

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ROGÉRIO PEREZ PEREIRA JUNIOR

TRANSMISSÃO ADAPTATIVA DE VÍDEO EM IOT UTILIZANDO
APRENDIZADO DE MÁQUINA

Leopoldina

2025

ROGÉRIO PEREZ PEREIRA JUNIOR

**TRANSMISSÃO ADAPTATIVA DE VÍDEO EM IOT UTILIZANDO
APRENDIZADO DE MÁQUINA**

Trabalho de Conclusão de
Curso apresentado ao Curso de Engenharia
de Computação do Centro Federal de
Educação Tecnológica de Minas Gerais,
do Campus Leopoldina, como parte do
requisito para obtenção do título de Bacharel
em Engenharia de Computação.

Orientador: Fabiano Pereira Bhering

Coorientador: Fernando Garcia Diniz

Campos Ferreira

Leopoldina

2025

Pereira Junior, Rogério Perez.

P436t Transmissão adaptativa de vídeo em IoT utilizando aprendizado de máquina. / Rogério Perez Pereira Júnior. – 2025.
108 f. : il. gráfs., tabs.

Orientador: Fabiano Pereira Bhering.
Coorientador: Fernando Garcia Diniz Campos Ferreira.
Bibliografia: f. 103-108.

Trabalho de Conclusão de Curso (Graduação) – Centro Federal de Educação Tecnológica de Minas Gerais. *Campus Leopoldina*. Engenharia de Computação, Departamento de Computação, 2025.

1. Redes de computadores. 2. Internet das coisas (IoT). 3. Câmeras de vídeo. 4. Aprendizado de máquina. 5. Medidas de variabilidade. I. Bhering, Fabiano Pereira. II. Ferreira, Fernando Garcia Diniz Campos. III. Título.

CDU: 004.7

Elaboração da ficha catalográfica pela bibliotecária Luzia Adriana Damasceno,
CRB 6º- 2305. / CEFET-MG *campus Leopoldina*.

ROGÉRIO PEREZ PEREIRA JUNIOR

**TRANSMISSÃO ADAPTATIVA DE VÍDEO EM IOT UTILIZANDO
APRENDIZADO DE MÁQUINA**

Trabalho de conclusão de curso apresentado
ao Curso de Engenharia de Computação do
Centro Federal de Educação Tecnológica de
Minas Gerais, do Campus Leopoldina, como
parte do requisito para obtenção do título de
Bacharel em Engenharia de Computação.

Aprovado em: 16 de dezembro de 2025

Banca Examinadora:

Prof. Dr. Fabiano Pereira Bhering- Orientador
CEFET-MG

Prof. Dr. Fernando Garcia Diniz Campos Ferreira- Coorientador
CEFET-MG

Prof. Dr. José Eduardo Henriques da Silva
CEFET-MG

Prof. Tiago Bittencourt Nazare
CEFET-MG

(Assinaturas Digitais)

**Leopoldina
2025**



ATA Nº 61/2025 - DECOMLP (11.61.11)

(Nº do Protocolo: 23062.066822/2025-91)

(Assinado digitalmente em 17/12/2025 09:15)

FABIANO PEREIRA BHERING

PROFESSOR ENS BASICO TECN TECNOLOGICO

DECOMLP (11.61.11)

Matrícula: ###564#0

(Assinado digitalmente em 17/12/2025 12:16)

FERNANDO GARCIA DINIZ CAMPOS FERREIRA

PROFESSOR ENS BASICO TECN TECNOLOGICO

DECOMLP (11.61.11)

Matrícula: ###845#9

(Assinado digitalmente em 20/12/2025 01:12)

JOSE EDUARDO HENRIQUES DA SILVA

PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO

DECOMLP (11.61.11)

Matrícula: ###170#4

(Assinado digitalmente em 17/12/2025 09:49)

TIAGO BITTENCOURT NAZARE

PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO

DECOMLP (11.61.11)

Matrícula: ###397#7

(Assinado digitalmente em 17/12/2025 21:10)

ROGÉRIO PEREZ PEREIRA JUNIOR

DISCENTE

Matrícula: 2019#####1

Visualize o documento original em <https://sig.cefetmg.br/documentos/> informando seu número: **61**, ano: **2025**, tipo: **ATA**, data de emissão: **17/12/2025** e o código de verificação: **0a90a3a783**

À minha mãe, que sempre esteve ao meu lado, cuidando e incentivando a ser forte, mesmo nos momentos mais difíceis. Seu amor incondicional e apoio constantes foram a base para eu alcançar meus objetivos.

À minha namorada, é essencial em minha caminhada. Você compartilhou comigo seus dias, suas forças e sonhos, sempre me fortalecendo e incentivando a seguir em frente, com paciência e dedicação. Sem o seu apoio diário, esse trabalho não teria sido possível.

Aos meus amigos e irmãos, que sempre estiveram presentes, oferecendo apoio, conselhos e, acima de tudo, irmandade. Vocês são pra mim fonte de motivação e o alicerce sobre o qual me apoio para seguir sempre em frente.

A todos vocês, minha eterna gratidão e amor.

AGRADECIMENTOS

Primeiramente, agradeço a Deus, que me guiou e me deu forças em todos os momentos desta caminhada. Sem Sua presença em minha vida, nada disso teria sido possível.

Agradeço à minha mãe, Elaine, pelo apoio constante, amor incondicional e dedicação. Você me ensinou a ser forte e a persistir nos momentos mais difíceis. Sem você, nada disso seria possível.

Em especial, dedico este trabalho à memória do meu tio Duda, que partiu na véspera da entrega deste TCC. Ele foi como um pai para mim, sempre presente, me ensinando e me dando motivos para acreditar e seguir em frente com a faculdade. Sua força, carinho e exemplos ficarão para sempre marcados em minha vida e em cada conquista. Também dedico à memória do meu pai, que sempre me ensinou a nunca desistir dos meus sonhos e cuja lembrança me deu forças nos momentos mais difíceis. Ambos foram fundamentais na minha trajetória e este trabalho é, sobretudo, uma homenagem à importância que tiveram na minha formação e na pessoa que sou hoje.

Agradeço imensamente à minha namorada, Silvia, pelo apoio incansável, pelas palavras de incentivo e por estar ao meu lado todos os dias, me fortalecendo e me motivando a seguir em frente. Sua paciência e amor foram fundamentais durante essa jornada.

Aos meus avós, Walter e Olga, que, mesmo à distância, sempre me incentivaram e me ajudaram como puderam. Suas palavras de apoio e carinho foram fundamentais para que eu pudesse continuar minha trajetória, e sou eternamente grato por tudo o que fizeram por mim.

Ao professor Fabiano Bhering, pela orientação e apoio prestados ao longo do desenvolvimento deste trabalho. Suas contribuições e conselhos foram fundamentais para o êxito desta pesquisa.

Aos amigos Iuri, Humberto, Vinícius, Mauro, Marcelo(s), Gabriel, Artur e Igor, por sempre estarem ao meu lado, oferecendo conselhos, apoio e boas risadas. Cada um de vocês teve um papel importante em minha jornada, me ajudando a seguir em frente, mesmo nos momentos mais desafiadores.

Ao CEFET-MG e ao corpo docente, por me proporcionarem um ambiente de aprendizado de excelência, onde pude crescer como estudante e como pessoa. Sou grato a todos os professores e colaboradores que contribuíram para minha formação.

A todos vocês, minha eterna gratidão. Este trabalho é resultado do apoio e da força de cada um.

“Deus não promete uma jornada sem desafios, mas garante que o sucesso será a recompensa para aqueles que confiam Nele“

Autor Desconhecido

RESUMO

Este trabalho apresenta uma estratégia para roteamento adaptativo em transmissão de vídeo em ambientes de Internet das Coisas (IoT), considerando a movimentação das taxas de bits geradas pelos fluxos de vídeo. O método proposto baseia-se na classificação automática da variação das taxas de bits de vídeo, uma métrica estatística que reflete a dinâmica do movimento e a complexidade das cenas. Essas informações, combinadas com parâmetros como a taxa de quadros (FPS), são utilizadas para estimar as necessidades de fluxo e auxiliar na seleção de rotas no framework FITPATH, visando otimizar a qualidade de serviço (QoS) e a experiência do usuário (QoE) em cenários heterogêneos. A avaliação da proposta foi realizada por meio de experimentos e simulações, considerando diferentes tipos de movimentações de cenas. Os resultados demonstram o potencial da proposta para integração com roteamento adaptativo ciente do tráfego do vídeo.

Palavras-chave: Roteamento Adaptativo, Streaming de Vídeo, Coeficiente de Variação, Múltiplas Câmeras, Redes de IoT.

ABSTRACT

This paper presents an adaptive routing strategy for Internet of Things (IoT) video streaming environments, considering the bitrate delivery generated by video streams. The proposed method is based on the automatic classification of video bitrate variations, a statistical metric that reflects motion dynamics and scene complexity. This information, combined with parameters such as frame rates (FPS), is used to estimate flow needs and assist in route selection in the FITPATH framework, optimizing quality of service (QoS) and user experience (QoE) in heterogeneous scenarios. The proposal was evaluated through experiments and simulations, considering different types of motion scenes. The results demonstrate the proposal's potential for integration with video traffic-aware adaptive routing.

Keywords: Adaptive Routing, Video Transmission, Dynamic Bitrate Estimation, Multiple Cameras, IoT Networks.

TERMO DE RESPONSABILIDADE POR AUTORIA DO TRABALHO DE CONCLUSÃO DE CURSO

Eu, **ROGÉRIO PEREZ PEREIRA JUNIOR**, matrícula **20193014811**, discente do Curso de Engenharia de Computação, declaro para os devidos fins, que o presente Trabalho de Conclusão de Curso, de título **TRANSMISSÃO ADAPTATIVA DE VÍDEO EM IOT UTILIZANDO APRENDIZADO DE MÁQUINA**, é de minha autoria e que estou ciente dos:

- Artigos 297 a 299 do Código Penal;
- Decreto-Lei n 2.848 de 7 de dezembro de 1940;
- Lei no 9.610, de 19 de fevereiro de 1998, sobre os Direitos Autorais;
- Regulamento Disciplinar Discente do CEFET-MG;
- E que plágio consiste na reprodução de obra alheia e submissão dela como trabalho próprio ou na inclusão, em trabalho próprio, de ideias, textos, tabelas ou ilustrações (quadros, figuras, gráficos, fotografias, retratos, lâminas, desenhos, organogramas, fluxogramas, plantas, mapas e outros) transcritos de obras de terceiros sem a devida e correta citação da referência.

Por ser verdade, e por ter ciência do referido artigo, firmo a presente declaração, isentando o(a) professor(a) orientador(a) **FABIANO PEREIRA BHERING**, o(a) professor(a) coorientador(a) **FERNANDO GARCIA DINIZ CAMPOS FERREIRA**, os membros da banca examinadora e o Centro Federal de Educação Tecnológica de Minas Gerais, Campus Leopoldina, de qualquer responsabilidade.

Leopoldina 16 de dezembro de 2025

Assinatura do Discente

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Exemplo de predição temporal entre quadros em uma sequência de vídeo, demonstrando a utilização de quadros de referência passados e futuros (I, P, B frames) para explorar a complexidade temporal e otimizar a compressão. Fonte: Autor.	28
Figura 2.2 – Exemplos de modos de particionamento de blocos (8×8) utilizados na predição por compensação de movimento no padrão H.264/AVC. Diferentes divisões permitem melhor adaptação à complexidade espacial e ao padrão de movimento em diferentes regiões do quadro. Fonte: Autor.	29
Figura 2.3 – Exemplo de particionamento de blocos em H.265/HEVC. O padrão HEVC permite particionamentos ainda mais flexíveis, como quadrees, adaptando-se melhor à complexidade espacial das imagens e otimizando a eficiência da compressão. Fonte: Autor.	31
Figura 2.4 – Comparação entre Taxa de Bits Constante (Constant Bit Rate) (CBR) e VBR ao longo de uma sequência de vídeo. A linha azul representa CBR (taxa de bits constante), enquanto a linha vermelha representa VBR (taxa de bits variável). Fonte: Autor.	33
Figura 2.5 – Comparação das taxas de bits observadas em cenas estáticas, de movimento lento e de movimento rápido. Fonte: Autor.	34
Figura 2.6 – Arquitetura do FITPATH para Seleção de Caminhos Adaptativos, Adaptado: [6].	46
Figura 3.1 – Proposta da Arquitetura do FITPATH para Seleção de Caminhos Adaptativos. Adaptado de [7].	48
Figura 3.2 – modelo final selecionado pelo AutoML (TPOT).	51

Figura 3.3–Comando utilizado para gerar o arquivo de <i>trace</i> a partir de um vídeo MP4 utilizando o MP4Trace. Fonte: Autor.	55
Figura 3.4–Exemplo real de trecho de um arquivo de <i>trace</i> contendo o índice do frame, seu tamanho em bytes e o timestamp correspondente. Fonte: Autor.	56
Figura 3.5–Fluxograma do processo evolutivo do AutoML (TPOT).	67
Figura 4.1–Pacotes médios por frame para as classes detectadas pelo classificador, gerados a partir dos traces reconstruídos.	85
Figura 4.2–Trace representativo da classe <i>Abuse</i> . Fonte: Autor	87
Figura 4.3–Trace representativo da classe <i>Arrest</i> . Fonte: Autor	87
Figura 4.4–Trace representativo da classe <i>Arson</i> . Fonte: Autor	88
Figura 4.5–Trace representativo da classe <i>Assault</i> . Fonte: Autor	88
Figura 4.6–Trace representativo da classe <i>Burglary</i> . Fonte: Autor	89
Figura 4.7–Trace representativo da classe <i>Explosion</i> . Fonte: Autor	89
Figura 4.8–Trace representativo da classe <i>Fighting</i> . Fonte: Autor	90
Figura 4.9–Trace representativo da classe <i>Road Accidents</i> . Fonte: Autor	90
Figura 4.10–Trace representativo da classe <i>Robbery</i> . Fonte: Autor	91
Figura 4.11–Trace representativo da classe <i>Shooting</i> . Fonte: Autor	91
Figura 4.12–Trace representativo da classe <i>Shoplifting</i> . Fonte: Autor	92
Figura 4.13–Trace representativo da classe <i>Stealing</i> . Fonte: Autor	92
Figura 4.14–Trace representativo da classe <i>Vandalism</i> . Fonte: Autor	93
Figura 4.15–Topologia Rede Mesh, Fonte: Autor.	95
Figura 4.16–Comparação da vazão agregada (2 fluxos, 3 saltos) entre transmissões CBR das diferentes classes.Fonte: Autor	96
Figura 4.17–Comparação da vazão agregada (2 fluxos, 3 saltos) entre transmissões VBR das diferentes classes.Fonte: Autor	97
Figura 4.18–Comparação do delay médio (2 fluxos, 3 saltos) entre transmissões VBR das diferentes classes.Fonte: Autor.	98

Figura 4.19-Comparação da perda de pacotes média (2 fluxos, 3 saltos) entre transmissões
VBR das diferentes classes.Fonte: Autor 98

LISTA DE TABELAS

Tabela 3.1 – Resumo das métricas extraídas dos vídeos a partir dos traces de transmissão.	61
Tabela 3.2 – Desempenho do classificador ExtraTrees otimizando 13 classes de vídeos.	77
Tabela 3.3 – Precisão, recall e F1-score por classe do modelo extraído pelo TPOT.	77
Tabela 3.4 – Exemplo de frames contendo tempo e bits por frame para a classe <i>Abuse</i> .	80
Tabela 4.1 – Bitrate médio estimado para cada uma das 13 classes, calculado a partir dos valores médios de bits por frame do consolidado. Fonte: Autor. . . .	96

LISTA DE ABREVIATURAS E SIGLAS

IoT Internet das Coisas

QoS Qualidade de Serviço (Quality of Service)

QoE Qualidade da Experiência (Quality of Experience)

CBR Taxa de Bits Constante (Constant Bit Rate)

VBR Taxa de Bits Variável (Variable Bit Rate)

MAPE Estimador de Desempenho Consciente de Multimídia (Multimedia-Aware Performance Estimator)

H.264 Codificação de Vídeo Avançada (Advanced Video Coding)

FPS Quadros por Segundo (Frames Per Second)

QP Parâmetro de Quantização (Quantization Parameter)

RL Aprendizado por Reforço (Reinforcement Learning)

SRM Roteamento de Fonte com Multipath (Source Routing with Multipath)

AODV Vetor de Distância sob Demanda Ad Hoc (Ad hoc On-Demand Distance Vector)

OLSR Roteamento de Estado de Enlace Otimizado (Optimized Link State Routing)

ETX Contagem de Transmissão Esperada (Expected Transmission Count)

MOS Pontuação Média de Opinião (Mean Opinion Score)

RNA Redes Neurais Artificiais (Artificial Neural Networks)

LISTA DE SÍMBOLOS

$\sum$	Soma
Δt	Intervalo de tempo
N	Número total de pacotes ou quadros transmitidos
tamanho_i	Tamanho do i -ésimo pacote (em bytes)
taxa_kbps	Taxa de bits em kilobits por segundo

