set Theory. **International Journal of Computational Intelligence Systems**, USA, v.9, n.sup1, p.3–24, 2016.

KARAGIANNIS, T.; BROIDO, A.; FALOUTSOS, M.; CLAFFY, K. Transport Layer Identification of P2P Traffic. In: ACM SIGCOMM CONFERENCE ON INTERNET MEASUREMENT, 4., 2004. **Proceedings...** USA, 2004. p.121–134.

KERAS. Disponível em: <https://keras.io/>, Keras Site. Acesso em dezembro de 2022.

KIM, H. et al. Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. In: ACM CONEXT CONFERENCE, 2008., 2008. **Proceedings...** USA, 2008. p.11.

KLEINROCK, L.; NAYLOR, W. E. On Measured Behavior of the ARPA Network. In: MAY 6-10, 1974, NATIONAL COMPUTER CONFERENCE AND EXPOSITION, 1974. **Proceedings...** USA, 1974. p.767–780.

KLEMENT, E.; MESIAR, R.; PAP, E. Triangular norms. Position paper I: basic analytical and algebraic properties. **Fuzzy Sets and Systems**, USA, v.143, n.1, p.5–26, 2004.

KLEMENT, E. P.; MESIAR, R.; PAP, E. **Triangular Norms**. Dordrecht: Kluwer Academic Publisher, 2000.

KLEMENT, E. P.; MESIAR, R.; PAP, E. **Triangular norms**. USA: Springer Science & Business Media, 2013. v.8.

KLIR, G. J. Developments in uncertainty-based information. **Advances in computers**, USA, v.36, p.255–332, 1993.

KLIR, G. J. **Uncertainty and Information**: Foundations of Generalized Information Theory. USA: Wiley-Interscience Malden, USA, 2005.

KRUSE, R.; GEBHARDT, J.; KLAWONN, F. **Foundations of fuzzy systems**. USA: Wiley, 1994.

KUNST, R. et al. Improving network resources allocation in smart cities video surveillance. **Computer Networks**, USA, v.134, p.228–244, 2018.

KUROSE, J. F.; ROSS, K. W. **Computer Networking**: A Top-Down Approach. 8.ed. Boston, MA: Pearson, 2021.

LABAYEN GUEMBE, V.; MAGAÑA LIZARRONDO, E.; MORATÓ OSÉS, D.; IZAL AZCÁRATE, M. Online classification of user activities using machine learning on network traffic. **Computer Networks, 2020, 181: 107557**, USA, 2020.

LIANG, Q.; MENDEL, J. M. MPEG VBR video traffic modeling and classification using fuzzy technique. **IEEE Transactions on Fuzzy systems**, USA, v.9, n.1, p.183–193, 2001.

LIBPCAP. Disponível em: <https://www.tcpdump.org/>, Libpcap Site. Acesso em dezembro de 2022.

LICHTENBERG, M. P.; CURLESS, J. R. DECnet Transport Architecture. **Digital Technical Journal**, USA, v.4, p.40–40, 1992.

LIU, J. et al. Effective and real-time in-app activity analysis in encrypted internet traffic streams. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 23., 2017. **Proceedings...** Canada, 2017. p.335–344.

LOTFOLLAHI, M.; SIAVOSHANI, M. J.; ZADE, R. S. H.; SABERIAN, M. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning. **Soft Computing**, USA, p.1–14, 2017.

MAMDANI, E. H. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. In: SIXTH INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC, 1976, Los Alamitos, CA, USA. **Proceedings...** IEEE Computer Society Press, 1976. p.196–202. (MVL '76).

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International journal of man-machine studies**, USA, v.7, n.1, p.1–13, 1975.

MATHUR, L.; RAHEJA, M.; AHLAWAT, P. Botnet detection via mining of network traffic flow. **Procedia computer science**, USA, v.132, p.1668–1677, 2018.

MATZENAUER, M. et al. On admissible total orders for typical hesitant fuzzy consensus measures. **Int. J. Intell. Syst.**, USA, v.37, n.1, p.264–286, 2022.

MENDEL, J. M. Type-2 fuzzy sets and systems: an overview. **IEEE Computational Intelligence Magazine**, USA, v.2, n.1, p.20–29, 2007.

MENDEL, J. M. On KM Algorithms for Solving Type-2 Fuzzy Set Problems. **IEEE Transactions on Fuzzy Systems**, USA, v.21, n.3, p.426–446, 2013.

MENDEL, J. M.; JOHN, R. I.; LIU, F. Interval Type-2 Fuzzy Logic Systems Made Simple. **IEEE Trans. Fuzzy Systems**, USA, v.14, n.6, p.808–821, 2006.

MIRANDA RIOS, V. de; INÁCIO, P. R.; MAGONI, D.; FREIRE, M. M. Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms. **Computer Networks**, USA, v.186, p.107792, 2021.

MO, H.; MENG, Y.; WANG, F.; WU, D. Interval Type-2 Fuzzy Hierarchical Adaptive Cruise Following-Control for Intelligent Vehicles. **IEEE CAA J. Autom. Sinica**, USA, v.9, n.9, p.1658–1672, 2022.

MORENO, J. E. et al. Design of an interval Type-2 fuzzy model with justifiable uncertainty. **Inf. Sci.**, USA, v.513, p.206–221, 2020.

MOUSTAFA, N.; SLAY, J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: INT. CONF. MIL. COMMUN. INF. SYST. ICMCIS, 2015., 2015. **Anais...** Poland, 2015. p.1–6.

MOUSTAFA, N.; SLAY, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. **Information Security Journal: A Global Perspective**, USA, v.25, n.1-3, p.18–31, 2016.

NARI, S.; GHORBANI, A. A. Automated Malware Classification Based on Network Behavior. In: INTERNATIONAL CONFERENCE ON COMPUTING, NETWORKING AND COMMUNICATIONS (ICNC), 2013., 2013. **Anais...** USA, 2013. p.642–647.

NGUYEN, T. T.; ARMITAGE, G. J. A Survey of Techniques for Internet Traffic Classification using Machine Learning. **IEEE Communications Surveys and Tutorials**, USA, v.10, n.1-4, p.56–76, 2008.

NIU, W. et al. A heuristic statistical testing based approach for encrypted network traffic identification. **IEEE Transactions on Vehicular Technology**, USA, v.68, n.4, p.3843–3853, 2019.

ONTIVEROS, E.; MELIN, P.; CASTILLO, O. Comparative study of interval Type-2 and general Type-2 fuzzy systems in medical diagnosis. **Information Science**, USA, v.525, p.37–53, 2020.

OSISANWO, F. et al. Supervised machine learning algorithms: classification and comparison. **International Journal of Computer Trends and Technology (IJCTT)**, USA, v.48, n.3, p.128–138, 2017.

PALMEIRA, E. S.; BEDREGAL, B. R. C.; MESIAR, R.; FERNÁNDEZ, J. A new way to extend t-norms, t-conorms and negations. **Fuzzy Sets Syst.**, USA, v.240, p.1–21, 2014.

PANDA, A. K.; KOSKO, B. Random Fuzzy-Rule Foams for Explainable AI. In: FUZZY INFORMATION PROCESSING 2020 - PROCEEDINGS OF THE 2020 ANNUAL CONFERENCE OF THE NORTH AMERICAN FUZZY INFORMATION PROCESSING SOCIETY, NAFIPS 2020, REDMOND, WA, USA, 20-22 AUGUST 2020, 2020. **Anais...** Springer, 2020. p.253–266. (Advances in Intelligent Systems and Computing, v.1337).

PARFENOV, D.; ZABRODINA, L.; ZHIGALOV, A.; BOLODURINA, I. Research of multiclass fuzzy classification of traffic for attacks identification in the networks. In: J. PHYS. CONF. SER., 2020. **Anais...** USA, 2020. v.1679, n.4, p.042023.

PEDRYCZ, W. An Introduction to Computing with Fuzzy Sets. **IEEE ASSP Magazine**, USA, v.190, 2021.

PEDRYCZ, W.; GOMIDE, F. et al. **An introduction to fuzzy sets**: analysis and design. USA: Mit Press, 1998.

PEKALA, B. **Uncertainty Data in Interval-Valued Fuzzy Set Theory - Properties, Algorithms and Applications**. USA: Springer, 2019. (Studies in Fuzziness and Soft Computing, v.367).

PEKALA, B. et al. Interval-valued equivalence measures respecting uncertainty in image processing. **Int. J. Intell. Syst.**, USA, v.36, n.6, p.2767–2796, 2021.

PEREIRA, F. C. et al. **The MPEG-4 book**. USA: Prentice Hall Professional, 2002.

POURGHAFFARI, A.; BARARI, M.; SEDIGHIAN KASHI, S. An efficient method for allocating resources in a cloud computing environment with a load balancing approach. **Concurrency and Computation: Practice and Experience**, USA, v.31, n.17, p.e5285, 2019. e5285 cpe.5285.

POYMANOVA, E.; TATARNIKOVA, T. Models and methods for studying network traffic. In: WAVE ELECTRONICS AND ITS APPLICATION IN INFORMATION AND TELECOMMUNICATION SYSTEMS (WECONF), 2018. **Anais...** USA, 2018. p.1–5.

QADER, K.; ADDA, M.; AL-KASASSBEH, M. Comparative analysis of clustering techniques in network traffic faults classification. **Int. j. innov. res. comput. commun. eng.**, USA, v.5, n.4, p.6551–6563, 2017.

QIU, T. et al. How can heterogeneous Internet of Things build our future: A survey. **IEEE Communications Surveys & Tutorials**, USA, v.20, n.3, p.2011–2027, 2018.