SUMÁRIO

1	Introdução	20
1.1	Objetivos	21
1.2	Justificativa	22
1.3	Metodologia	23
1.3.1	Levantamento e Definição dos Requisitos	23
1.3.2	Extração e Preparação dos Dados	24
1.3.3	Definição e Implementação do Modelo de Classificação	24
1.3.4	Classificação em Tempo de Execução	25
1.3.5	Integração com o Roteamento Adaptativo	25
1.3.6	Testes e Validação Experimental	25
2	Referencial Teórico	26
2.1	Transmissão de Vídeo em Redes Sem Fio	26
2.2	Compressão de Vídeo	27
2.2.1	Taxas de Bits de Vídeo	31
2.3	Protocolos de Roteamento para Transmissão de Vídeo em Redes Sem Fio	34
2.3.1	Protocolos de Roteamento Tradicionais	35
2.3.2	Protocolos Adaptativos para Vídeo	35
2.4	Classificação de Variação da Taxa de Bits para Roteamento Adaptativo	36
2.4.1	Modelos de Estimativa de Taxa de Bits	37
2.4.2	Algoritmos de AutoML para Classificação Supervisionada	40
2.4.2.1	Árvores de Decisão e Métodos Baseados em Conjuntos	41
2.4.2.2	Support Vector Machines	42
2.4.2.3	Regressão Logística e Naive Bayes	42

2.4.2.4	<i>K-Nearest Neighbors</i>	43
2.4.2.5	Métodos Baseados em Gradiente	43
2.4.3	Framework FITPATH para Roteamento Adaptativo	43
2.4.3.1	Heurística Iterativa de Seleção de Caminhos	44
2.4.3.2	Estimativa Determinística de Desempenho com o MAPE	45
2.4.3.3	Limitações do Modelo Atual	46
3	Proposta e Desenvolvimento	48
3.1	Modelagem para Classificação dos Fluxos de Vídeo	49
3.1.0.1	Reprodutibilidade e Condições de Generalização	53
3.1.1	Geração dos Traces com MP4Trace	54
3.1.1.1	Estrutura do Arquivo de <i>Trace</i>	55
3.1.2	Processamento dos Traces e Extração das Features Estatísticas	56
3.1.3	Métricas extraídas dos fluxos de vídeo	57
3.1.3.1	Média (Média)	58
3.1.3.2	Desvio padrão (Desvio Padrão)	58
3.1.3.3	Coeficiente de variação (Coef. Var)	58
3.1.3.4	Máximo e mínimo (Máx, Mín)	59
3.1.3.5	Amplitude (Amplitude)	59
3.1.3.6	IQR — Intervalo interquartil (IQR)	59
3.1.3.7	Spikes (Spikes)	59
3.1.3.8	Curtose (Curtose)	60
3.1.3.9	Assimetria (Assimetria)	60
3.1.3.10	Diferença absoluta média (MAD)	60
3.1.3.11	Percentil 90 (P90)	61
3.1.3.12	Média global	61
3.1.3.13	Construção do Dataset Consolidado	61
3.1.4	Treinamento e Seleção Automática do Modelo com TPOT	62

3.1.4.1	Preparação dos Dados de Treinamento e Teste	62
3.1.4.2	Otimização Automática com TPOT	64
3.1.5	Modelo Selecionado pelo AutoML (TPOT)	65
3.1.5.1	Pré-Processador	69
3.1.5.2	Classificador	69
3.1.5.3	Hiperparâmetros do ExtraTrees e sua função	70
3.1.5.4	Avaliação do Modelo	71
3.1.5.5	Matriz de Confusão	73
3.1.5.6	Matriz de Confusão Absoluta	74
3.1.5.7	Matriz de Confusão Normalizada	74
3.1.5.8	Acurácia	74
3.1.5.9	Precisão	75
3.1.5.10	Recall	75
3.1.5.11	Precisão, Recall e F1-score (Macro)	76
3.1.5.12	Classificação dos Vídeos por Meio do Modelo Treinado	79
3.2	Integração com o FITPATH	81
3.2.1	Implementação das Taxas Simuladas para o MAPE	81
3.2.2	Benefícios da integração final	82
4	Experimentos e Resultados	84
4.1	Distribuição das Classes	84
4.2	Comportamento Temporal dos Fluxos de Vídeo	86
4.3	Performance do MAPE utilizando o Classificador	94
5	Conclusão	99
5.1	Limitações do Trabalho	99
5.2	Trabalhos Futuros	101
	REFERÊNCIAS	103

1 INTRODUÇÃO

A transmissão de vídeo em tempo real é um dos maiores desafios em redes heterogêneas, como as redes de Internet das Coisas (IoT), onde as condições de largura de banda, latência e perda de pacotes podem variar significativamente. Nessas redes, a compressão de vídeo é necessária para garantir uma transmissão eficiente, pois reduz o volume de dados a serem enviados, tornando viável a comunicação mesmo em cenários com restrições severas de recursos. Sem compressão, a quantidade de dados gerada por vídeos de alta resolução seria incompatível com a capacidade de transmissão de muitos dispositivos IoT. Além disso, técnicas modernas de compressão permitem reduzir a taxa de bits sem comprometer significativamente a qualidade visual, garantindo uma melhor adaptação às condições dinâmicas do canal e mantendo a (Qualidade da Experiência (Quality of Experience) (QoE)) do usuário final [44, 49].

Em redes IoT, a transmissão de vídeo deve ser otimizada para lidar com a variabilidade das condições de rede, como largura de banda instável, latência e perdas de pacotes [11]. Para garantir a eficiência da transmissão, a classificação da variação da taxa de bits necessária para transmitir um vídeo é fundamental. O codificador de vídeo desempenha um papel crucial ao se ajustar dinamicamente às mudanças na complexidade das cenas, como movimentos rápidos e variações de conteúdo. Este ajuste dinâmico deve ser realizado de forma que a qualidade do vídeo seja preservada mesmo quando as condições de rede mudam significativamente. Modelos matemáticos que utilizam métricas de atividade espacial e temporal, juntamente com parâmetros como o Parâmetro de Quantização (Quantization Parameter) (QP) e a Quadros por Segundo (Frames Per Second) (FPS), têm se mostrado eficazes para prever essas flutuações e otimizar a configuração do codificador [32]. No entanto, essas classificações precisam ser integradas aos mecanismos de seleção de rotas, para permitir uma seleção de caminhos que considere as características específicas dos fluxos de vídeo a serem transmitidos.

Essa integração é importante para uma transmissão adaptativa, capaz de preservar a qualidade do vídeo em diferentes cenários de rede.

Neste sentido, esse trabalho propõe estratégias de roteamento adaptativo por meio de um modelo de classificação da variação da taxa de bits de vídeos, considerando a dinâmica das cenas. O modelo pode ser integrado ao framework FITPATH [6], projetado para selecionar dinamicamente os melhores caminhos de transmissão em redes IoT, levando em conta as estimativas da variação da taxa de bits dos fluxos de vídeo.

Diante desse cenário, surge a seguinte questão: como a integração de um modelo de classificação da variação da taxa de bits ao framework FITPATH [6] pode melhorar a transmissão de vídeo em redes IoT, considerando a preservação da qualidade e a adaptação às condições de rede?

A combinação do modelo de estimativa da variação da taxa de bits com o framework FITPATH [6] visa não apenas melhorar a Qualidade de Serviço (Quality of Service) (QoS), mas também proporcionar uma maior qualidade de experiência para o usuário QoE. Espera-se que os resultados deste trabalho possam auxiliar na seleção de rotas para melhorias no desempenho da rede, permitindo uma entrega de vídeo mais eficiente e com melhor qualidade, mesmo em condições adversas de rede [11].

1.1 Objetivos

Este trabalho tem como principal objetivo apresentar estratégias de roteamento adaptativo para otimizar a transmissão de vídeo em redes IoT heterogêneas, propondo um modelo para classificar a taxa de bits dinâmicas de vídeos utilizando métricas de atividade espacial e temporal, para melhorar a qualidade de serviço (QoS) e a experiência do usuário QoE.

Os objetivos específicos são:

- **Desenvolver um modelo de classificação da variação de taxa de bits**, baseado

em métricas de atividade espacial e temporal, além de parâmetros de codificação como o QP e a FPS.

- **Identificar a dinâmica das diferentes movimentações de cena em vídeos**, avaliando de que forma as variações na complexidade das cenas influenciam a taxa de bits requerida para assegurar uma qualidade de vídeo satisfatória.
- **Integrar o modelo de classificação com o framework FITPATH**, para otimizar o mecanismo de seleção de caminhos para transmissão de vídeos, considerando as condições da rede em tempo real.
- **Avaliar a precisão do modelo de classificação**, verificando sua capacidade de classificar a taxa de bits em diferentes cenários de transmissão, com base na codificação dos vídeos.
- **Analisar o desempenho do modelo proposto em relação as estimativas usadas pelo FITPATH**, como atraso, perdas de pacotes e vazão da rede, comparando com modelos baseado em taxa constante.

1.2 Justificativa

A constante evolução das redes de comunicação e o crescente consumo de conteúdo multimídia, especialmente vídeos, impõem novos desafios para a transmissão eficiente de dados. As redes de Internet das Coisas IoT têm se tornado cada vez mais populares, especialmente em aplicações como cidades inteligentes, saúde conectada e sistemas de monitoramento [11]. No entanto, essas redes enfrentam limitações significativas, como largura de banda restrita, latência variável e recursos computacionais limitados, o que pode prejudicar a qualidade da transmissão de vídeo em tempo real [51].

Este trabalho se justifica pela necessidade de desenvolver um modelo de estimativa da variação da taxa de bits que, além de considerar métricas de atividade espacial

e temporal, se integre a um sistema adaptativo para seleção de caminhos, como o FITPATH [6]. A integração da estimativa da variação da taxa de bits com o mecanismo de seleção de caminhos pode otimizar ainda mais a alocação de recursos na rede. Especificamente, uma classificação pode ser usada para prever as flutuações na demanda de largura de banda, ajustando a seleção dos melhores caminhos de transmissão de acordo com a necessidade de dados dos fluxos de vídeo em tempo real. Ao integrar essas classificações, o FITPATH [6] poderia ajustar dinamicamente os caminhos de transmissão com base na quantidade de dados necessária em cada momento, garantindo uma utilização mais eficiente da largura de banda disponível e uma transmissão de vídeo contínua e de alta qualidade.

1.3 Metodologia

Este trabalho caracteriza-se como uma pesquisa aplicada e descritiva, cujo objetivo é propor e avaliar um modelo de classificação da taxa de bits de fluxos de vídeo em redes IoT, com foco na otimização do roteamento adaptativo em ambientes sem fio. A metodologia adotada define explicitamente os procedimentos técnicos empregados, desde a obtenção dos dados até a validação experimental da solução proposta.

A metodologia é organizada nas etapas descritas a seguir.

1.3.1 Levantamento e Definição dos Requisitos

Inicialmente, são definidos os requisitos do sistema de classificação da taxa de bits a partir da análise das características dos fluxos de vídeo e das limitações inerentes às redes IoT. São considerados parâmetros de codificação de vídeo, como o QP e a FPS, bem como métricas de atividade espacial e temporal extraídas diretamente do tráfego gerado pelos vídeos.

Essas métricas incluem valores estatísticos calculados sobre janelas temporais

fixas do fluxo, tais como média, desvio padrão, coeficiente de variação, valores máximo e mínimo da taxa instantânea de bits, além da identificação de variações abruptas (spikes). A definição desses requisitos estabelece quais atributos serão utilizados como variáveis de entrada do modelo de classificação.

1.3.2 Extração e Preparação dos Dados

A extração dos dados é realizada a partir de logs de transmissão de vídeo, nos quais são registrados os tamanhos dos pacotes associados a cada quadro transmitido. A partir desses dados brutos, a taxa de bits é estimada em kbps considerando o tamanho dos pacotes, a taxa de quadros do vídeo e o intervalo temporal entre quadros.

Os dados são organizados em janelas temporais deslizantes com tamanho fixo, permitindo a análise do comportamento da taxa de bits ao longo do tempo. Para cada janela, são calculadas as métricas estatísticas previamente definidas. Em seguida, os dados passam por uma etapa de pré-processamento, que inclui normalização, padronização de formatos numéricos e balanceamento das classes por meio de técnicas de reamostragem, com o objetivo de evitar viés no treinamento do modelo.

1.3.3 Definição e Implementação do Modelo de Classificação

O modelo de classificação da taxa de bits é implementado utilizando técnicas de aprendizado de máquina supervisionado. Os vetores de características extraídos das janelas temporais constituem as entradas do modelo, enquanto a variável de saída corresponde à classe de taxa de bits do fluxo de vídeo, categorizada em níveis discretos (por exemplo, baixa, média e alta).

O conjunto de dados é dividido em subconjuntos de treinamento e teste, respeitando a proporção entre classes. O treinamento do modelo é realizado com ajuste dos hiperparâmetros e utilização de estratégias para lidar com desbalanceamento de classes. A avaliação do desempenho é feita por meio de métricas como acurácia, matriz de confusão

e medidas de precisão e revocação, permitindo verificar a capacidade do modelo em classificar corretamente os diferentes padrões de tráfego de vídeo.

1.3.4 Classificação em Tempo de Execução

Após o treinamento, o modelo é aplicado em um cenário de execução contínua, no qual novos dados de tráfego são processados em tempo quase real. À medida que novas janelas de dados são formadas, as métricas são extraídas e fornecidas ao classificador, que estima a classe de taxa de bits correspondente ao comportamento recente do fluxo de vídeo.

Essa classificação dinâmica permite acompanhar variações no perfil do tráfego ao longo da transmissão, fornecendo informações atualizadas sobre a demanda de rede associada a cada fluxo de vídeo.

1.3.5 Integração com o Roteamento Adaptativo

A saída do modelo de classificação é integrada ao framework FITPATH, sendo utilizada como parâmetro auxiliar no processo de seleção de múltiplos caminhos de transmissão. A classe de taxa de bits estimada é empregada para orientar decisões de roteamento, de forma a selecionar caminhos mais adequados às exigências do fluxo de vídeo, considerando a capacidade e as condições da rede.

Essa integração tem como objetivo reduzir perdas de pacotes e variações abruptas de qualidade, contribuindo para a melhoria da QoE dos usuários finais.

1.3.6 Testes e Validação Experimental

Por fim, são realizados experimentos controlados para validar a solução proposta. Os testes avaliam o impacto da classificação da taxa de bits nas decisões de roteamento e no desempenho da transmissão de vídeo.

2 REFERENCIAL TEÓRICO

2.1 Transmissão de Vídeo em Redes Sem Fio

A transmissão de vídeo em redes sem fio apresenta uma série de desafios específicos devido às características das próprias redes, como a variação na qualidade do sinal, a interferência e as limitações de largura de banda. Em redes sem fio, como as utilizadas em IoT, as condições de rede podem ser altamente dinâmicas, com flutuações constantes em termos de latência, largura de banda e perda de pacotes. Esses fatores podem afetar diretamente a qualidade da transmissão de vídeo, que depende de uma entrega estável e contínua de dados [2, 11].

A largura de banda disponível em redes sem fio é frequentemente compartilhada entre múltiplos dispositivos, o que pode resultar em congestionamento e em variações de velocidade de transmissão. Além disso, a latência das redes sem fio pode ser maior do que em redes cabeadas, devido ao tempo necessário para que os dados percorram as diferentes camadas de comunicação e enfrentem obstáculos físicos, como paredes e outros dispositivos interferentes [11]. Esses problemas tornam a transmissão de vídeo em tempo real um desafio, pois vídeos de alta qualidade requerem uma largura de banda estável e uma latência baixa para manter a continuidade da imagem e do som.

A compressão de vídeo se torna fundamental para mitigar os efeitos da largura de banda limitada, permitindo a redução da quantidade de dados transmitidos sem comprometer a qualidade visual [32]. No entanto, a adaptação da taxa de bits é um aspecto crítico, pois, em ambientes sem fio, é necessário ajustar a taxa de transmissão de acordo com as condições de rede, a fim de evitar interrupções e perdas de pacotes que podem degradar a experiência do usuário [6, 51].

Em redes sem fio, a implementação de mecanismos de adaptação em tempo real é necessária para lidar com as flutuações de rede e evitar uma deterioração significativa

da qualidade do vídeo. Sem essas estratégias, variações na largura de banda, latência ou perda de pacotes poderiam comprometer a entrega do conteúdo. Para mitigar esses efeitos, utilizam-se técnicas como controle de taxa de bits, ajustes dinâmicos na compressão do vídeo e estratégias de roteamento adaptativo, que permitem selecionar os melhores caminhos de transmissão, garantindo maior eficiência e menor latência possível [11].

2.2 Compressão de Vídeo

A compressão de vídeo pode ser definida como o conjunto de técnicas e algoritmos utilizados para reduzir a quantidade de dados necessária para representar uma sequência de imagens em movimento, de forma a viabilizar seu armazenamento ou transmissão em redes de comunicação, mantendo um nível aceitável de qualidade visual. Esse processo é essencial em sistemas multimídia modernos, especialmente em cenários com restrições de largura de banda, como redes sem fio e ambientes baseados em IoT [32].

De acordo com Lottermann et al. [32], técnicas modernas de compressão de vídeo permitem reduzir significativamente a taxa de bits necessária para a transmissão, sem prejuízo perceptível da qualidade visual. Essa eficiência é alcançada por meio da exploração de redundâncias inerentes ao sinal de vídeo, que podem ser classificadas, principalmente, como redundância temporal e redundância espacial.