QUINLAN, J. **C4.5**: Programs for Machine Learning. USA: Morgan Kauffman, 1993.

RAMEZANI, F.; Lu, J.; HUSSAIN, F. An online fuzzy Decision Support System for Resource Management in cloud environments. In: JOINT IFSA WORLD CONGRESS AND NAFIPS ANNUAL MEETING (IFSA/NAFIPS), 2013. **Anais...** Canada, 2013. p.754–759.

RAMEZANI, F.; NADERPOUR, M.; LU, J. A multi-objective optimization model for virtual machine mapping in cloud data centres. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (FUZZ-IEEE), 2016. **Anais...** Canada, 2016. p.1259–1265.

RAO, A. et al. Network characteristics of video streaming traffic. In: OF THE SEVENTH CONFERENCE ON EMERGING NETWORKING EXPERIMENTS AND TECHNOLOGIES, 2011. **Proceedings...** USA, 2011. p.1–12.

REISER, R. H. S.; BEDREGAL, B. R. C. Correlation in Interval-Valued Atanassov's Intuitionistic Fuzzy Sets - Conjugate and Negation Operators. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, USA, v.25, n.5, p.787–820, 2017.

REISER, R. H. S.; BEDREGAL, B. R. C.; REIS, G. A. A. dos. Interval-valued fuzzy coimplications and related dual interval-valued conjugate functions. **J. Comput. Syst. Sci.**, USA, v.80, n.2, p.410–425, 2014.

REZAEI, S.; LIU, X. Deep learning for encrypted traffic classification: An overview. **IEEE communications magazine**, USA, v.57, n.5, p.76–81, 2019.

RIZZI, A.; IACOVAZZI, A.; BAIOCCHI, A.; COLABRESE, S. A low complexity real-time internet traffic flows neuro-fuzzy classifier. **Computer Networks**, USA, v.91, p.752–771, 2015.

RIZZI, A.; PANELLA, M.; MASCIOLI, F. F. Adaptive resolution min-max classifiers. **IEEE Transactions on Neural Networks**, USA, v.13, n.2, p.402–414, 2002.

ROSS, T. Classical Sets and Fuzzy Sets. **Fuzzy logic with engineering applications**, USA, p.117–173, 2010.

ROSS, T. J. **Classical Relations and Fuzzy Relations**. USA: John Wiley and Sons, Ltd, 2010. 48-88p.

ROSS, T. J. Properties of membership functions, fuzzification, and defuzzification. **Fuzzy logic with engineering applications**, USA, p.89–116, 2010.

SALMAN, O.; ELHAJJ, I. H.; KAYSSI, A.; CHEHAB, A. A review on machine learning–based approaches for Internet traffic classification. **Annals of Telecommunications**, USA, v.75, n.11, p.673–710, 2020.

SAMBUC, R. **Fonctions and Floues**: Application a l'aide au Diagnostic en Pathologie Thyroidienne. FRA: Faculté de Médecine de Marseille, 1975.

SANDVINE. **The Global Internet Phenomena Report COVID-19 Spotlight**. USA: Sandvine Fremont, CA, 2020.

SANI, Y.; MAUTHE, A.; EDWARDS, C. Adaptive bitrate selection: A survey. **IEEE Communications Surveys & Tutorials**, USA, v.19, n.4, p.2985–3014, 2017.

SANTANA, F.; BEDREGAL, B.; VIANA, P.; BUSTINCE, H. On admissible orders over closed subintervals of [0,1]. **Fuzzy Sets and Systems**, USA, v.399, p.44–54, 2020. Fuzzy Intervals.

SANZ, J. A.; BUSTINCE, H. A wrapper methodology to learn interval-valued fuzzy rule-based classification systems. **Appl. Soft Comput.**, USA, v.104, p.107249, 2021.

SANZ, J. A.; FERNANDEZ, A.; BUSTINCE, H.; HERRERA, F. IVTURS: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection. **IEEE Transactions on Fuzzy Systems**, USA, v.21, n.3, p.399–411, 2013.

SANZ, J. A. et al. A case study on medical diagnosis of cardiovascular diseases using a Genetic Algorithm for Tuning Fuzzy Rule-Based Classification Systems with Interval-Valued Fuzzy Sets. In: IEEE SYMPOSIUM ON ADVANCES IN TYPE-2 FUZZY LOGIC SYSTEMS, T2FUZZ 2011, PARIS, FRANCE, APRIL 12-13, 2011, 2011., 2011. **Anais...** IEEE, 2011. p.9–15.

SANZ, J. A.; SOLA, H. B.; FERNÁNDEZ, A.; HERRERA, F. On the cooperation of interval-valued fuzzy sets and genetic tuning to improve the performance of fuzzy decision trees. In: FUZZ-IEEE 2011, IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, TAIPEI, TAIWAN, 27-30 JUNE, 2011, PROCEEDINGS, 2011. **Anais...** IEEE, 2011. p.1247–1254.