A compressão de vídeo e a complexidade temporal está associada às variações dinâmicas do conteúdo ao longo do tempo, refletidas nas mudanças entre quadros consecutivos de uma sequência [48, 51]. Cenas que apresentam movimentos intensos ou alterações rápidas de conteúdo tendem a possuir maior complexidade temporal, o que impacta diretamente a eficiência da compressão e a taxa de bits necessária para a transmissão [32].

Para lidar com esse tipo de variação, técnicas de predição temporal são amplamente empregadas nos algoritmos de compressão de vídeo, fazendo uso de quadros de referência

passados e futuros com o objetivo de codificar apenas as diferenças relevantes entre quadros consecutivos, reduzindo assim a quantidade de dados transmitidos [48].

No contexto desses métodos, diferentes tipos de quadros desempenham papéis específicos na codificação de vídeo, conforme descrito na literatura e nos padrões de compressão modernos [40, 48]:

- **I-frames (intraquadros):** armazenam a imagem completa, sendo codificados sem dependência de outros quadros;
- **P-frames (quadros preditivos):** codificam apenas as diferenças em relação a quadros de referência anteriores;
- **B-frames (quadros bidirecionais):** exploram informações de quadros anteriores e posteriores para maximizar a eficiência da codificação.

A Figura 2.1 ilustra de forma esquemática o mecanismo de predição temporal em uma sequência de vídeo, evidenciando o uso de quadros de referência passados e futuros na codificação interquadro, conforme adotado em padrões como H.264/AVC [1, 48].

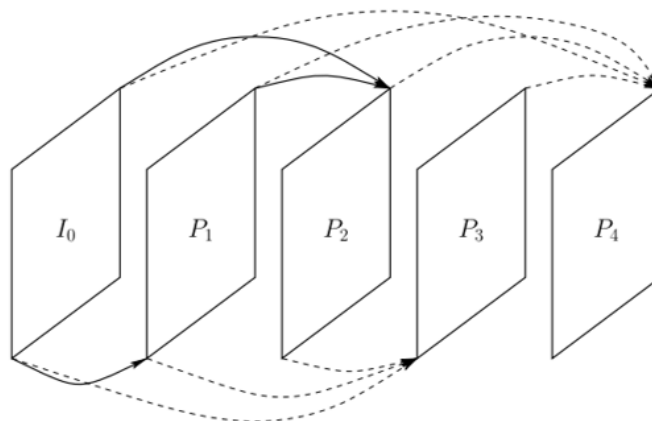


Figura 2.1 – Exemplo de predição temporal entre quadros em uma sequência de vídeo, demonstrando a utilização de quadros de referência passados e futuros (I, P, B frames) para explorar a complexidade temporal e otimizar a compressão. Fonte: Autor.

A complexidade espacial está relacionada ao nível de detalhe, textura e variação dentro de cada quadro individual. Regiões com maior riqueza de detalhes demandam maior quantidade de bits para serem representadas com qualidade, impactando diretamente a eficiência da compressão [48].

Para explorar essa característica, algoritmos modernos de compressão utilizam técnicas de codificação intraquadro, que exploram similaridades espaciais dentro do próprio quadro. No padrão H.264/AVC, diferentes modos de particionamento de blocos são empregados (como 8×8 , 4×8 , 8×4 e 4×4), permitindo melhor adaptação à complexidade espacial presente em diferentes regiões da imagem [40].

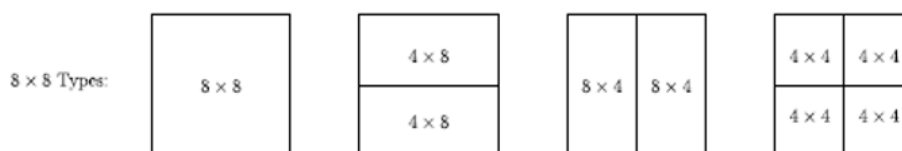


Figura 2.2 – Exemplos de modos de particionamento de blocos (8×8) utilizados na predição por compensação de movimento no padrão H.264/AVC. Diferentes divisões permitem melhor adaptação à complexidade espacial e ao padrão de movimento em diferentes regiões do quadro. Fonte: Autor.

Entre os codecs modernos, o Codificação de Vídeo Avançada (Advanced Video Coding) (H.264) — também conhecido como AVC (Advanced Video Coding) — destaca-se por sua ampla utilização em transmissões de vídeo em tempo real, devido à elevada eficiência de compressão e à manutenção de boa qualidade visual, mesmo em taxas de bits relativamente baixas [31]. O H.264 é amplamente empregado em serviços de streaming, videoconferências e diversas outras aplicações de vídeo ao vivo.

A codificação entre quadros (*interframe*) utiliza técnicas que permitem explorar as semelhanças temporais ao longo da sequência de vídeo. Um dos principais mecanismos empregados é a *predição com compensação de movimento*, fundamental em padrões como H.264/AVC e H.265/HEVC [48]. Essa técnica envolve dois estágios principais:

- **Estimativa de movimento:** para cada macrobloco do quadro atual, identifica-se a região mais semelhante em quadros de referência anteriores ou futuros, utilizando algoritmos de correspondência de blocos que determinam vetores de movimento precisos, muitas vezes com resolução de subpixel (meio ou um quarto de pixel) [24];
- **Compensação de movimento:** calcula-se o residual, ou seja, a diferença entre o bloco previsto e o bloco real do quadro atual, que então é transformado, quantizado e compactado por meio de codificação por comprimento de código variável (VLC).

Destaca-se que a predição com compensação de movimento é a etapa mais demandante computacionalmente em codecs modernos, podendo representar entre 80 e 90% do tempo total de processamento, especialmente em implementações baseadas no H.264/MPEG-4 AVC [24].

No entanto, o H.265 (HEVC - High Efficiency Video Coding) oferece uma compressão superior em comparação ao H.264, proporcionando uma redução significativa na taxa de bits necessária para transmitir vídeos de alta qualidade. Isso torna o H.265 ideal para vídeos em alta definição (HD) e 4K, permitindo uma transmissão de vídeo com menor consumo de largura de banda e mantendo excelente qualidade visual. Apesar de suas vantagens em termos de compressão, o H.265 exige maior poder de processamento para codificação e decodificação, o que pode ser um obstáculo em dispositivos com recursos limitados, como os encontrados em redes de IoT ou dispositivos móveis [11].

O padrão H.265/HEVC, mostrado na Figura 2.3, amplia a flexibilidade do particionamento de blocos, utilizando uma estrutura hierárquica baseada em *quadrees*, o que permite um ajuste ainda mais preciso às variações da complexidade espacial nos quadros.



Figura 2.3 – Exemplo de particionamento de blocos em H.265/HEVC. O padrão HEVC permite particionamentos ainda mais flexíveis, como quadrees, adaptando-se melhor à complexidade espacial das imagens e otimizando a eficiência da compressão. Fonte: Autor.

2.2.1 Taxas de Bits de Vídeo

Na transmissão de vídeo em redes, a taxa de bits é um fator determinante para a qualidade visual e a eficiência da comunicação, uma vez que define o volume de dados transmitidos por segundo e impacta diretamente na largura de banda necessária [32, 48]. Tradicionalmente, a taxa de bits pode ser controlada por duas abordagens principais: CBR (Constant Bitrate) e Taxa de Bits Variável (Variable Bit Rate) (VBR) (Variable Bitrate). Cada abordagem possui vantagens e desvantagens, que variam de acordo com as condições de rede e os requisitos da aplicação.

- CBR: Como o nome sugere, o CBR mantém a taxa de bits fixa durante toda a transmissão. Embora essa abordagem seja simples e previsível, ela não aproveita eficientemente a largura de banda disponível, especialmente em vídeos com cenas de diferentes complexidades. Em cenas mais simples, a quantidade de dados necessária para representar o vídeo poderia ser reduzida, mas com o CBR a taxa de bits permanece constante, desperdiçando largura de banda. Por outro lado, em cenas complexas, a taxa de bits fixada pelo CBR pode não ser suficiente

para manter a qualidade do vídeo, resultando em distorções visuais ou perda de detalhes [32].

- VBR: Diferente do CBR, o VBR ajusta a taxa de bits dinamicamente, conforme a complexidade da cena em cada momento. Em trechos com pouca movimentação, como imagens estáticas ou com baixo detalhamento, o VBR reduz a taxa de bits, economizando largura de banda. Já em cenas mais dinâmicas e complexas, onde mais detalhes precisam ser representados, a taxa de bits aumenta. Esse ajuste contínuo torna o VBR mais eficiente em termos de utilização da largura de banda e, em muitos casos, resulta em melhor qualidade visual em comparação ao CBR, pois permite alocar mais dados para cenas complexas sem desperdiçar recursos em trechos mais simples [11].

A variação da taxa de bits em esquemas de Taxa de Bits Variável (VBR) está diretamente relacionada à complexidade espacial e à complexidade temporal das cenas do vídeo. A complexidade espacial refere-se ao nível de detalhe, textura e variação presentes em cada quadro individual, enquanto a complexidade temporal está associada à quantidade de movimento ou mudanças entre quadros sucessivos.

Assim, cenas estáticas ou com baixa complexidade espacial e temporal exigem uma taxa de bits menor, pois há menos informações novas a serem codificadas. Em contrapartida, cenas com alta complexidade espacial (muitos detalhes e texturas) ou alta complexidade temporal (movimento intenso, cortes rápidos) demandam taxas de bits mais elevadas para preservar a qualidade visual e evitar a perda de detalhes importantes.

O uso do VBR permite otimizar a taxa de bits de acordo com a complexidade de cada cena, como ilustrado na Figura 2.4. Em cenas estáticas, como paisagens, a taxa de bits é reduzida significativamente, economizando largura de banda. Para cenas com movimento lento, como uma pessoa caminhando, a taxa de bits aumenta moderadamente para capturar os detalhes necessários. Por outro lado, cenas com movimento rápido,

como perseguições ou mudanças bruscas de cenário, demandam uma alta taxa de bits para manter a qualidade visual e preservar os detalhes [45].

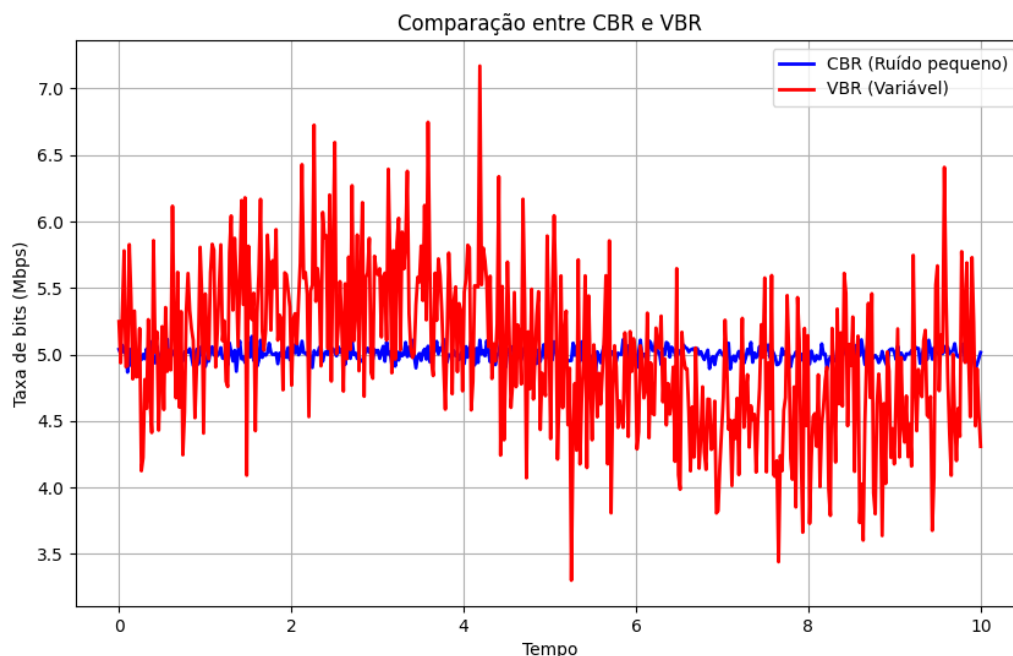


Figura 2.4 – Comparação entre CBR e VBR ao longo de uma sequência de vídeo. A linha azul representa CBR (taxa de bits constante), enquanto a linha vermelha representa VBR (taxa de bits variável). Fonte: Autor.

Esse comportamento adaptativo da taxa de bits, em função da complexidade das cenas, impacta diretamente o tráfego de vídeo nas redes. As transmissões em VBR podem apresentar picos de uso de banda — conhecidos como *rajadas* — especialmente durante cenas altamente dinâmicas. O conhecimento dessa dinâmica é fundamental para o planejamento e provisionamento dos recursos de rede, garantindo que os sistemas estejam preparados para lidar com variações abruptas no consumo de largura de banda sem prejudicar a experiência do usuário.

A Figura 2.5 exemplifica como diferentes tipos de cenas — estática, movimento lento e movimento rápido — resultam em diferentes padrões de taxa de bits ao longo do

tempo, reforçando a necessidade de técnicas inteligentes de adaptação e gerenciamento de rede em aplicações de vídeo.

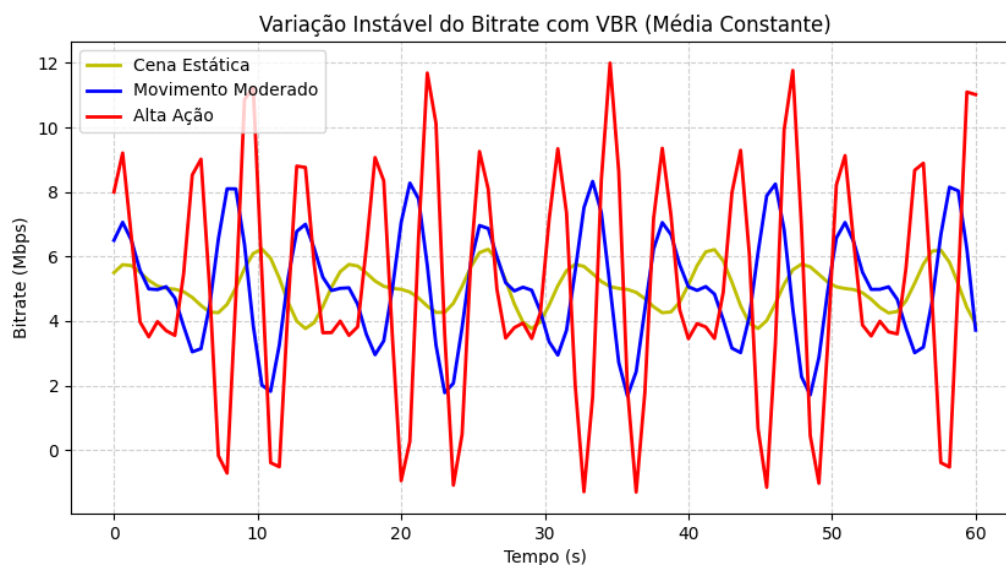


Figura 2.5 – Comparação das taxas de bits observadas em cenas estáticas, de movimento lento e de movimento rápido. Fonte: Autor.

2.3 Protocolos de Roteamento para Transmissão de Vídeo em Redes Sem Fio

A transmissão de vídeo em tempo real em redes sem fio apresenta desafios específicos, como a necessidade de garantir uma transmissão estável e de alta qualidade em condições de rede dinâmicas. Para garantir a entrega eficiente e contínua do vídeo, os protocolos de roteamento desempenham um papel fundamental.

Diversos protocolos de roteamento têm sido propostos com o objetivo de melhorar a transmissão de vídeo em redes sem fio. De modo geral, esses protocolos podem ser classificados em duas categorias principais: protocolos tradicionais de roteamento, que utilizam métricas fixas, e protocolos adaptativos, que consideram as condições dinâmicas da rede, como variações de largura de banda, atraso e perda de pacotes [4, 33].

2.3.1 Protocolos de Roteamento Tradicionais

Os protocolos de roteamento tradicionais, como o Roteamento de Estado de Enlace Otimizado (Optimized Link State Routing) (OLSR) (Optimized Link State Routing) e o Vetor de Distância sob Demanda Ad Hoc (Ad hoc On-Demand Distance Vector) (AODV), são amplamente utilizados em redes sem fio. Embora esses protocolos sejam eficientes em redes convencionais, eles podem não ser ideais para a transmissão de vídeo devido à sua falta de adaptação dinâmica a mudanças nas condições da rede.

O OLSR é um protocolo baseado em um método de roteamento de estado de link, onde cada nó mantém informações sobre a topologia da rede. Embora seja eficiente em ambientes estáticos, ele não leva em consideração as variações temporais e a dinâmica de largura de banda necessárias para a transmissão de vídeo em tempo real [13].

O protocolo AODV opera de maneira dinâmica, estabelecendo rotas somente quando necessário. Embora seja eficiente em termos de consumo de recursos, sua capacidade de lidar com flutuações de rede, como a variação de largura de banda para fluxos de vídeo, é limitada [42].

2.3.2 Protocolos Adaptativos para Vídeo

Para melhorar a transmissão de vídeo em redes sem fio, alguns protocolos de roteamento foram especificamente adaptados para lidar com a variação da largura de banda e a demanda de dados dos fluxos de vídeo. Esses protocolos ajustam dinamicamente as rotas de transmissão com base nas condições da rede e nas necessidades específicas de largura de banda do vídeo.