SANZ, J.; FERNANDEZ, A.; BUSTINCE, H.; HERRERA, F. IVTURS: a linguistic fuzzy rule-based classification system based on a new Interval-Valued fuzzy reasoning method with TUning and Rule Selection. **IEEE Transactions on Fuzzy Systems**, USA, v.21, n.3, p.399–411, 2013.

SEDDIKI, M.; PRADO, R. P. de; MUNOZ-EXPÓSITO, J. E.; GARCÍA-GALÁN, S. Fuzzy Rule-Based Systems for Optimizing Power Consumption in Data Centers. In: IMAGE PROCESSING AND COMMUNICATIONS CHALLENGES 5, 2014, Heidelberg. **Anais...** Springer International Publishing, 2014. p.301–308.

SHAFIQ, M.; YU, X.; WANG, D. Robust Feature Selection for IM Applications at Early Stage Traffic Classification Using Machine Learning Algorithms. In: IEEE 19TH INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS; IEEE 15TH INTERNATIONAL CONFERENCE ON SMART CITY; IEEE 3RD INTERNATIONAL CONFERENCE ON DATA SCIENCE AND SYSTEMS (HPCC/SMARTCITY/DSS), 2017. **Anais...** USA, 2017. p.239–245.

SHALAGINOV, A.; FRANKE, K. Automated generation of fuzzy rules from large-scale network traffic analysis in digital forensics investigations. In: INT. CONF OF SOFT COMPUTING AND PATTERN RECOGNITION (SOCPAR), 7., 2015. **Anais...** Japan, 2015. p.31–36.

SHAW, I. S. **Controle e Modelagem Fuzzy**. USA: Edgard Blücher LTDA, 1999.

SOLA, H. B. et al. Interval Type-2 Fuzzy Sets are Generalization of Interval-Valued Fuzzy Sets: Toward a Wider View on Their Relationship. **IEEE Trans. Fuzzy Syst.**, [S.l.], v.23, n.5, p.1876–1882, 2015.

SOLA, H. B. et al. Interval Type-2 Fuzzy Sets are Generalization of Interval-Valued Fuzzy Sets: Toward a Wider View on Their Relationship. **IEEE Transactions on Fuzzy Systems**, USA, v.23, n.5, p.1876–1882, 2015.

SOLEYMANPOUR, S.; SADR, H.; BEHESHTI, H. An Efficient Deep Learning Method for Encrypted Traffic Classification on the Web. In: INTERNATIONAL CONFERENCE ON WEB RESEARCH (ICWR), 6., 2020. **Anais...** Iran, 2020. p.209–216.

SPITERI, K.; SITARAMAN, R.; SPARACIO, D. From theory to practice: Improving bitrate adaptation in the DASH reference player. **ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)**, USA, v.15, n.2s, p.1–29, 2019.

STARCZEWSKI, J. T. **Advanced Concepts in Fuzzy Logic and Systems with Membership Uncertainty**. Berlin: Springer, 2013. (Studies in Fuzziness and Soft Computing, v.284).

STATISTA. Disponível em: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>, Statista Site. Acesso em dezembro de 2022.

STATISTA. Disponível em: <https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/>, Statista Site. Acesso em dezembro de 2022.

STOCKHAMMER, T. Dynamic adaptive streaming over HTTP– standards and design principles. In: ACM CONFERENCE ON MULTIMEDIA SYSTEMS, 2011. **Proceedings...** USA, 2011. p.133–144.

SUGENO, M. An introductory survey of fuzzy control. **Information sciences**, USA, v.36, n.1-2, p.59–83, 1985.

TAHAEI, H. et al. The rise of traffic classification in IoT networks: A survey. **Journal of Network and Computer Applications**, USA, v.154, p.102538, 2020.

TAKÁČ, Z. Inclusion and subsethood measure for interval-valued fuzzy sets and for continuous type-2 fuzzy sets. **Fuzzy Sets and Systems**, USA, v.224, p.106–120, 2013.

TAKÁČ, Z. Aggregation of fuzzy truth values. **Information Sciences**, USA, v.271, p.1–13, 2014.

TAKÁC, Z. et al. Width-Based Interval-Valued Distances and Fuzzy Entropies. **IEEE Access**, USA, v.7, p.14044–14057, 2019.

Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. **IEEE Transactions on Systems, Man, and Cybernetics**, USA, v.SMC-15, n.1, p.116–132, Jan 1985.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining, (First Edition)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

TAN, W. W.; CHUA, T. W. Uncertain rule-based fuzzy logic systems: introduction and new directions (Mendel, JM; 2001)[book review]. **IEEE Computational Intelligence Magazine**, USA, v.2, n.1, p.72–73, 2007.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer networks**. 6th.ed. USA: Pearson Education, 2021.