O protocolo Roteamento de Fonte com Multipath (Source Routing with Multipath) (SRM) permite a utilização de múltiplos caminhos para transmissão de dados, o que pode melhorar a resistência a perdas de pacotes e flutuações de largura de banda em redes sem fio. A utilização de múltiplos caminhos é particularmente útil em redes móveis, onde as

condições de conectividade podem mudar rapidamente [11].

O framework FITPATH propõe um mecanismo adaptativo de seleção de múltiplos caminhos, desenvolvido para redes IoT, com o objetivo de otimizar a transmissão de vídeo a partir das condições dinâmicas da rede e dos requisitos de taxa de bits dos fluxos de vídeo. Para isso, o FITPATH emprega uma abordagem heurística iterativa que avalia o desempenho da rede em tempo real, permitindo a seleção dos caminhos de transmissão mais adequados e a mitigação dos efeitos de interferência entre fluxos. O principal objetivo do framework é aumentar a eficiência da transmissão e maximizar a Qualidade de Experiência (QoE), mesmo em cenários adversos de rede [6].

Os detalhes do funcionamento do FITPATH e da proposta de integração adotada neste trabalho são apresentados na Subseção 2.4.3.

2.4 Classificação de Variação da Taxa de Bits para Roteamento Adaptativo

A classificação multiclasse da variação da taxa de bits necessária para a transmissão de vídeos constitui um elemento central para a otimização do uso da largura de banda e para a garantia da Qualidade de Serviço (QoS) em redes com recursos restritos. A literatura apresenta diversos métodos voltados à modelagem, estimativa e previsão do tráfego de vídeo, geralmente baseados em métricas de atividade espacial e temporal extraídas do conteúdo [30, 48].

Entretanto, conforme apontado por Li et al. [30], a maior parte dos trabalhos concentra-se na estimativa direta da taxa de bits ou em classificações simplificadas do tráfego, não explorando de forma sistemática abordagens de classificação multiclasse voltadas à tomada de decisão em mecanismos de controle e roteamento adaptativo.

Nesse contexto, Lottermann et al. [32] propõem um modelo baseado em características de movimento para prever a taxa de compressão de vídeos, demonstrando que padrões de movimentação influenciam diretamente a necessidade de largura de banda. Embora

eficaz, o modelo concentra-se na predição contínua da taxa de bits, não abordando explicitamente o problema sob a perspectiva de uma classificação multiclasse orientada à seleção de rotas. o.

Contudo, esses métodos geralmente tratam o problema como binário ou com foco apenas na estimativa direta da taxa, e não abordam a classificação multiclasse da variação de taxa de bits (por exemplo, baixa, média e alta complexidade), nem lidam de maneira eficaz com flutuações dinâmicas durante a transmissão. Além disso, tais modelos não apresentam integração com mecanismos adaptativos de roteamento, especialmente em cenários reais de redes distribuídas e heterogêneas.

No contexto de mecanismos como o FITPATH, que opera de forma adaptativa e orientada à performance da rede, a identificação prévia da classe de variação da taxa de bits pode otimizar a seleção de caminhos. Isso ocorre porque a dinâmica da taxa de bits é diretamente influenciada pela complexidade da cena e pela intensidade dos movimentos presentes no vídeo. Sem essa estimativa multiclasse, o protocolo pode falhar ao selecionar as rotas, seja alocando largura de banda insuficiente para vídeos de alta complexidade, seja desperdiçando recursos ao superdimensionar rotas para vídeos de baixa complexidade.

2.4.1 Modelos de Estimativa de Taxa de Bits

Além dos modelos tradicionais de estimativa contínua da taxa de bits, pesquisas mais recentes têm incorporado abordagens de classificação multiclasse para categorizar vídeos de acordo com sua complexidade, nível de movimento ou demanda de largura de banda. Essa abordagem permite separar os vídeos em classes como *baixa*, *média* e *alta* variação, oferecendo maior interpretabilidade para sistemas adaptativos.

Do ponto de vista teórico, Ting e Witten [47] discutem questões fundamentais da classificação multiclasse, destacando que muitos algoritmos de aprendizado são originalmente binários e requerem técnicas de decomposição para lidar com múltiplas

categorias. Os autores analisam diferentes estratégias de decomposição, como *one-vs-one* e *one-vs-all*, bem como os impactos dessas abordagens na propagação de erros e na ambiguidade entre classes.

Complementarmente, Rifkin e Klautau [41] apresentam uma análise detalhada do método *one-vs-all*, investigando seu desempenho, escalabilidade e custo computacional em problemas de classificação multiclasse. Os autores demonstram que, apesar de sua simplicidade, o método pode alcançar desempenho competitivo quando comparado a abordagens mais complexas, especialmente em cenários com número reduzido de classes e quando há necessidade de decisões interpretáveis.

Os resultados indicam que o *one-vs-all* é particularmente adequado para aplicações nas quais as classes possuem significado semântico claro e impacto direto na tomada de decisão, além de apresentar baixo custo computacional. Essas características tornam a abordagem alinhada aos requisitos deste trabalho, que busca classificar a variação da taxa de bits em níveis discretos (*baixa*, *média* e *alta*) com o objetivo de subsidiar mecanismos de roteamento adaptativo em redes com recursos restritos.

No domínio de vídeo, Li et al. [29] propõem um modelo baseado em aprendizado profundo para classificar vídeos segundo o nível de movimento, evidenciando que a separação multiclasse contribui para uma representação mais adequada da complexidade visual. Abordagens semelhantes também são observadas em estudos de Qualidade de Experiência, como o trabalho de Yamagishi e Hayashi [50], que utilizam classificadores para mapear métricas objetivas em categorias subjetivas de qualidade percebida.

A pesquisa sobre controle adaptativo de taxa de bits passou a incorporar de forma significativa técnicas de aprendizado de máquina, com o objetivo de melhorar a alocação de largura de banda e garantir maior estabilidade na transmissão. Um marco importante nesse contexto é a proposta de Souihi et al. [5], que introduzem o método Q2ABR. Esse modelo utiliza Aprendizado por Reforço (Reinforcement Learning) (RL) para otimizar

a seleção de representações de vídeo com base na Qualidade de Experiência (QoE) do usuário, avaliada por meio da métrica Pontuação Média de Opinião (Mean Opinion Score) (MOS).

A solução Q2ABR demonstra elevada eficácia ao otimizar a utilização da largura de banda e proporcionar maior estabilidade na experiência do usuário, conforme evidenciado pela melhoria de métricas objetivas de desempenho da transmissão e pelo aumento dos valores médios de Qualidade de Experiência (QoE), medidos por meio da métrica MOS, mesmo sob condições de rede variáveis [5].

Com a evolução das redes de comunicação e o aumento da variabilidade nas condições dos canais, novas abordagens baseadas em aprendizado profundo passaram a ser exploradas. Zhu et al. [52] propõem uma solução baseada em redes neurais artificiais (Redes Neurais Artificiais (Artificial Neural Networks) (RNA)) para a estimativa de estado em sistemas não lineares com restrições de taxa de bits. A proposta incorpora um mecanismo de alocação adaptativa de taxa de bits combinado a um modelo supervisionado de aprendizado profundo, permitindo que o estimador se ajuste ao ruído e às condições do canal em tempo real.

O método supera limitações presentes em abordagens tradicionais ao adaptar dinamicamente os parâmetros de estimativa, resultando em maior precisão e eficiência em cenários de transmissão com largura de banda limitada.

A principal limitação de abordagens recentes baseadas em aprendizado profundo reside na dependência de elevado poder computacional para processar grandes volumes de dados e realizar estimativas de taxa de bits em tempo real. Modelos desse tipo frequentemente envolvem milhares ou milhões de parâmetros, exigindo grande número de operações por inferência e, em muitos casos, o uso de hardware especializado, como GPUs, o que dificulta sua aplicação em ambientes com recursos limitados.

Em contraste, métodos de classificação supervisionada tradicionais apresentam

complexidade computacional significativamente menor, operando com número reduzido de parâmetros e demandando apenas centenas de operações por decisão. Na prática, isso se traduz em tempos de inferência da ordem de microssegundos a poucos milissegundos, mesmo em plataformas com capacidade computacional restrita, como dispositivos IoT ou gateways de rede.

Com o aumento da demanda por transmissões de vídeo em alta definição e 4K, cujas taxas de bits podem ultrapassar dezenas de megabits por segundo, torna-se crítica a necessidade de mecanismos de decisão rápidos e eficientes para o gerenciamento da largura de banda. Diante desse cenário, este trabalho propõe o uso de métodos de classificação supervisionada para gerar modelos de baixo custo computacional, capazes de classificar a variação da taxa de bits em tempo quase real, viabilizando sua integração a mecanismos de roteamento adaptativo em ambientes com recursos limitados.

2.4.2 Algoritmos de AutoML para Classificação Supervisionada

A utilização de técnicas de *Automated Machine Learning* (AutoML) tem crescido significativamente, especialmente em cenários que envolvem grande número de hiperparâmetros, múltiplos algoritmos candidatos e necessidade de otimização automática de desempenho [17, 22]. Nesse contexto, o *Tree-based pipeline Optimization Tool* (TPOT) destaca-se como uma ferramenta que emprega algoritmos genéticos para otimizar automaticamente os modelos completos de aprendizado supervisionado [36].

O TPOT funciona explorando um grande espaço de combinações possíveis entre algoritmos de pré-processamento, seleção de características e classificadores, além de seus respectivos hiperparâmetros [36]. Para avaliar cada modelo candidato, o TPOT utiliza validação cruzada (*k-fold cross-validation*) [35], na qual o conjunto de dados é particionado em k subconjuntos aproximadamente iguais. Em cada iteração, $k - 1$ subconjuntos são usados para treinamento e o subconjunto restante é empregado para validação, garantindo que todas as partes do conjunto de dados sejam utilizadas

tanto para treino quanto para teste em algum momento. A média obtida ao longo das k execuções fornece uma estimativa de desempenho menos dependente de uma única divisão dos dados, reduzindo variâncias decorrentes de amostragens particulares e diminuindo o risco de superestimar ou subestimar a real capacidade do modelo [25]. Assim, o processo torna-se estatisticamente mais robusto e confiável. Cada indivíduo da população representa um modelo, o qual é avaliado segundo a métrica definida pelo usuário (por exemplo, *accuracy* ou *balanced accuracy*) [36].

A seleção de modelos e hiperparâmetros ocorre de forma evolutiva: os melhores modelos, segundo a métrica de validação, são selecionados para reprodução; operadores genéticos — como *mutation* e *crossover* — criam novas combinações de modelos e configurações, permitindo a exploração contínua do espaço de busca [12, 18]. Dessa forma, o TPOT testa, combina e aprimora diferentes algoritmos clássicos de classificação supervisionada, convergindo gradualmente para o modelo com melhor desempenho [36].

Durante a otimização, o TPOT avalia várias classes de algoritmos tradicionais, incluindo árvores de decisão, métodos de conjunto, máquinas de vetor de suporte, métodos bayesianos, regressão logística, k -vizinhos mais próximos e modelos baseados em gradiente. Cada um desses métodos contribui para a busca evolutiva, oferecendo soluções com diferentes estruturas, complexidades e vieses, o que permite à ferramenta selecionar automaticamente o modelo mais adequado ao conjunto de dados em análise.

2.4.2.1 Árvores de Decisão e Métodos Baseados em Conjuntos

As árvores de decisão constituem os blocos fundamentais para diversos classificadores usados pelo TPOT. Elas particionam o espaço das características por meio de regras hierárquicas baseadas em atributos, formando um modelo interpretável e de baixo custo computacional [39]. Além das árvores individuais, o TPOT avalia métodos baseados em conjunto, como *Random Forest* e *Extra Trees*. O *Random Forest*, introduzido por Breiman [16], constrói múltiplas árvores independentes a partir de amostras aleatórias

e combina suas previsões por votação, reduzindo a variância e mitigando o problema de overfitting. Já os classificadores *Extra Trees* (*Extremely Randomized Trees*) utilizam divisões totalmente aleatórias nos nós, o que aumenta a diversidade entre as árvores e pode melhorar o desempenho em tarefas ruidosas [21]. Esses métodos são amplamente explorados pelo TPOT devido à sua robustez, escalabilidade e capacidade de capturar interações complexas entre atributos.

2.4.2.2 Support Vector Machines

O TPOT também inclui modelos baseados em *Support Vector Machines* (SVM), introduzidos por Cortes e Vapnik [14]. O SVM busca encontrar o hiperplano ótimo que maximiza a margem entre classes, proporcionando elevada capacidade de generalização. A ferramenta testa diferentes variantes, como SVM linear e SVM com kernel RBF, que permitem lidar tanto com dados linearmente separáveis quanto com padrões complexos não lineares no espaço das características. Métodos baseados em kernel são particularmente úteis em conjuntos de dados de alta dimensionalidade, sendo frequentemente selecionados quando a estrutura do problema favorece fronteiras de decisão mais refinadas.

2.4.2.3 Regressão Logística e Naive Bayes

A regressão logística é outro algoritmo avaliado pelo TPOT devido à sua simplicidade, interpretabilidade e desempenho consistente em problemas lineares [23]. Como modelo probabilístico, ela estima a probabilidade das classes a partir de uma combinação linear das características, sendo uma ferramenta tradicional em tarefas de classificação multiclasse. Já o *Naive Bayes* é um classificador probabilístico baseado no teorema de Bayes, que assume independência condicional entre atributos [34]. Apesar de sua simplicidade, o Naive Bayes apresenta bom desempenho em problemas com muitas dimensões e é extremamente rápido, tornando-o uma opção valiosa nos ciclos de avaliação do TPOT.

2.4.2.4 *K-Nearest Neighbors*

O TPOT também avalia o algoritmo *K-Nearest Neighbors* (KNN), introduzido por Cover e Hart [15]. O KNN classifica cada instância com base nas classes dos seus k vizinhos mais próximos, definidos por uma métrica de distância. Esse método não assume nenhuma distribuição prévia e é eficaz em problemas onde a relação entre atributos é altamente não linear. Entretanto, por ser computacionalmente custoso em grandes volumes de dados, tende a ser selecionado pelo TPOT apenas quando seu desempenho se destaca significativamente sobre alternativas mais eficientes.

2.4.2.5 Métodos Baseados em Gradiente

Outra categoria de modelos explorados pelo TPOT são os métodos baseados em gradiente, como *Gradient Boosting Machines* (GBM) e *XGBoost*. Esses modelos constroem uma sequência de árvores fracas, onde cada nova árvore corrige erros residuais das anteriores [19]. Tais métodos apresentam forte capacidade de modelagem em dados complexos, sendo com frequência os melhores desempenhos em competições de ciência de dados. O TPOT testa diferentes variantes desses algoritmos, ajustando automaticamente hiperparâmetros como profundidade das árvores, taxa de aprendizado e número de estimadores.

2.4.3 Framework FITPATH para Roteamento Adaptativo

O FITPATH é um framework de roteamento adaptativo multipath proposto para otimizar a transmissão de fluxos multimídia em redes com características dinâmicas e recursos limitados, como redes IoT. Conforme apresentado por Bhering et al. [8], o FITPATH tem como objetivo selecionar, em tempo quase real, os caminhos de transmissão mais adequados para cada fluxo de vídeo, levando em consideração tanto as condições instantâneas da rede quanto os requisitos de taxa de bits e sensibilidade à qualidade dos fluxos.

Diferentemente de abordagens tradicionais de roteamento estático, o FITPATH adota um mecanismo de decisão adaptativo baseado em uma heurística iterativa. Esse mecanismo permite que as rotas de transmissão sejam continuamente ajustadas em resposta a variações na largura de banda disponível, no atraso, na taxa de perdas de pacotes e no nível de interferência entre fluxos concorrentes. Como resultado, o framework busca maximizar a eficiência da transmissão e melhorar a Qualidade de Experiência (QoE) percebida pelos usuários finais.

2.4.3.1 Heurística Iterativa de Seleção de Caminhos

A heurística do FITPATH opera por meio de ciclos sucessivos de monitoramento, avaliação e adaptação. Inicialmente, o framework mantém um conjunto de caminhos candidatos entre os nós de origem e destino, obtidos a partir de mecanismos de roteamento multipath. Esses caminhos são considerados alternativas potenciais para a transmissão dos fluxos de vídeo, não sendo assumidos como fixos ao longo do tempo.

Em cada iteração do processo heurístico, o estado da rede é avaliado a partir de métricas de desempenho relevantes, incluindo vazão estimada, atraso médio, taxa de perda de pacotes e interferência entre enlaces. Essas métricas refletem o comportamento dinâmico da rede e o impacto da coexistência de múltiplos fluxos concorrentes.

Com base nessas informações, o FITPATH calcula um custo heurístico para cada caminho candidato. Esse custo representa a adequação do caminho para transportar determinado fluxo de vídeo, considerando seus requisitos de taxa de bits e sua sensibilidade a degradações de desempenho. Caminhos com menor custo heurístico são priorizados ou selecionados para a transmissão, podendo ser utilizados de forma combinada no contexto de roteamento multipath.

Esse processo de avaliação não é definitivo. À medida que novas informações sobre o estado da rede são obtidas, os custos dos caminhos são recalculados e as decisões de roteamento podem ser ajustadas. Dessa forma, caminhos que apresentem

degradação de desempenho podem ser descartados ou ter sua prioridade reduzida, enquanto alternativas mais favoráveis podem ser incorporadas. Esse comportamento caracteriza o caráter iterativo e adaptativo da heurística do FITPATH, permitindo decisões de baixo custo computacional adequadas a ambientes dinâmicos.