TAVALLAEE, M.; BAGHERI, E.; LU, W.; GHORBANI, A. A. A detailed analysis of the KDD CUP 99 data set. In: IEEE SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE FOR SECURITY AND DEFENSE APPLICATIONS, 2009. **Anais...** USA, 2009. p.1–6.

TCPDUMP. Disponível em: <https://www.tcpdump.org/>, Tcpdump Site. Acesso em dezembro de 2022.

TEAM, T. P. D. **pandas-dev/pandas**: Pandas. USA: Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.3509134>.

TENSORFLOW. Disponível em: <https://www.tensorflow.org/>, TensorFlow Site. Acesso em dezembro de 2022.

THEIN, T.; MYO, M. M.; PARVIN, S.; GAWANMEH, A. Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers. **Journal of King Saud University - Computer and Information Sciences**, SA, 2018.

TOGOU, M. A.; MUNTEAN, G.-M. An Elastic DASH-based Bitrate Adaptation Scheme for Smooth On-Demand Video Streaming. In: IEEE INTERNATIONAL SYMPOSIUM ON BROADBAND MULTIMEDIA SYSTEMS AND BROADCASTING (BMSB), 2022. **Anais...** China, 2022. p.1–6.

TOOSI, A. N.; BUYYA, R. A Fuzzy Logic-Based Controller for Cost and Energy Efficient Load Balancing in Geo-distributed Data Centers. In: IEEE/ACM 8TH INTERNATIONAL CONFERENCE ON UTILITY AND CLOUD COMPUTING (UCC), 2015. **Anais...** Greece, 2015. p.186–194.

TRIGUERO, I. et al. KEEL 3.0: an open source software for multi-stage analysis in data mining. **International Journal of Computational Intelligence Systems**, USA, 2017.

TURKSEN, I. B. Interval valued fuzzy sets based on normal forms. **Fuzzy Sets and Systems**, USA, v.20, n.2, p.191–210, 1986.

TURKSEN, I.; ZHONG, Z. An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets. **Fuzzy Sets and Systems**, USA, v.34, n.3, p.323–346, 1990.

VAN DER WALT, S.; COLBERT, S. C.; VAROQUAUX, G. The NumPy array: a structure for efficient numerical computation. **Computing in science & engineering**, USA, v.13, n.2, p.22–30, 2011.

VELAN, P.; ČERMÁK, M.; ČELEDA, P.; DRAŠAR, M. A survey of Methods for Encrypted Traffic Classification and Analysis. **International Journal of Network Management**, USA, v.25, n.5, p.355–374, 2015.

VENKATRAMAN, B. R.; NEWMAN-WOLFE, R. E.; CHOW, R.; LATCHMAN, H. A. Measurements and Characterization of Traffic in a University Environment. In: SOUTHEAST REGIONAL CONFERENCE, 30., 1992. **Proceedings...** USA, 1992. p.45–52.

VON ALTROCK, C. **Fuzzy logic and neurofuzzy applications in business and finance**. USA: Prentice-Hall, Inc., 1996.

WAGNER, C. Juzzy - A Java based toolkit for Type-2 Fuzzy Logic. In: IEEE SYMP. ON ADVANCES IN TYPE-2 FUZZY LOGIC SYSTEMS (T2FUZZ), 2013. **Anais...** USA, 2013. p.45–52.

Wagner, C.; Hagras, H. Interpreting fuzzy set operations and Multi Level Agreement in a Computing with Words context. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (FUZZ-IEEE 2011), 2011. **Anais...** Taiwan, 2011. p.2139–2146.

WALDEN, D. C. Host-to-host Protocols. **Network Systems and Software. Infotech State of–˜ A-6t Report**, USA, v.24, p.287–316, 1975.

WANG, P.; CHEN, X.; YE, F.; SUN, Z. A Survey of Techniques for Mobile Service Encrypted Traffic Classification Using Deep Learning. **IEEE Access**, USA, v.7, p.54024–54033, 2019.

WANG, W. et al. End-to-End Encrypted Traffic Classification with One-Dimensional Convolution Neural Networks. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENCE AND SECURITY INFORMATICS (ISI), 2017., 2017. **Anais...** Germany, 2017. p.43–48.

WEKA. Disponível em: <https://www.cs.waikato.ac.nz/ml/weka/>, Weka Site. Acesso em dezembro de 2022.

WIEGAND, T.; SULLIVAN, G. J.; BJONTEGAARD, G.; LUTHRA, A. Overview of the H. 264/AVC video coding standard. **IEEE Transactions on circuits and systems for video technology**, USA, v.13, n.7, p.560–576, 2003.

WIRESHARK. Disponível em: <https://www.wireshark.org/>, Wireshark Site. Acesso em dezembro de 2022.

WU, D. Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 2012. **Anais...** Australia, 2012. p.1–8.

WU, D.; NIE, M. Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems. In: FUZZ-IEEE, 2011. **Anais...** IEEE, 2011. p.2131–2138.