2.4.3.2 Estimativa Determinística de Desempenho com o MAPE

Um elemento central do funcionamento do FITPATH é a forma como as métricas de desempenho da rede são obtidas. Tradicionalmente, a avaliação de desempenho em redes é realizada por meio de simuladores estocásticos, como o ns-3, que incorporam elementos aleatórios para modelar o comportamento da rede. Embora precisos, esses simuladores apresentam elevado custo computacional e tempo de execução, tornando-se inviáveis para aplicações que exigem decisões em tempo real [9].

Para contornar essa limitação, Bhering et al. [10] incorporam ao FITPATH o Estimador de Desempenho Consciente de Multimídia (Multimedia-Aware Performance Estimator) (MAPE) (Multimedia-Aware Performance Estimator), um estimador determinístico projetado para fornecer previsões de desempenho em tempo real para fluxos multimídia transmitidos sobre redes multihop IEEE 802.11. O MAPE permite estimar, de forma eficiente e com baixo custo computacional, métricas como vazão, atraso e taxa de perda para cada fluxo individualmente, considerando a interferência entre múltiplos fluxos concorrentes.

As estimativas fornecidas pelo MAPE são utilizadas diretamente pela heurística do FITPATH como entrada para o cálculo do custo heurístico dos caminhos. Dessa forma, o MAPE atua como um módulo de estimação de desempenho, enquanto a heurística do FITPATH é responsável pela tomada de decisão de roteamento com base nessas estimativas.

2.4.3.3 Limitações do Modelo Atual

Apesar de sua eficiência, o MAPE assume que os fluxos de dados possuem taxa constante (CBR), não incorporando um analisador do tráfego efetivamente gerado pelos fluxos de vídeo. Essa suposição simplifica o processo de estimação, mas pode introduzir imprecisões em cenários nos quais os fluxos apresentam comportamento de taxa variável (VBR), caracterizado por rajadas e variações abruptas de taxa de bits.

Como consequência, a heurística do FITPATH pode tomar decisões de roteamento baseadas em estimativas que não refletem plenamente a variabilidade real do tráfego de vídeo, o que pode comprometer a adequação dos caminhos selecionados em determinados cenários. Essa limitação evidencia a necessidade de mecanismos capazes de caracterizar de forma mais precisa o comportamento dos fluxos de vídeo, especialmente no que se refere à variação da taxa de bits ao longo do tempo.

A Figura 2.6 ilustra a arquitetura geral do FITPATH, destacando seus principais componentes, incluindo o módulo de estimativa de desempenho (MAPE), o processo heurístico de seleção de caminhos e a interação entre os fluxos de vídeo e os múltiplos caminhos de transmissão.

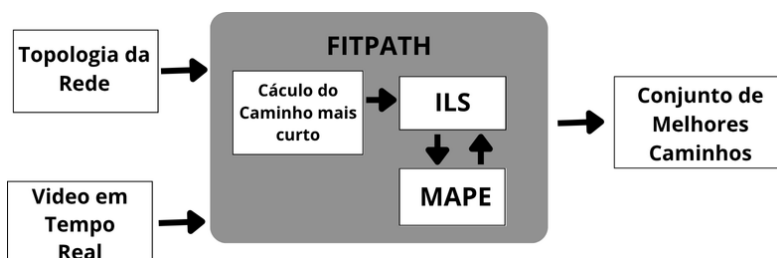


Figura 2.6 – Arquitetura do FITPATH para Seleção de Caminhos Adaptativos, Adaptado: [6].

O processo de decisão inicia-se com a coleta de informações sobre o estado da rede e os requisitos dos fluxos de vídeo. Em seguida, o algoritmo heurístico avalia os caminhos disponíveis com base nas estimativas fornecidas pelo MAPE, selecionando aqueles mais adequados para a transmissão. Esse processo é repetido iterativamente,

permitindo ajustes dinâmicos nas rotas conforme ocorrem variações na taxa de bits dos fluxos ou nas condições da rede.

As estimativas de desempenho fornecidas pelo MAPE são fundamentais para o processo de roteamento adaptativo. No entanto, essas estimativas são baseadas na suposição de fluxos com taxa constante (CBR), uma vez que o estimador não incorpora um analisador do tráfego gerado pelos fluxos de vídeo. Essa limitação pode comprometer a qualidade das rotas selecionadas, especialmente em cenários com tráfego de taxa variável (VBR), caracterizado por rajadas e variações abruptas de taxa de bits, que não são plenamente capturadas pelo modelo atual.

3 PROPOSTA E DESENVOLVIMENTO

Propõe-se, neste trabalho, o desenvolvimento de uma estratégia de roteamento adaptativo para transmissão de vídeo em tempo real, incorporando um classificador multiclasse da variação da taxa de bits como nova métrica de suporte à decisão. A proposta tem como objetivo aprimorar a eficiência na seleção de rotas ao fornecer ao mecanismo de roteamento informações mais precisas sobre o comportamento dos fluxos de vídeo, permitindo estimativas mais confiáveis das condições de rede necessárias para sua transmissão.

O modelo desenvolvido integra a classificação da variação da taxa de bits ao protocolo FITPATH, utilizando essa informação como parâmetro adicional para a escolha adaptativa de caminhos, conforme ilustrado na Figura 3.1. Ao classificar cada janela de vídeo em uma dentre doze classes associadas a diferentes tipos de cena, o sistema passa a identificar perfis distintos de comportamento do tráfego, relacionados à variabilidade temporal e espacial do conteúdo e à correspondente demanda de largura de banda.

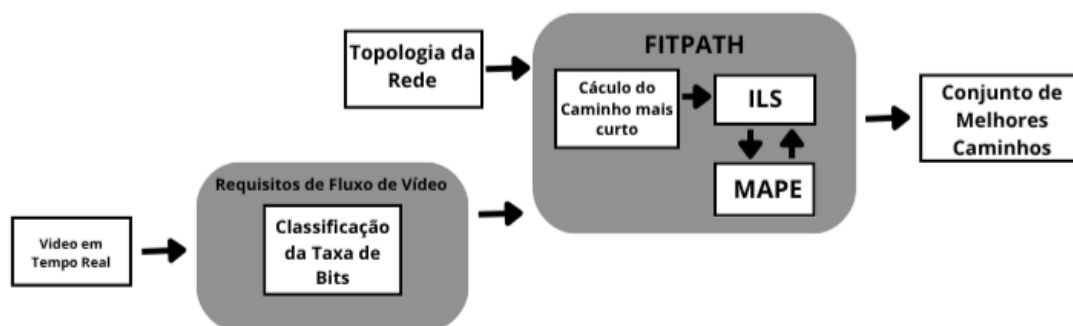


Figura 3.1 – Proposta da Arquitetura do FITPATH para Seleção de Caminhos Adaptativos. Adaptado de [7].

A integração do classificador ao FITPATH é realizada de forma complementar ao estimador MAPE, sem modificar a estrutura original do framework. No funcionamento original, o MAPE estima o desempenho dos caminhos assumindo fluxos com taxa

constante (CBR), a partir de uma taxa média previamente definida como requisito do fluxo de vídeo.

Na proposta deste trabalho, a classe atribuída a cada janela de vídeo é utilizada como informação adicional de entrada para o processo de decisão do FITPATH, permitindo ajustar os requisitos de taxa de bits considerados pelo MAPE. Cada classe representa um perfil específico de comportamento do tráfego, refletindo padrões típicos de variação da taxa de bits e diferentes níveis de sensibilidade a atraso, perda de pacotes e flutuações de largura de banda.

Com base nessas estimativas ajustadas, a heurística ILS do FITPATH calcula o custo dos caminhos disponíveis de forma mais aderente ao comportamento real de fluxos de taxa variável (VBR), priorizando aqueles que apresentam maior compatibilidade entre a capacidade estimada da rede e o perfil de tráfego associado à classe de cena identificada. Esse mecanismo permite diferenciar fluxos com taxas médias semelhantes, porém com padrões de variação distintos, reduzindo decisões inadequadas de roteamento em cenários com rajadas de tráfego e contribuindo para a manutenção da Qualidade de Serviço (QoS) durante a transmissão de vídeo em tempo real.

3.1 Modelagem para Classificação dos Fluxos de Vídeo

O modelo de classificação dos fluxos de vídeo desenvolvido neste trabalho é fundamentado em um processo sequencial composto pelas etapas de extração, consolidação, preparação e aplicação das métricas extraídas a partir dos *traces* de cada vídeo. Inicialmente, para cada vídeo analisado, são reconstruídos os frames a partir dos registros temporais presentes no arquivo de *trace*, de modo que o valor acumulado dos pacotes em cada intervalo correspondente a um frame seja somado. Esse procedimento permite representar o vídeo como uma série temporal de intensidade de tráfego, associada indiretamente à variação da movimentação da cena.

Em seguida, os dados são normalizados por meio do *RobustScaler*. A escolha dessa técnica de normalização é motivada pelas características do conjunto de dados analisado, que apresenta distribuições assimétricas e a presença de valores extremos decorrentes de rajadas de tráfego e variações abruptas típicas de fluxos de vídeo com taxa variável (VBR). Diferentemente de métodos como o *StandardScaler* ou o *MinMaxScaler*, que são fortemente influenciados por valores extremos, o *RobustScaler* realiza o escalonamento com base na mediana e no intervalo interquartil, reduzindo o impacto de outliers e preservando melhor as relações relativas entre as janelas temporais analisadas.

Essa escolha contribui para uma representação mais estável e informativa das métricas estatísticas extraídas dos *traces* de vídeo, favorecendo o processo de aprendizado do classificador e a identificação de padrões consistentes de movimentação entre diferentes tipos de cena.

A Figura 3.2 apresenta uma visão geral do modelo de classificação adotado neste trabalho, sintetizando de forma estruturada as principais etapas que compõem o processo de modelagem dos fluxos de vídeo. O objetivo desse modelo é transformar as métricas extraídas dos *traces* em informações úteis para tomada de decisão, permitindo que o sistema determine, de forma automática, a classe de movimentação correspondente a cada janela temporal do vídeo.

O fluxo inicia-se com o conjunto de *features* derivadas da série temporal de 60 quadros, representadas em bits por frame. Essas métricas capturam diferentes aspectos estatísticos da movimentação do vídeo, como variação temporal, presença de picos, intensidade das oscilações e estabilidade local. Em seguida, os dados são normalizados por meio do *RobustScaler*, técnica robusta a outliers e adequada para distribuições assimétricas — características frequentemente observadas em fluxos de vídeo com movimentação complexa.

Após o pré-processamento, o vetor de atributos é utilizado no processo de

avaliação conduzido pelo AutoML. O TPOT explorou diversos modelos candidatos e identificou o *ExtraTreesClassifier* como o modelo que melhor conseguiu separar as classes do conjunto de dados. Aqui, o objetivo não é empregar o modelo para predição operacional, mas sim verificar se o conjunto de *features* e a estrutura do dataset são suficientemente informativos para permitir que um classificador aprenda padrões consistentes. Dessa forma, o desempenho alcançado funciona como um indicador da qualidade e da adequação do dataset ao problema de classificação proposto.

Por fim, o modelo ilustrado no fluxograma 3.2 fornece evidências empíricas sobre a capacidade discriminativa das métricas extraídas. Essa avaliação orienta a validação do dataset e confirma sua viabilidade para integração com sistemas adaptativos, como o FITPATH, que poderão utilizar o nível de movimentação do vídeo como parâmetro para seleção de rotas e ajustes dinâmicos de taxa de bits.

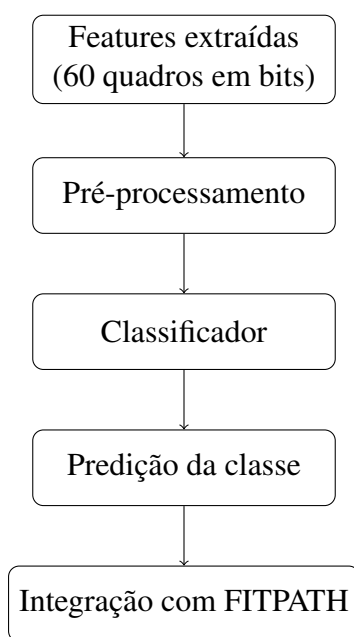


Figura 3.2 – modelo final selecionado pelo AutoML (TPOT).

As métricas extraídas de cada janela temporal são organizadas em um vetor de características que representa o comportamento estatístico do fluxo de vídeo naquele

intervalo. Cada vetor corresponde a uma janela específica e constitui a unidade básica utilizada no processo de classificação. As características que compõem esse vetor estão sintetizadas na Tabela 3.1.

Conforme apresentado na Tabela 3.1, o vetor de características é composto por métricas de tendência central e dispersão da taxa de bits por frame, como média, desvio padrão e coeficiente de variação, bem como por valores extremos e amplitude (máximo, mínimo e amplitude). Adicionalmente, são incluídas métricas robustas e de variabilidade local, como o intervalo interquartil (IQR), a média das diferenças absolutas (MAD) e o número de variações abruptas (*spikes*), que capturam a presença de rajadas típicas de fluxos de vídeo com taxa variável.

O vetor também incorpora descritores da forma da distribuição, como curtose e assimetria, além de percentis elevados, como o percentil 90 (P90), que refletem comportamentos extremos do tráfego associados a cenas de maior movimentação. Em conjunto, essas métricas permitem caracterizar de forma abrangente a dinâmica temporal e a variabilidade da taxa de bits, possibilitando a identificação de padrões específicos associados a diferentes tipos de cena e perfis de movimentação.

Para o desenvolvimento do modelo de classificação supervisionada, foi utilizado o *DCSASS Dataset*, assim como [26], composto por vídeos pertencentes a categorias distintas como *Normal*, *Accident*, *Violence*, *Burglary*, entre outras. Cada vídeo possui um rótulo explícito definido pelo próprio nome do arquivo, o que possibilita a construção de um modelo multiclasse sem dependência de heurísticas externas ou classificações baseadas em limiares. Assim, o nome da classe do vídeo foi utilizado como variável alvo (*label*) em todas as etapas do aprendizado supervisionado.

Diferentemente de abordagens baseadas em *motion score* ou classificações heurísticas por limiares, o presente trabalho não utiliza qualquer métrica sintetizada para determinar as classes. A classificação depende exclusivamente dos rótulos fornecidos

pelo dataset, permitindo que o modelo aprenda automaticamente padrões estatísticos que distinguem cada categoria. Essa abordagem assegura consistência metodológica entre a etapa de extração das características e o processo de rotulagem, além de garantir que o classificador capte diretamente a variabilidade intrínseca presente em cada classe do DCSASS.

A modelagem para classificação dos fluxos de vídeo foi conduzida em etapas sequenciais que abrangem desde a preparação dos vídeos até a integração com o MAPE usado no FITPATH. A seguir são apresentados os detalhes de cada uma das etapas. O modelo de classificação foi treinado, validado e testado utilizando exclusivamente o DCSASS Dataset. A aplicação do classificador no contexto do roteamento adaptativo, por sua vez, ocorre sobre fluxos de vídeo distintos, representados por meio de *traces* de tráfego extraídos de vídeos codificados com os mesmos parâmetros e no mesmo padrão de compressão.

A generalização do modelo é viabilizada pelo fato de que o classificador não opera diretamente sobre o conteúdo visual dos vídeos, mas sobre vetores de características estatísticas derivados do comportamento do tráfego gerado. Assim, desde que os fluxos de vídeo apresentem padrões de codificação e variabilidade temporal semelhantes, e sejam representados no mesmo espaço de atributos definido neste trabalho, o modelo pode ser aplicado a novos conjuntos de dados sem necessidade de re-treinamento.

3.1.0.1 Reprodutibilidade e Condições de Generalização

A reprodutibilidade do presente trabalho é garantida pelo uso de um dataset público (DCSASS), pela definição explícita do modelo de processamento e pela descrição detalhada das etapas de extração de *traces*, segmentação temporal, cálculo das métricas estatísticas e treinamento do modelo de classificação. Dessa forma, a repetição dos experimentos utilizando o mesmo conjunto de dados e as mesmas configurações resulta em desempenhos equivalentes, assegurando a reprodutibilidade metodológica da proposta.

Quanto à generalização, destaca-se que o classificador não opera diretamente sobre o conteúdo visual ou semântico dos vídeos, mas sobre vetores de características estatísticas derivados do comportamento do tráfego gerado. Assim, a aplicação do modelo a novos conjuntos de dados é viável desde que os fluxos de vídeo apresentem características compatíveis com o domínio considerado neste trabalho, incluindo o uso do mesmo padrão de codificação, a extração de *traces* no mesmo formato e a representação dos dados no mesmo espaço de atributos.

Nessas condições, o modelo pode ser aplicado a fluxos de vídeo distintos sem necessidade de re-treinamento, uma vez que a similaridade entre os dados de treinamento e de aplicação reside na equivalência estatística das características extraídas, e não na identidade do conteúdo visual ou das classes semânticas originais dos vídeos.

3.1.1 Geração dos Traces com MP4Trace

A primeira etapa do modelo proposto consiste na extração sistemática das informações estruturais de cada quadro dos vídeos utilizados no experimento. Essa etapa é responsável por transformar arquivos de vídeo no formato MP4 em arquivos de *trace* padronizados, contendo, para cada quadro, seu índice sequencial, o tamanho comprimido em bytes e o respectivo carimbo temporal. Esses dados constituem a base para todas as análises posteriores, incluindo reconstrução temporal, cálculo da taxa instantânea de bits e extração de métricas estatísticas (Figura 3.3).

Para a realização dessa tarefa, foi desenvolvido um script em Python que automatiza a varredura de todos os vídeos presentes no diretório de entrada e executa o MP4Trace em modo estruturado. O processo assegura robustez, reprodutibilidade e padronização, garantindo que todos os vídeos sejam tratados de maneira homogênea.

O script percorre recursivamente o diretório contendo os arquivos MP4 (`ROOT_DIR`), identifica automaticamente todos os vídeos e executa a rotina de extração para cada um deles. Para cada arquivo encontrado, o MP4Trace é invocado por meio do Windows

Subsystem for Linux (WSL), permitindo a utilização nativa de ferramentas de análise multimídia em ambiente Linux.

A execução é realizada com o comando:

```
mp4trace -f -s video.mp4 <ip_destino> <porta_udp> > st_video.txt
```

Figura 3.3 – Comando utilizado para gerar o arquivo de *trace* a partir de um vídeo MP4 utilizando o MP4Trace. Fonte: Autor.

Esse comando gera um arquivo de *trace* contendo, linha a linha, os campos referentes ao índice do quadro, seu tamanho comprimido e o timestamp correspondente, os quais serão posteriormente utilizados nas etapas de análise e processamento do modelo.

3.1.1.1 Estrutura do Arquivo de *Trace*

Cada arquivo de *trace*, como ilustrado na Figura 3.4, possui três colunas fundamentais: índice do frame, tamanho do pacote em bytes e timestamp. Inicialmente, os valores são lidos e convertidos para tipos numéricos apropriados. Em seguida, a série temporal é reconstruída quadro a quadro por meio de uma agregação determinística baseada no índice do frame.

Nesse processo, todos os registros associados a um mesmo frame são combinados utilizando operações de agregação simples: os tamanhos dos pacotes são somados, de modo a obter o tamanho total do frame em bytes, enquanto o menor timestamp observado é atribuído como referência temporal do frame. Essa etapa não constitui um método de agrupamento estatístico ou de aprendizado não supervisionado, mas sim uma operação de *group-by* necessária para a reconstituição fiel da sequência temporal original da transmissão de vídeo.

```
<id_do_frame> <tamanho_em_bytes> <timestamp_em_segundos>
```

Para ilustrar como esse formato se manifesta na prática durante a extração dos dados, a Figura 3.4 apresenta um trecho real obtido a partir dos experimentos conduzidos neste trabalho.

```
0 3388 0.000000
1 39    0.033333
2 318   0.066667
3 41    0.100000
4 221   0.133333
```

Figura 3.4 – Exemplo real de trecho de um arquivo de *trace* contendo o índice do frame, seu tamanho em bytes e o timestamp correspondente. Fonte: Autor.

3.1.2 Processamento dos Traces e Extração das Features Estatísticas

Após a geração dos arquivos de *trace*, que contêm informações associadas a cada quadro dos vídeos, realiza-se o processamento desses arquivos e a extração das métricas estatísticas que serão utilizadas como atributos no modelo de classificação. Nessa etapa, a série temporal de bytes por quadro é transformada em um conjunto de características numéricas capazes de capturar o comportamento dinâmico do fluxo, especialmente no que se refere à variação da taxa de bits ao longo do tempo.

A reconstrução da série temporal é necessária porque a ferramenta `MP4Trace` pode gerar múltiplos registros para um mesmo quadro, dependendo do codificador utilizado e do formato do contêiner MP4, particularmente em fluxos comprimidos com H.264. Assim, os registros correspondentes a um mesmo quadro são agregados de forma determinística, permitindo a obtenção de uma representação única por frame.

Após essa reconstrução, o primeiro quadro de cada série temporal é removido, uma vez que, em diversos vídeos, ele apresenta um tamanho atípico associado ao carregamento inicial do fluxo ou à presença de um *I-frame* de abertura, o que poderia distorcer as métricas estatísticas calculadas posteriormente.

Além disso, para padronizar o tamanho de entrada do modelo de classificação, apenas os primeiros 60 quadros de cada vídeo são considerados, conforme definido em:

$$FRAMES_LIMITE = 60$$

Esse procedimento assegura que todos os vídeos sejam representados por vetores de mesma dimensão, independentemente de sua duração original. O vetor resultante, composto pelos 60 quadros selecionados, é tratado como uma única janela temporal, a partir da qual são extraídas diversas métricas estatísticas capazes de descrever a dinâmica temporal da série. Todas as métricas são calculadas com base nos valores de bytes por quadro.

3.1.3 Métricas extraídas dos fluxos de vídeo

A partir dessa série, um conjunto de métricas estatísticas é calculado com o objetivo de capturar diferentes aspectos da dinâmica temporal do vídeo, incluindo intensidade média, variabilidade, irregularidade, presença de picos e estrutura estatística avançada da distribuição. As métricas estatísticas adotadas neste trabalho não foram selecionadas de forma arbitrária. Sua escolha foi guiada por critérios metodológicos alinhados ao objetivo de caracterizar a variação temporal da taxa de bits em fluxos de vídeo com codificação variável (VBR). Em particular, foram priorizadas métricas capazes de capturar diferentes dimensões do comportamento do tráfego, incluindo intensidade média, dispersão, irregularidade temporal, presença de rajadas e propriedades da distribuição estatística.

Adicionalmente, a seleção considerou métricas amplamente empregadas na literatura de análise de séries temporais, processamento de sinais e caracterização de tráfego multimídia, bem como seu baixo custo computacional, fator relevante para aplicações associadas a mecanismos de roteamento adaptativo. Dessa forma, o conjunto

de métricas adotado busca um equilíbrio entre poder discriminativo, interpretabilidade e viabilidade computacional, sendo adequado à identificação de padrões associados a diferentes níveis e tipos de movimentação de cena.

Essas métricas foram selecionadas devido à sua capacidade de distinguir conteúdos com diferentes níveis de movimentação, permitindo que o classificador aprenda padrões característicos de fluxos de complexidade temporal. A seguir, cada métrica é detalhada individualmente.

3.1.3.1 Média (Média)

Representa o valor médio de bits por quadro ao longo dos 60 frames analisados. Essa coluna da tabela quantifica a intensidade global do tráfego de cada classe. Vídeos como *Fighting* e *Shoplifting*, por exemplo, apresentam médias elevadas, indicando alta demanda de bits devido à movimentação intensa.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

3.1.3.2 Desvio padrão (Desvio Padrão)

A coluna de desvio padrão indica o grau de dispersão dos valores em torno da média. Classes com cenas mais caóticas, como *Explosion* ou *Robbery*, exibem desvios maiores.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

3.1.3.3 Coeficiente de variação (Coef. Var)

Razão entre o desvio padrão e a média, normalizando a variabilidade em relação ao nível médio de bits. Na análise exploratória realizada neste trabalho, essa métrica

mostrou-se particularmente discriminativa para classes como *Robbery*, *Explosion* e *Fighting*, que apresentam elevada oscilação relativa da taxa de bits ao longo das janelas analisadas.

$$CV = \frac{\sigma}{\mu}$$

3.1.3.4 Máximo e mínimo (**Máx, Mín**)

O maior e o menor valor de bits por frame. Classes como *Explosion* e *Burglary* apresentam máximos elevados, associados a picos de movimento; o mínimo reflete períodos de pouca mudança na cena.

3.1.3.5 Amplitude (**Amplitude**)

Diferença entre o valor máximo e mínimo, presente na coluna correspondente da tabela. Essa métrica quantifica a oscilação total da série temporal do vídeo.

$$\text{Amplitude} = x_{\max} - x_{\min}$$

3.1.3.6 IQR — Intervalo interquartil (**IQR**)

Representa a diferença entre os quartis 75% e 25%. Na tabela, vídeos como *Shoptlifting* apresentam IQR elevado, refletindo variação interna significativa.

$$IQR = Q_3 - Q_1$$

3.1.3.7 Spikes (**Spikes**)

Número de transições abruptas entre quadros consecutivos, utilizando limiar de $T = 500 \times 8$ bits. Na tabela, classes como *Burglary*, *Robbery* e *Stealing* apresentam três

spikes, indicando maior instabilidade temporal.

$$\text{Spikes} = \sum_{i=2}^n \mathbb{I}(|x_i - x_{i-1}| > T)$$

3.1.3.8 Curtose (**Curtose**)

Mede o peso das caudas da distribuição, destacando presença de quadros com valores extremamente altos. Na tabela, *Burglary* possui curtose muito elevada, sugerindo picos intensos.

$$\text{kurt} = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \mu}{\sigma} \right)^4$$

3.1.3.9 Assimetria (**Assimetria**)

Quantifica o grau de inclinação da distribuição. Valores positivos indicam frequência maior de quadros leves com alguns picos muito altos — caso observado em classes como *Arson* e *Robbery*.

$$\text{skew} = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \mu}{\sigma} \right)^3$$

3.1.3.10 Diferença absoluta média (**MAD**)

Presente na coluna MAD da tabela, essa métrica captura a suavidade da série temporal. Valores altos indicam mudanças frequentes ou cortes rápidos.

$$\text{MAD} = \frac{1}{n-1} \sum_{i=2}^n |x_i - x_{i-1}|$$

3.1.3.11 Percentil 90 (P90)

Indica o nível acima do qual estão os 10% dos quadros mais pesados. Classes como *Shoplifting* e *Fighting* exibem P90 elevado, refletindo cenas com picos de movimentação.

3.1.3.12 Média global

Corresponde à média dos 60 valores da série original e fornece uma visão geral do comportamento temporal, funcionando como referência para escalonamento dos níveis de taxa de bits entre classes.

Conjuntamente, as métricas listadas na Tabela 3.1 caracterizam de forma abrangente o comportamento temporal das treze classes, permitindo identificar padrões de movimentação, variabilidade, irregularidade e intensidade dos picos, fornecendo assim uma base estatística sólida para a classificação automatizada dos vídeos.

3.1.3.13 Construção do Dataset Consolidado

Para cada arquivo `st_` processado, o conjunto de atributos descrito anteriormente é armazenado como uma linha de uma tabela. Essa tabela é então exportada para o arquivo:

Tabela 3.1 – Resumo das métricas extraídas dos vídeos a partir dos traces de transmissão.

Vídeo	Média	Desvio Padrão	Coef. Var.	Máx	Mín	Amplitude	IQR	Spikes	Curtose	Assimetria	MAD	P90
Abuse	269.43	1133.90	4.21	8828	14	8814	230.25	5	52.95	7.34	322.79	352.3
Abuse	278.35	1185.49	4.26	9213	14	9199	208.25	7	52.61	7.31	355.93	433.6
Abuse	261.27	1213.82	4.65	9463	14	9449	184.00	1	53.93	7.44	291.15	314.8
Abuse	273.52	1235.55	4.52	9642	14	9628	200.00	1	53.97	7.45	316.22	279.1
Abuse	296.92	1198.76	4.04	9297	14	9283	281.00	3	51.77	7.23	372.54	364.1

Esse arquivo representa o conjunto de dados utilizado no treinamento e na avaliação dos modelos de classificação. Para cada vídeo, ele reúne tanto os atributos estatísticos calculados a partir dos *traces* quanto o nome do arquivo já normalizado, permitindo sua associação posterior com a classe de movimentação correspondente.

Esses atributos constituem a base do processo de modelagem supervisionada, pois permitem que o classificador aprenda a distinguir diferentes padrões de complexidade visual com base exclusivamente no comportamento da taxa de bits ao longo do tempo.

3.1.4 Treinamento e Seleção Automática do Modelo com TPOT

Esta subseção é responsável pela construção, otimização e avaliação do modelo de classificação multiclasse utilizado para identificar o nível de movimentação de cada vídeo. Para esse fim, são empregadas técnicas de aprendizado supervisionado e ferramentas automatizadas de AutoML, com o objetivo de selecionar a melhor combinação de algoritmos e hiperparâmetros. O processo é conduzido de forma sistemática, garantindo rigor experimental e permitindo a reprodutibilidade completa dos resultados.

3.1.4.1 Preparação dos Dados de Treinamento e Teste

Antes de fazer o treinamento é necessário organizar os dados em conjuntos distintos para as etapas de treinamento e teste. Essa preparação passou a ser realizada de forma automatizada, garantindo que cada subconjunto contenha representações adequadas dos vídeos disponíveis. O procedimento envolve tanto a divisão do arquivo consolidado de atributos quanto a separação física dos respectivos arquivos de *trace*, mantendo a integridade e a correspondência entre os dados.

Para assegurar uma divisão equilibrada e evitar que vídeos de um mesmo tipo fiquem concentrados apenas em um dos grupos, empregou-se a técnica de divisão estratificada via `train_test_split`. Nesse processo, a coluna `video` foi utilizada como critério de estratificação, garantindo que cada tipo de conteúdo estivesse representado tanto no conjunto de treino quanto no conjunto de teste.

A divisão adotou os seguintes parâmetros:

$$test_size = 0.20, \quad random_state = 42$$

resultando em 80% dos vídeos destinados ao treinamento e 20% destinados à avaliação do modelo. Os arquivos gerados, `treino_features.csv` e `teste_features.csv`, reúnem as *features* e metadados necessários para a etapa de modelagem, servindo como base para a validação da capacidade discriminativa do conjunto de atributos.

Ainda na etapa de treinamento, os atributos são normalizados por meio do método `RobustScaler`. Esse procedimento realiza o escalonamento das features com base na mediana e no intervalo interquartil, reduzindo a influência de valores extremos e tornando os atributos mais comparáveis entre si. Essa escolha é particularmente adequada neste contexto, uma vez que as métricas extraídas apresentam magnitudes distintas (por exemplo, média, amplitude e curtose) e distribuições assimétricas, características comuns em fluxos de vídeo com taxa variável.

Além disso, diversos algoritmos avaliados pelo AutoML são sensíveis à escala dos dados, de modo que a normalização contribui para uma convergência mais estável dos modelos e reduz possíveis vieses durante o processo de aprendizado. O `scaler` final é armazenado em arquivo (`scaler_tpot.pkl`) para ser reutilizado posteriormente no modelo operacional integrado ao FITPATH.

Como entrada, esta etapa utiliza os arquivos `treino_features.csv` e `teste_features.csv`. Cada linha desses arquivos corresponde a um vídeo e contém:

- atributos estatísticos extraídos da série temporal de bits por quadro;
- o nome do vídeo já normalizado;
- metadados complementares utilizados para rastreabilidade.

A coluna contendo o nome do vídeo (`video`) é utilizada para representar o rótulo real de cada amostra (`label_video`). Cada vídeo possui uma classe distinta, totalizando 13 categorias, correspondentes aos 13 tipos de conteúdo vídeo presentes no dataset utilizado.

Para o treinamento, são removidas colunas auxiliares que não devem ser utilizadas como atributos:

$$\text{Exclusões} = \{video, class\ label_video\}.$$

O vetor de atributos (X) é formado por todas as demais colunas. Os rótulos (y) são obtidos diretamente de `label_video`.

3.1.4.2 Otimização Automática com TPOT

Agora etapa de treinamento, o TPOT foi responsável por explorar diferentes modelos de classificação, avaliar combinações distintas de algoritmos e hiperparâmetros, otimizar a métrica *accuracy* e selecionar, ao final, o modelo com melhor desempenho no conjunto de validação. Para este estudo, foram adotados os seguintes parâmetros: `generations = 300, population_size = 100, scoring = "accuracy", max_time_mins = 20 e n_jobs = -1`, permitindo a utilização de todos os núcleos de processamento disponíveis. A métrica de acurácia foi adotada como critério de otimização no processo de busca conduzido pelo TPOT. Embora a acurácia seja reconhecidamente sensível a cenários de forte desbalanceamento entre classes, sua utilização neste trabalho é justificada pelo perfil do conjunto de dados analisado, que apresenta distribuição moderadamente equilibrada entre as classes, sem a presença de categorias raras ou dominantes.

Além disso, o objetivo do uso do AutoML neste estágio é explorar o espaço de modelos e hiperparâmetros e verificar a capacidade discriminativa do conjunto de

atributos proposto, e não otimizar uma métrica específica associada a custos assimétricos entre classes. Nesse contexto, a acurácia fornece uma medida global adequada do desempenho do classificador, permitindo a seleção de um modelo base consistente.

Ressalta-se ainda que a acurácia é utilizada apenas como métrica de busca durante o processo automático de seleção de modelos, sendo complementada por análises adicionais de desempenho nas etapas subsequentes, de modo a fornecer uma avaliação mais abrangente do comportamento do classificador em cenário multiclasse.

Ao término dessa etapa, obteve-se toda a configuração necessária para materializar um modelo capaz de classificar os vídeos em 13 categorias distintas. Essas categorias não representam apenas rótulos semânticos, mas perfis diferenciados de comportamento do tráfego de vídeo, definidos a partir das métricas estatísticas extraídas dos *traces* gerados durante a transmissão simulada.

Cada classe corresponde a um padrão específico de movimentação de cena, refletido em características como intensidade média da taxa de bits, variabilidade temporal e ocorrência de picos de tráfego. Dessa forma, o modelo fundamenta sua decisão exclusivamente nas propriedades estatísticas do fluxo de dados, e não no conteúdo visual direto dos vídeos.