WU, J.; LUO, M. Fixed points of involutive interval-valued negations. **Fuzzy Sets and Systems**, USA, v.182, n.1, p.110–118, 2011. 70th Anniversary of Yingming Liu.

XU, Z.; YAGER, R. R. Some geometric aggregation operators based on intuitionistic fuzzy sets. **International journal of general systems**, USA, v.35, n.4, p.417–433, 2006.

YAMANSAVASCILAR, B.; GUVENSAN, M. A.; YAVUZ, A. G.; KARSLIGIL, M. E. Application identification via network traffic classification. In: INTERNATIONAL CONFERENCE ON COMPUTING, NETWORKING AND COMMUNICATIONS (ICNC), 2017., 2017. **Anais...** USA, 2017. p.843–848.

YANG, Y. et al. TLS/SSL Encrypted Traffic Classification with Autoencoder and Convolutional Neural Network. In: IEEE 20TH INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS, 2018. **Anais...** UK, 2018. p.362–369.

YAO, X. Evolving Artificial Neural Networks. **Proceedings of the IEEE**, USA, v.87, n.9, p.1423–1447, 1999.

ZADEH, L. A. Fuzzy sets. **Information and control**, USA, v.8, n.3, p.338–353, 1965.

ZADEH, L. A. Outline Of A New Approach To The Analysis Of Complex Systems And Decision Processes. **Systems, Man and Cybernetics, IEEE Transactions on**, USA, n.1, p.28–44, 1973.

ZADEH, L. A. Fuzzy logic and approximate reasoning. **Synthese**, USA, v.30, n.3-4, p.407–428, 1975.

ZADEH, L. A. Fuzzy logic, neural networks, and soft computing. **Communications of the ACM**, USA, v.37, n.3, p.77–84, 1994.

ZAPATA, H. et al. Interval-valued implications and interval-valued strong equality index with admissible orders. **Int. J. Approx. Reason.**, USA, v.88, p.91–109, 2017.

ZENG, W.; FENG, S. Approximate reasoning algorithm of interval-valued fuzzy sets based on least square method. **Information Sciencies**, USA, v.272, p.73–83, 2014.

Apêndices

APÊNDICE A – Ferramentas para Captura e Análise do Tráfego de Rede

Nesta Tese foram utilizadas as ferramentas Tcpdump[1], Wireshark[2] e CicFlowMeter[3] para captura e análise de tráfego de rede, as quais são baseadas na biblioteca Libpcap (LIBPCAP, 2022).

Estas ferramentas foram selecionadas tendo como critério central sua ampla aceitação pela comunidade técnico-científica, tanto nacional, como internacional, associado ao fato de serem disponibilizadas de forma publica na condição de software de código-fonte aberto

A.1 Ferramenta Tcpdump

A ferramenta Tcpdump (TCPDUMP, 2022) permite a captura e a análise de tráfego de rede. A interface é baseada em linha de comando e possui um grande número de filtros para realizar capturas de pacotes de rede. Os filtros podem ser aplicados a partir de informações dos cabeçalhos dos pacotes, características dos pacotes ou limites de tempo ou quantidade de pacotes a serem capturados.

O Tcpdump é distribuído sob a licença BSD (*Berkeley Software Distribution*) sendo um um software livre. O formato das capturas faz uso da biblioteca Libpcap (LIBPCAP, 2022).

A.2 Ferramenta Wireshark

O conjunto de aplicativos da ferramenta Wireshark (WIRESHARK, 2022) possibilita a captura e análise de pacotes de forma gráfica. Os pacotes podem ser capturados por linha de comando, por meio da ferramenta Tshark, ou por meio da interface gráfica. A partir dos arquivos de captura podem ser realizadas análises dos fluxos de tráfego de rede com o uso das dezenas de funcionalidades disponíveis na ferramenta.

A ferramenta Wireshark é distribuída sob a licença GPL (*GNU General Public License*) sendo um um software livre. Também no caso da Wireshark as capturas realizadas fazem uso da biblioteca Libpcap (LIBPCAP, 2022), o que contribui de modo oportuno para uma padronização das informações capturadas e seu decorrente pro-

[1] `https://www.tcpdump.org/`
[2] `https://www.wireshark.org/`
[3] `https://github.com/ahlashkari/CICFlowMeter`

cessamento, quando da geração dos *Datasets* a serem utilizados.

A.3 Ferramenta CicFlowMeter

A ferramenta CicFlowMeter[4] (GERARD DRAPPER GIL, 2022b) é um gerador e analisador de fluxo do tráfego de rede, sua escolha deu-se à pela ampla adoção na comunidade acadêmica, possuir licença sem custos para uso e ter a capacidade de extração de atributos a partir dos fluxos de rede. Esta ferramenta pode ser utilizada para analisar fluxos bidirecionais, onde o primeiro pacote determina as direções para frente (origem para destino (*upstream*)) e para trás (destino para origem (*downstream*)).

A ferramenta CicFlowMeter também fornece extensão de atributos para extrair informações úteis dos fluxos de tráfego de rede. Esta extensão pode extrair informações como endereços IP, portas, protocolos, taxas de transferência, tamanhos de pacotes, tempos de viagem, entre outras informações. Esta extensão também permite aos usuários criar seus próprios filtros de atributos para selecionar o tráfego de rede de interesse.

[4] `https://github.com/ahlashkari/CICFlowMeter`

APÊNDICE B – Classificadores

Nesta Tese foram utilizados algoritmos para classificação baseados em lógica fuzzy e aprendizagem de máquina. O critério adotado para a escolha dos algoritmos foi baseado nos trabalhos relacionados, bem como na premissa de serem disponibilizados para comunidade técnico-cientifica sem custos e/ou na condição de software de código fonte aberto.

B.1 Classificadores baseados em Lógica Fuzzy

Foram usados sistemas de classificação fuzzy baseados em regras (FRBCS (*Fuzzy Rule-Based Classification Systems*)) nos experimentos, precisamente, três deles que são baseados em lógica fuzzy tipo 1: ChiRW (Fuzzy Rule Learning Model by the Chi Approach with Rule Weights (CORDÓN; JESUS; HERRERA, 1999), FARC-HD (*Fuzzy Association Rule-based Classification method for High-Dimensional problems*), conforme em (ALCALÁ-FDEZ; ALCALÁ; HERRERA, 2011) e FURIA (*Fuzzy Unordered Rule Induction Algorithm*) introduzido em (HÜHN; HÜLLERMEIER, 2009). O último algoritmo usado é baseado na lógica fuzzy do tipo 2, denominado IVTURS (*Linguistic fuzzy rule-based classification Interval-Valued fuzzy reasoning method with TUning and Rule Selection*) (SANZ et al., 2013b). Tais algoritmos estão disponíveis para uso na ferramenta KEEL (ALCALÁ-FDEZ et al., 2009)[1], incluindo informações adicionais.

ChiRW: o objetivo é obter uma base de regras fuzzy que melhor se adapte aos dados de treinamento. Para gerar a Base de Regras fuzzy (RB (*Rules Base*)) este FRBCS projetou um método que determina a relação entre as variáveis do problema e estabelece uma associação entre o espaço das features e o espaço das classes.

Fuzzy-FARCHD: o objetivo é ter um classificador baseado em regras fuzzy preciso e compacto com baixo custo computacional. Este método extrai regras de associação fuzzy limitando a ordem das associações para obter um conjunto reduzido de regras candidatas com menos atributos no antecedente. Em seguida, ele usa um esquema de ponderação de padrões para reduzir o número de regras candidatas, pré-selecionando as regras com melhor qualidade. Finalmente, uma

[1] Disponível em: `http://www.keel.es`

seleção de regras genéticas e um ajuste lateral são aplicados para selecionar um pequeno conjunto de regras de associação fuzzy com alta precisão de classificação.

FURIA: o objetivo é gerar um conjunto compacto de boas regras difusas a partir de dados numéricos. O algoritmo FURIA realizada a aprendizagem de regras fuzzy baseado na implementação RIPPER (COHEN, 1995b). A principal diferença entre FURIA e RIPPER é que FURIA não faz uso de regras padrão. Além disso, o FURIA tem um procedimento de poda (*pruning*) alterado, o que significa que a poda durante o IREP (*Incremental Reduced Error Pruning*) (FÜRNKRANZ; WIDMER, 1994) foram desativadas permanentemente. Esta descoberta experimental melhorou a taxa de classificação.

IVTURS: o objetivo é extrair um conjunto compacto de boas regras fuzzy a partir de dados numéricos. O IVTURS é composto de três etapas: 1) A geração inicial de um sistema fuzzy intervalar baseado em regras de classificação IV-FRBCS (*Interval-Valued Fuzzy Rule-Based Classification System*). Para fazer isso, a base de regras é aprendida primeiro usando o algoritmo FARC-HD e então, os rótulos linguísticos são modelados como conjuntos fuzzy intervalares (IVFSs (*Interval-Valued Fuzzy Sets*)); 2) A aplicação de um IV-FRM (*Interval-Valued Fuzzy Reasoning Method*); e 3) Uma etapa de otimização usando a sinergia entre o ajuste da equivalência e a seleção da regra.

B.2 Classificadores Baseados em Aprendizagem de Máquina

A abordagem FuzzyNetClass também considera algoritmos de aprendizagem de máquina. Esses algoritmos são bastante conhecidos pela comunidade acadêmica e tiveram bons resultados de desempenho para classificação (OSISANWO et al., 2017). Precisamente, foram considerados os algoritmos: SVM (*Support Vector Machine*) (CORTES; VAPNIK, 1995), C.45 (QUINLAN, 1993), KNN (*K-Nearest Neighbors*) (COVER; HART, 1967) e GANN (*Genetic Algorithm-Neural Network*) (YAO, 1999), brevemente descritos a seguir[2].