A distinção entre esses perfis pode ser observada na Figura 4.1, que apresenta a variação do volume médio de pacotes entre as classes, evidenciando diferenças consistentes no comportamento do tráfego associado a cada categoria. Os rótulos específicos das classes são discutidos em detalhes na seção de resultados.

3.1.5 Modelo Selecionado pelo AutoML (TPOT)

A Figura 3.5 ilustra o processo evolutivo empregado pelo TPOT no contexto do AutoML. Inicialmente, o algoritmo gera uma população de pipelines de aprendizado de máquina construídos de forma aleatória. Cada pipeline representa uma combinação válida

de etapas de pré-processamento, algoritmo de classificação e respectivos hiperparâmetros, compondo o espaço de busca do problema.

Esses pipelines iniciais são avaliados com base na métrica de acurácia, utilizada como função de aptidão. A partir dessa avaliação, é aplicado um mecanismo de seleção que combina elitismo e seleção por torneio. O elitismo garante a preservação dos melhores pipelines da geração atual, enquanto a seleção por torneio escolhe indivíduos candidatos à reprodução com base em seu desempenho relativo.

Os pipelines selecionados são então submetidos a operadores genéticos de cruzamento e mutação. O cruzamento combina partes de diferentes pipelines, enquanto a mutação introduz variações aleatórias, como a alteração de hiperparâmetros ou a substituição de algoritmos. Esse processo permite explorar de forma eficiente um espaço de busca combinatorial amplo, sem a necessidade de avaliação exaustiva de todas as combinações possíveis.

O ciclo de avaliação, seleção e reprodução é repetido ao longo das gerações, até que seja identificado um pipeline com desempenho superior. Ao final do processo, o TPOT exporta automaticamente o modelo de melhor desempenho encontrado, que, neste trabalho, corresponde à combinação do `RobustScaler` com o classificador `ExtraTrees`.

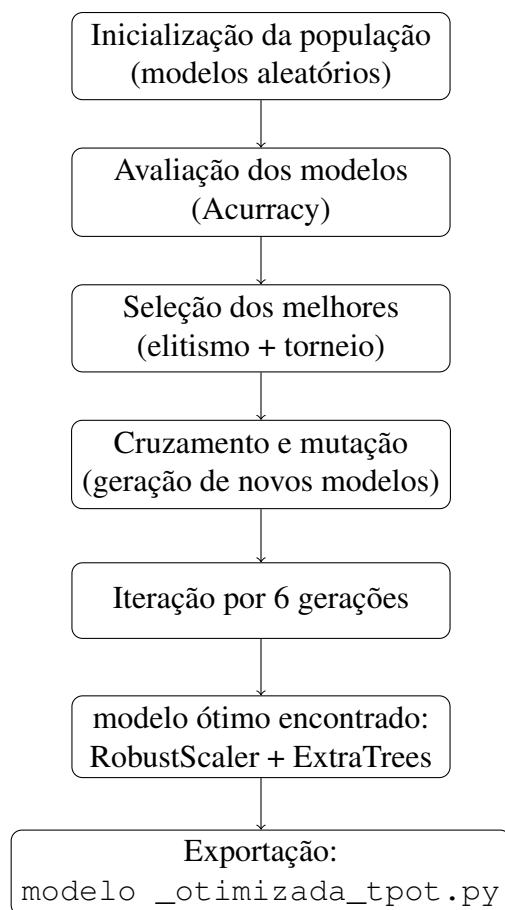


Figura 3.5 – Fluxograma do processo evolutivo do AutoML (TPOT).

Após seis gerações de evolução, o TPOT convergiu para o seguinte modelo final:

Listing 3.1 – modelo final obtido pelo TPOT. Fonte: Autor

```
1 RobustScaler()  
2     ExtraTreesClassifier(  
3         bootstrap = False,  
4         criterion = "entropy",  
5         max_features = 0.85,  
6         min_samples_leaf = 2,  
7         min_samples_split = 3,  
8         n_estimators = 100  
9 )
```

No processo de otimização automática, o TPOT avaliou um total de 4000 modelos distintos. Esse número decorre da combinação dos parâmetros definidos: uma população inicial de 100 indivíduos e trezentas gerações evolutivas adicionais, cada uma contendo 100 novos modelos avaliados, resultando em:

$$N = 1000 + (300 \times 100) = 4000 \text{ execuções.}$$

Cada indivíduo corresponde a uma configuração completa de pré-processamento e classificador, treinada e validada segundo a métrica accuracy a . Todas as execuções foram registradas no arquivo `tpot_full_log.csv`.

No presente trabalho, a função de aptidão utilizada pelo TPOT considera exclusivamente a métrica de acurácia, conforme definido pelo parâmetro `scoring`. Essa acurácia é calculada a partir de validação cruzada interna, sendo utilizada como valor escalar para ordenar os indivíduos da população durante o processo evolutivo.

A opção por uma única métrica de aptidão decorre da necessidade de manter um critério objetivo e computacionalmente eficiente para a seleção dos pipelines, característica intrínseca a algoritmos genéticos. Além disso, considerando que o problema abordado

é multiclasse, sem custos assimétricos entre categorias e com distribuição moderadamente equilibrada, a acurácia fornece uma medida global adequada para guiar a busca automática no espaço de soluções. Ressalta-se que essa métrica é empregada apenas como critério de otimização no AutoML, não substituindo análises complementares de desempenho realizadas posteriormente.

A seguir são apresentados os componentes selecionados e a justificativa técnica de cada um deles.

3.1.5.1 Pré-Processador

O pré-processador escolhido pelo TPOT foi o *RobustScaler*, que normaliza os dados utilizando estatísticas robustas aos outliers, como a mediana e o intervalo interquartil (IQR). Esse tipo de escalonamento mostra-se particularmente adequado para este trabalho, pois as séries temporais de bits por quadro apresentam picos abruptos que distorcem métodos baseados em média e desvio padrão; vídeos com movimentação intensa exibem distribuições assimétricas, caracterizadas por alta curtose e skewness, nas quais normalizadores como o *StandardScaler* podem introduzir viés; e, adicionalmente, o *RobustScaler* preserva as relações relativas entre as classes mesmo na presença de valores extremos. Dessa forma, o TPOT identificou que a natureza estatística dos fluxos demanda um pré-processamento robusto, o que tornou esse método superior aos demais ao longo das gerações evolutivas.

3.1.5.2 Classificador

O classificador final escolhido foi o ExtraTrees (*Extremely Randomized Trees*), um modelo conjunto baseado em árvores de decisão totalmente randomizadas. Sua seleção é coerente com o tipo de problema abordado neste trabalho, pois o modelo apresenta capacidade para capturar relações não lineares entre as métricas estatísticas extraídas dos vídeos, demonstra robustez diante de dados ruidosos ou não escalonados

— característica reforçada pelo uso prévio do *RobustScaler* —, alcança desempenho elevado em problemas multiclasse com grande variabilidade intra-classe, como ocorre nas 13 categorias de vídeos analisados, e ainda mantém baixo risco de *overfitting* devido à aleatorização extrema aplicada na divisão dos nós. A seguir, são descritos os hiperparâmetros otimizados pelo TPOT e seu papel no comportamento do modelo.

3.1.5.3 Hiperparâmetros do ExtraTrees e sua função

No modeloselecionado, o parâmetro `criterion = entropy` define que a impureza das divisões é calculada por entropia, em vez do índice de Gini. A entropia é mais sensível a pequenas diferenças na distribuição das classes, o que favorece problemas de alta granularidade, como o presente, composto por 13 categorias distintas. O parâmetro `max_features = 0.85` estabelece que, a cada divisão, 85% dos atributos são considerados; tal configuração mantém grande parte da informação disponível, ao mesmo tempo em que introduz aleatoriedade suficiente para reduzir o risco de *overfitting*. O hiperparâmetro `min_samples_leaf = 2` exige pelo menos duas amostras por folha, evitando divisões excessivamente específicas — como folhas contendo apenas uma amostra — e, assim, favorecendo a capacidade de generalização do modelo. De modo complementar, `min_samples_split = 3` determina que uma divisão só pode ocorrer quando há pelo menos três amostras, o que contribui para evitar divisões instáveis, especialmente relevantes em conjuntos de dados com variabilidade temporal.

Além disso, o parâmetro `bootstrap = False` indica que todas as amostras são utilizadas no treinamento de cada árvore, em vez de se aplicar o procedimento de *bagging*; em modelos extremamente randomizados, essa configuração tende a melhorar o desempenho por preservar a diversidade gerada pela aleatoriedade interna das divisões. Por fim, o hiperparâmetro `n_estimators = 100` define que o ensemble é composto por 100 árvores, número que representa um equilíbrio adequado entre estabilidade estatística, rapidez na inferência — requisito importante para o funcionamento do FITPATH — e

custo computacional moderado.

A escolha final do pipeline pelo TPOT é corroborada por evidências quantitativas extraídas dos próprios dados e do processo evolutivo. As estatísticas descritivas das features indicam distribuições assimétricas e presença de valores extremos, justificando o uso do RobustScaler. A seleção do ExtraTreesClassifier reflete a necessidade de capturar relações não lineares entre as métricas extraídas, evidenciada pelo melhor desempenho desse modelo em relação a alternativas avaliadas durante o AutoML.

Adicionalmente, o desempenho consistente obtido na validação cruzada interna, significativamente superior ao nível aleatório esperado em um cenário com 13 classes, indica boa capacidade de generalização. Por fim, a estabilidade do modelo frente a métricas associadas a ruído e variações abruptas sugere baixa sensibilidade às flutuações inerentes aos fluxos de vídeo com taxa variável.

3.1.5.4 Avaliação do Modelo

A consiste na aplicação do modelo treinado aos arquivos de *trace* produzidos para o conjunto de teste, com o objetivo de classificar cada vídeo em uma das 13 classes definidas e, posteriormente, consolidar essas informações em formatos adequados para análise, visualização e integração ao módulo MAPE do FITPATH. Essa etapa representa a operacionalização do classificador e a transformação dos dados brutos em informações gerenciais úteis para a tomada de decisão em sistemas adaptativos.

A avaliação do desempenho de modelos de classificação supervisionada é usada para verificar a confiabilidade dos resultados obtidos em tarefas de aprendizado de máquina e orientar a escolha do modelo mais adequado para cada aplicação. Esse processo permite analisar não apenas a acurácia do modelo, mas também sua capacidade de generalização para novos dados, ajudando a identificar problemas como *overfitting* ou *underfitting*. O *underfitting* ocorre quando o modelo é incapaz de aprender os padrões relevantes presentes nos dados de treinamento, geralmente devido a uma estrutura

excessivamente simples ou a um número insuficiente de parâmetros, resultando em baixo desempenho tanto nos dados de treino quanto nos dados de teste.

O conjunto de teste, por sua vez, é empregado exclusivamente para a avaliação definitiva do classificador já otimizado. Como esses dados não participam da fase de ajuste, eles fornecem uma estimativa imparcial da capacidade de generalização do modelo para novos vídeos. Dessa forma, a avaliação sobre o conjunto de teste representa o desempenho real do sistema em um cenário independente, caracterizando a medida final de desempenho do classificador.

A avaliação do desempenho do classificador é especialmente relevante no presente trabalho, uma vez que o objetivo consiste em distinguir entre treze classes de vídeos, cada qual associada a um padrão distinto de movimentação e demanda de taxa de bits. Nesse contexto, métricas globais — como acurácia, *balanced accuracy*, precisão macro, *recall* macro e *F1-score* macro — tornam-se essenciais, pois fornecem uma visão equilibrada do desempenho em cenários multiclasse e potencialmente desbalanceados.

A avaliação de modelos de classificação supervisionada é tradicionalmente baseada em medidas derivadas da matriz de confusão, que organiza os resultados preditos em quatro categorias fundamentais: verdadeiros positivos (VP), falsos positivos (FP), falsos negativos (FN) e verdadeiros negativos (VN). A partir dessas quantidades, é possível construir indicadores capazes de capturar não apenas o desempenho global do modelo, mas também seu comportamento por classe, especialmente em cenários desbalanceados [3, 20, 28, 43].

Dentre as métricas de avaliação mais empregadas na literatura de classificação supervisionada destacam-se a acurácia, a precisão, o *recall* e o *F1-score* [3, 20]. No presente trabalho, a adoção dessas métricas decorre de uma escolha metodológica alinhada às características do problema abordado, que envolve classificação multiclasse com 13 categorias, sem custos assimétricos entre classes e com distribuição moderadamente

equilibrada.

A acurácia fornece uma visão global do desempenho do classificador, enquanto as métricas de precisão, *recall* e *F1-score*, calculadas na forma macro, permitem avaliar de maneira equilibrada o comportamento do modelo em todas as classes, evitando que categorias específicas dominem a análise.

Outras métricas amplamente utilizadas na literatura, como AUROC e AUPRC, não foram priorizadas neste estudo por apresentarem menor interpretabilidade no contexto multiclasse adotado ou por serem mais indicadas em cenários binários ou com classes raras críticas. Métricas como o coeficiente Kappa de Cohen, a *Balanced Accuracy* e o Coeficiente de Correlação de Matthews (MCC), por sua vez, foram empregadas na etapa de avaliação final sobre o conjunto de teste, com o objetivo de fornecer uma análise complementar e mais robusta do desempenho do classificador.

3.1.5.5 Matriz de Confusão

A matriz de confusão fornece uma representação detalhada da distribuição dos acertos e erros do modelo, relacionando valores reais e valores preditos para cada classe [38]. A partir dela é possível extrair:

- **VP** (verdadeiros positivos) — amostras corretamente classificadas;
- **FP** (falsos positivos) — amostras preditas incorretamente como pertencentes a uma classe;
- **FN** (falsos negativos) — amostras de uma classe real preditas como outra classe;
- **VN** (verdadeiros negativos) — acertos relativos às demais classes.

3.1.5.6 Matriz de Confusão Absoluta

A matriz de confusão absoluta representa a distribuição dos acertos e erros do classificador em valores absolutos, relacionando as classes reais às classes preditas. Cada elemento M_{ij} da matriz indica o número de amostras pertencentes à classe real i que foram classificadas como pertencentes à classe j .

Essa representação permite identificar padrões específicos de confusão entre classes, sendo fundamental para a análise detalhada do desempenho do modelo em problemas de classificação multiclasse.

3.1.5.7 Matriz de Confusão Normalizada

A matriz de confusão normalizada é obtida a partir da matriz de confusão absoluta por meio da normalização de cada linha pelo número total de amostras da respectiva classe real. Dessa forma, cada elemento passa a representar a proporção de acertos ou erros associados a uma classe.

Essa normalização facilita a comparação entre classes com diferentes quantidades de amostras e fornece uma interpretação proporcional do desempenho do classificador em cada categoria.

3.1.5.8 Acurácia

A acurácia (*accuracy*) mede a proporção de previsões corretas em relação ao total de amostras classificadas. Ela é definida como:

$$\text{Acurácia} = \frac{\text{número de acertos}}{\text{número total de amostras}} \quad (3.1)$$

Em termos da matriz de confusão multiclasse, assume a forma:

$$\text{Acurácia} = \frac{\sum_{i=1}^k \text{TP}_i}{N}, \quad (3.2)$$

onde k é o número de classes, TP_i representa o número de verdadeiros positivos da classe i e N é o total de amostras avaliadas [20, 37]. Embora popular, a acurácia pode ser inadequada quando o conjunto de dados é desbalanceado, pois tende a privilegiar classes majoritárias [3]. Por isso, outras métricas complementares são indispensáveis.

3.1.5.9 Precisão

A precisão (ou *precision*) mede a proporção de predições positivas que são realmente corretas. Ela é definida como:

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (3.3)$$

onde VP representa os verdadeiros positivos e FP os falsos positivos. A precisão indica, portanto, quão confiáveis são as amostras que o modelo classificou como pertencentes a uma determinada classe.

3.1.5.10 Recall

O *recall* (também chamado de sensibilidade ou *true positive rate* — TPR) quantifica a capacidade do modelo de identificar corretamente todas as instâncias positivas daquela classe:

$$\text{Recall} = \frac{VP}{VP + FN} \quad (3.4)$$

onde FN corresponde aos falsos negativos. Valores baixos de *recall* indicam que o modelo está deixando de detectar elementos que deveriam ser reconhecidos, o que é especialmente crítico em tarefas onde a detecção completa é essencial.

3.1.5.11 Precisão, Recall e F1-score (Macro)

A precisão (*precision*) mede a proporção de predições corretas dentre todas as amostras atribuídas a uma determinada classe, sendo definida por:

$$\text{Precisão} = \frac{VP}{VP + FP}. \quad (3.5)$$

O *recall* (ou sensibilidade) quantifica a capacidade do modelo em identificar corretamente todas as amostras pertencentes a uma classe:

$$\text{Recall} = \frac{VP}{VP + FN}. \quad (3.6)$$

O F1-score corresponde à média harmônica entre precisão e *recall*, integrando ambas as métricas em um único indicador:

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}. \quad (3.7)$$

No presente trabalho, essas métricas são calculadas na forma macro, isto é, são computadas individualmente para cada classe e posteriormente promediadas. Essa abordagem atribui o mesmo peso a todas as classes, sendo adequada para cenários multiclasse com possível desbalanceamento.