SVM: um modelo de aprendizado supervisionado com algoritmos de aprendizado associados que analisam dados usados para classificação e análise de regressão. Os dados são transformados por meio de uma função Kernel, que aumenta a dimensionalidade dos dados. Esse aumento permite que os dados sejam separados por um hiperplano com probabilidade muito maior e estabelece uma medida mínima de erro de probabilidade de predição.

[2]Esses algoritmos são de código fonte aberto e estão disponíveis na ferramenta KEEL.

C.45: um algoritmo de geração de árvore de decisão que induz regras de classificação na forma de árvores de decisão a partir de um conjunto de exemplos dados. A árvore de decisão é construída de cima para baixo. Em cada etapa, um teste para o nó real é escolhido, começando com o nó raiz, que separa melhor os exemplos dados por classes. C45 é baseado no algoritmo ID3. As extensões ou melhorias do ID3 são: ele considera valores indisponíveis ou ausentes nos dados, manipula faixas contínuas de valores de atributos, escolhe uma medida de seleção de atributo apropriada, maximizando o ganho, e poda as árvores de decisão resultantes.

KNN: um algoritmo de aprendizado de máquina supervisionado que pode resolver problemas de classificação e regressão. Uma instância é classificada aplicando o método dos K vizinhos mais próximos, ou seja, considerando as K instâncias com a menor distância àquela a ser classificada.

GANN: um método híbrido que combina algoritmos genéticos (GA) e redes neurais artificiais (ANN) que são aplicados em problemas de predição. Cada rede é codificada usando duas matrizes: uma binária que armazena se cada conexão está ativa e uma real que corresponde aos pesos das conexões da rede. O cruzamento padrão de dois pontos é usado.

APÊNDICE C – Ambientes de *Soft Computing*

Nesta Tese foram utilizadas duas ferramentas e uma plataforma para concepção dos procedimentos empregados na prototipação da abordagem FuzyNetClass, as quais foram utilizadas tanto para a classificação de tráfego de rede, como para seleção e validação de atributos utilizados na classificação.

C.1 Ferramenta Juzzy

A modelagem do sistema FuzzyNetClass foi realizada utilizando a ferramenta Juzzy, inicialmente apresentada a comunidade científica em (WAGNER, 2013). Esta plataforma vem sendo constantemente revisada e evoluída, tendo seu emprego cada vez mais disseminado nos últimos anos (D'ALTERIO et al., 2020).

A plataforma Juzzy apresenta alternativas para modelagem e implementação de sistemas fuzzy tipo-1 e tipo-2, sendo um projeto de código-fonte aberto atualmente disponível para comunidade acadêmica, e que recentemente abordou o tratamento para conjuntos fuzzy tipo-2 de intervalo restrito. Os diferentes spectos da sua implementação podem ser encontrados no repositório mantido pelo grupo de pesquisa, o qual pode ser acessado pela URL: http:juzzy.wagnerweb.net/

C.2 Plataforma WEKA

A ferramenta WEKA (*Waikato Environment for Knowledge Analysis*)[1] é uma suíte de algoritmos para o processo de KDD (*Knowledge Data Extraction*) (TAN; STEINBACH; KUMAR, 2005). Atualmente é mantida por uma comunidade de entusiastas por ser um software livre disponível sobre a licença GPL, que permite que os usuários criem modelos de aprendizado de máquina usando algoritmos de aprendizado de máquina, incluindo árvores de decisão, regressão logística e agrupamento.

A WEKA provê diversos algoritmos para seleção de atributos, com possibilidade do uso de parâmetros para ajustes no comportamento de cada um dos algoritmos. A ferramenta provê uma variedade de técnicas e algoritmos de seleção de atributos. Estes algoritmos possibilitam identificar os melhores atributos para incluir em modelos de aprendizagem de máquina, pois dão suporte aos procedimentos de remoção dos

[1] `https://www.cs.waikato.ac.nz/ml/weka/`

atributos irrelevantes ou redundantes que podem prejudicar o desempenho do modelo.

C.3 Ferramenta KEEL

A ferramenta KEEL (*Knowledge Extraction based on Evolutionary Learning*)) (TRIGUERO et al., 2017) foi desenvolvida em linguagem Java e implementa dezenas de algoritmos para classificação de dados, que pode ser usada para um grande número de diferentes tarefas de descoberta de dados de conhecimento.

O ambiente do KEEL fornece uma interface gráfica intuitiva baseada no fluxo de dados para projetar experimentos com diferentes conjuntos de dados e algoritmos de inteligência computacional. Além disto, a ferramenta provê módulo para importação e edição de *Datasets* com geração do particionamento dos dados para testes e treinamento.