Os resultados apresentados na Tabela 3.2 representam, portanto, a qualidade efetiva do modelo. Da mesma forma, as métricas por classe (Tabela 3.3) evidenciam o comportamento do classificador em cada categoria específica, permitindo identificar classes mais desafiadoras ou com maior variabilidade interna. Essa análise detalhada é crucial para validar a robustez do modelo e orientar sua futura integração ao MAPE do FITPATH.

Tabela 3.2 – Desempenho do classificador ExtraTrees otimizando 13 classes de vídeos.

Métrica	Valor
Acurácia	0.6243
Balanced Accuracy	0.5806
Precisão Macro	0.6184
Recall Macro	0.5806
F1-score Macro	0.5950
Coeficiente de Matthews (MCC)	0.5779
Coeficiente Kappa de Cohen	0.5772

Complementarmente, a Tabela 3.3 detalha o desempenho específico em cada uma das 13 classes, permitindo analisar como o modelo se comporta frente às diferenças intrínsecas entre os tipos de conteúdo. Essa análise por classe é especialmente importante porque o conjunto de dados apresenta variações significativas de distribuição, com algumas classes muito mais representadas que outras.

A combinação das duas tabelas oferece um panorama completo da performance do modelo: enquanto a primeira sumariza o desempenho geral, a segunda revela eventuais assimetrias ou dificuldades específicas em classes particulares, fornecendo informações essenciais para avaliar a adequação do modelo no contexto do FITPATH.

Tabela 3.3 – Precisão, recall e F1-score por classe do modelo extraído pelo TPOT.

Classe	Precisão	Recall	F1-score
Abuse	0.64	0.73	0.68
Arrest	0.52	0.46	0.49
Arson	0.59	0.61	0.60
Assault	0.64	0.57	0.60
Burglary	0.62	0.62	0.62
Explosion	0.59	0.42	0.49
Fighting	0.55	0.53	0.54
RoadAccidents	0.62	0.60	0.61
Robbery	0.63	0.74	0.68
Shooting	0.65	0.51	0.57
Shoplifting	0.67	0.60	0.63
Stealing	0.62	0.67	0.64
Vandalism	0.71	0.49	0.58

Os resultados apresentados nas Tabelas 3.2 e 3.3 demonstram que o classificador otimizado pelo TPOT obteve desempenho consistente considerando a complexidade do problema, composto por 13 classes distintas e fortes diferenças de distribuição entre elas. O modelo baseado em *ExtraTreesClassifier* apresentou acurácia global de 62,4%, com *macro F1-score* de 0,595 e coeficientes Kappa e MCC acima de 0,57, indicando concordância moderada e boa capacidade discriminativa mesmo em cenário multiclasse desbalanceado.

Classes com maior número de amostras, como Robbery, Stealing e Burglary, apresentaram métricas superiores, como esperado pela literatura. Já classes com menor representatividade, como *Fighting* e *Explosion*, exibiram métricas inferiores, efeito típico de problemas com desbalanceamento estrutural. Ainda assim, o modelo manteve desempenho estável, validando a eficácia do modelo desenvolvido.

O modelo final selecionado pelo TPOT atingiu uma acurácia global de 0,6243 na classificação das 13 categorias de vídeos. Embora a acurácia seja uma medida intuitiva, ela não captura adequadamente a complexidade de problemas multiclasse e desbalanceados. Por isso, métricas adicionais foram analisadas.

O Balanced Accuracy apresentou valor de 0,5806, indicando que o modelo mantém desempenho estável mesmo em classes menos representadas. Já a Precisão Macro e o Recall Macro foram de 0,6184 e 0,5806, respectivamente, refletindo que o modelo consegue identificar corretamente um conjunto diversificado de padrões de tráfego sem depender de classes dominantes.

O F1-score Macro, de 0,5950, demonstra que o classificador alcança um equilíbrio adequado entre precisão e recall em todas as classes, o que é essencial para aplicações que exigem distinção entre comportamentos de movimentação diferentes.

Os coeficientes Cohen Kappa (0,5772) e MCC (0,5779) também confirmam que o modelo possui concordância moderada com os rótulos reais, superando modelos

baseados apenas em aleatoriedade e capturando padrões reais na distribuição dos dados.

No contexto deste trabalho, tais resultados são particularmente expressivos, pois o problema envolve 13 classes com comportamentos distintos, utiliza séries temporais curtas de apenas 60 pontos, apresenta vídeos com grande variabilidade interna, possui um dataset naturalmente desbalanceado e ainda apresenta classes com sobreposição estatística relevante. Assim, o desempenho alcançado demonstra que o modelo encontrado pelo TPOT, composto por RobustScaler seguido de ExtraTreesClassifier, é capaz de identificar padrões consistentes de movimentação em fluxos de vídeo e fornece uma base sólida para sua integração ao módulo MAPE do FITPATH.

3.1.5.12 Classificação dos Vídeos por Meio do Modelo Treinado

Para permitir essa integração, foi desenvolvido um script de monitoramento capaz de operar em tempo real sobre os arquivos de *trace* gerados pelas fontes de vídeo. Esse módulo realiza leitura contínua dos logs de tráfego produzidos pelas câmeras ou fluxos de vídeo, segmenta o fluxo em janelas fixas de 60 quadros, extrai automaticamente as *features* estatísticas definidas nas etapas anteriores, normaliza os atributos utilizando o *scaler* treinado e, por fim, infere a classe correspondente por meio do modelo selecionado pelo TPOT. Cada janela é classificada de forma independente, permitindo que o sistema identifique mudanças no padrão de movimentação ao longo do vídeo e reaja imediatamente a picos ou quedas na demanda de bits. O script inicia carregando os arquivos de treinamento (.pkl), que correspondem ao modelo final selecionado automaticamente pelo TPOT e ao normalizador ajustado durante o processo de treinamento.

Para cada arquivo de *trace* armazenado no diretório definido (RAIZ), o script realiza a leitura no formato <frame> <bytes> <tempo>, converte os valores de bytes para bits por meio da relação $bits = bytes \times 8$, reconstrói a sequência temporal dos quadros, extrai as mesmas *features* estatísticas utilizadas no treinamento, normaliza os

atributos com o `scaler` e classifica o vídeo em uma das 13 categorias a partir do modelo treinado. Após esse processo, cada vídeo é associado à classe estimada e armazenado na estrutura `videos_por_classe`, o que possibilita agrupar vídeos com padrões de movimentação semelhantes.

Para cada classe identificada, o script cria uma planilha específica dentro do arquivo:

Tabela 3.4 – Exemplo de frames contendo tempo e bits por frame para a classe *Abuse*.

Frame	Tempo (s)	Bits por frame (média da classe)
0	0.000000	865.68
1	0.033333	1149.47
2	0.066667	974.30
3	0.100000	2321.09
4	0.133333	1092.63
5	0.166667	1472.96
6	0.200000	1832.43
7	0.233333	1653.63
8	0.266667	1714.22
9	0.300000	1211.70
⋮	⋮	⋮

Fonte: Elaborado pelo autor a partir da aba *Abuse* do arquivo *fitpath_13classes.xlsx*.

Cada planilha contém a coluna *frame*, numerada de 0 a 59, seguida por uma coluna para cada vídeo pertencente à classe, na qual são registrados os valores de bits por quadro correspondentes. Ao final, inclui-se uma coluna adicional que representa a média da classe, calculada segundo

$$\bar{b}(i) = \frac{1}{N} \sum_{k=1}^N bits_k(i),$$

em que N corresponde ao número de vídeos atribuídos à classe e $bits_k(i)$ indica a quantidade de bits no quadro i do vídeo k .

O resultado desse processo é um conjunto de séries temporais agregadas que descrevem o comportamento médio de cada classe em termos de variação de bits por frame.

3.2 Integração com o FITPATH

A integração do modelo de classificação com o FITPATH constitui a etapa final e central deste trabalho, pois permite que o sistema de roteamento adaptativo utilize informações extraídas diretamente do conteúdo do vídeo para ajustar dinamicamente a taxa de bits alocada a cada fluxo. Em vez de utilizar valores constantes do alvo da taxa de codificação do vídeo, o FITPATH passa a operar orientado por dados reais de movimentação, fornecidos continuamente pelo classificador treinado.

3.2.1 Implementação das Taxas Simuladas para o MAPE

Para permitir a simulação determinística do FITPATH, foi necessário definir perfis realistas de demanda de rede para cada fluxo. Essa etapa foi implementada utilizando diretamente os valores médios obtidos a partir dos arquivos de *trace* (*st_*), o que garante forte aderência ao comportamento real dos vídeos analisados. A partir dos *traces* individuais pertencentes a cada classe, foi construída uma janela temporal contendo a taxa média de bits observada, definida por

$$\text{kbps}(t) = \frac{8 \times \text{bytes}(t) \times \text{FPS}}{1000}.$$

Esse valor médio por classe passa a ser utilizado como entrada para o MAPE, permitindo que o sistema simule o comportamento de um fluxo representativo daquela categoria. Assim, preserva-se a dinâmica temporal característica do conteúdo transmitido, possibilitando que as avaliações conduzidas pelo FITPATH reflitam a variabilidade real observada experimentalmente, bem como a demanda de recursos associada a diferentes padrões de movimentação — elementos que impactam diretamente métricas de vazão, atraso e perda sob distintas condições de rede.

As 13 classes definidas ao longo da modelagem correspondem a padrões distintos de complexidade visual, estimados a partir da quantidade de bits por frame registrada

nos *traces* experimentais. Para cada classe C_i , o FITPATH utiliza uma curva média de consumo de bits ao longo dos 60 frames analisados, representada como

$$C_i \longrightarrow \mu_i(f), \quad f \in [0, 59],$$

em que $\mu_i(f)$ indica o traço médio de bits da classe C_i no frame f , calculado diretamente a partir dos logs de transmissão.

A partir desses perfis médios, o FITPATH extrai o valor agregado

$$\text{trace}_{\text{requerido}} = \overline{\mu_i},$$

que sintetiza a demanda típica da classe e funciona como parâmetro de entrada para o módulo adaptativo do protocolo. Dessa forma, classes com maior variabilidade e picos intensos de bits requerem maior alocação de banda, enquanto classes com baixa movimentação podem operar com rotas mais simples e menor consumo de recursos.

O mecanismo resultante permite que a configuração de recursos da rede deixe de ser fixa e passe a refletir diretamente a complexidade temporal das cenas presentes no dataset. Com isso, reduz-se o desperdício de capacidade, aprimora-se a escolha de rotas e alcança-se maior eficiência operacional — uma vez que o objetivo das multiclasss é fornecer ao FITPATH perfis representativos e estáveis de consumo.

3.2.2 Benefícios da integração final

A alocação de recursos passa a ser proporcional à complexidade do vídeo, permitindo que o sistema responda rapidamente a mudanças súbitas no conteúdo transmitido e reduzindo desperdícios de largura de banda. Além disso, a integração mantém a qualidade de experiência (QoE) mesmo em cenários de rede limitados e possibilita que as decisões de roteamento sejam orientadas por dados reais, em vez de heurísticas fixas. Por fim, o processo de incorporação ao MAPE ocorre de forma simples e eficiente, transformando o FITPATH de um mecanismo estático para uma solução cognitiva,

adaptativa e sensível ao conteúdo, capaz de ajustar a transmissão de acordo com a movimentação visual detectada em cada janela.

4 EXPERIMENTOS E RESULTADOS

A metodologia de avaliação da proposta foi realizada por meio de simulações realizadas utilizando os scripts desenvolvidos em Python e os códigos do MAPE, considerando diferentes cenários de rede e complexidade das cenas.

Inicialmente, com o objetivo de validar a definição das classes, foram avaliadas as distribuições das classes assim como os traces de vídeo gerados para cada classe. Em seguida, foi avaliado a aplicação do modelo treinado em cenários reais, para demonstrar a aplicabilidade do classificador em vídeos dinâmicos, simulando um ambiente prático de transmissão. Por fim, foi avaliado a integração com o FITPATH, considerando as simulação das taxas médias definidas pelas classes de vídeo.

4.1 Distribuição das Classes

Com o objetivo de analisar o comportamento médio de cada classe de vídeo ao longo do tempo, foi construída uma representação consolidada das curvas de bits por frame, obtidas a partir dos *traces* processados na etapa de extração. A Figura 4.1 apresenta as curvas médias correspondentes às 13 categorias utilizadas no classificador, permitindo uma visualização comparativa direta do padrão de consumo de bits entre diferentes tipos de conteúdo.

Cada linha do gráfico representa a média dos 60 quadros iniciais de todos os vídeos pertencentes a uma determinada classe, expressa em bits por frame. Essa abordagem evidencia como diferentes perfis de movimentação apresentam assinaturas temporais distintas: classes associadas a maior dinâmica visual, como *Robbery*, *Explosion* e *RoadAccidents*, exibem picos acentuados e maior variabilidade entre os frames; por outro lado, classes de menor movimentação, como *Stealing*, *Vandalism* e *Shoplifting*, apresentam curvas mais suaves, com menor amplitude e variação temporal.

A sobreposição das curvas em um único gráfico facilita a comparação entre os perfis médios, permitindo observar tendências gerais, diferenças estruturais entre classes e padrões recorrentes de comportamento. Essa visualização fornece uma interpretação intuitiva da complexidade temporal de cada categoria e justifica, do ponto de vista estatístico, a necessidade de uma modelagem multiclasse: cada classe segue uma distribuição específica de consumo de bits, refletindo a natureza distinta das cenas que representa.

Assim, o gráfico consolidado complementa a análise das métricas estatísticas extraídas e reforça o papel das curvas médias na geração das taxas simuladas utilizadas pelo mecanismo adaptativo do FITPATH.

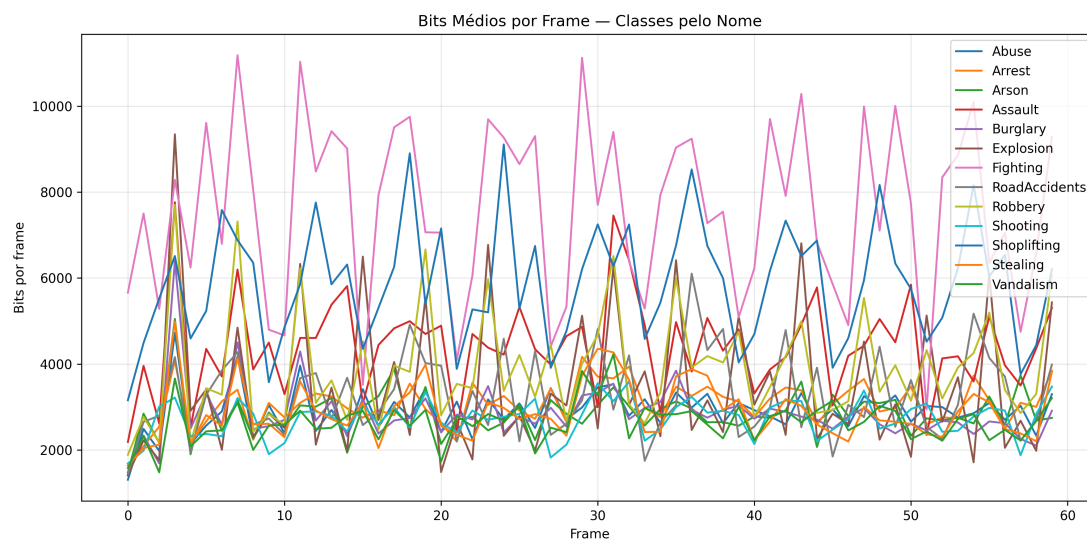


Figura 4.1 – Pacotes médios por frame para as classes detectadas pelo classificador, gerados a partir dos traces reconstruídos.

Fonte: Elaborado pelo autor a partir dos dados experimentais.

4.2 Comportamento Temporal dos Fluxos de Vídeo

A partir das curvas médias consolidadas, torna-se possível realizar uma análise mais aprofundada do comportamento temporal dos fluxos de vídeo. A principal motivação desse estudo é compreender como a taxa de bits evolui ao longo dos frames e de que forma essa evolução reflete a complexidade visual de cada classe. Para isso, foram gerados gráficos individuais para cada categoria, nos quais o eixo horizontal representa a progressão dos frames, enquanto o eixo vertical exibe o valor médio de bits transmitidos em cada posição temporal da janela de 60 quadros.

Essa forma de visualização permite distinguir claramente padrões característicos de cada classe. Vídeos de menor movimentação tendem a apresentar curvas suaves, com baixa amplitude e variação temporal reduzida, indicando estabilidade no consumo de bits ao longo da sequência. Em contrapartida, classes associadas a cenas dinâmicas exibem picos abruptos, oscilações acentuadas e flutuações frequentes, refletindo maior demanda de codificação e maior variabilidade temporal.

Além de facilitar a interpretação visual dos padrões presentes nos dados, esses gráficos desempenham um papel fundamental na validação da metodologia adotada. As diferenças observadas entre as curvas corroboram os critérios estatísticos empregados na classificação, demonstrando que as classes realmente capturam níveis distintos de complexidade temporal.

A seguir são apresentadas as curvas médias dos bits por frame para cada uma das 13 classes, acompanhadas de uma breve análise interpretativa. Cada gráfico reflete a estrutura temporal média extraída diretamente dos `traces` processados.

A classe *Abuse* apresenta média baixa (3000 bits/frame) e amplitude de apenas 144 bits (Figura 4.2), indicando comportamento predominantemente estável e de baixa complexidade temporal.

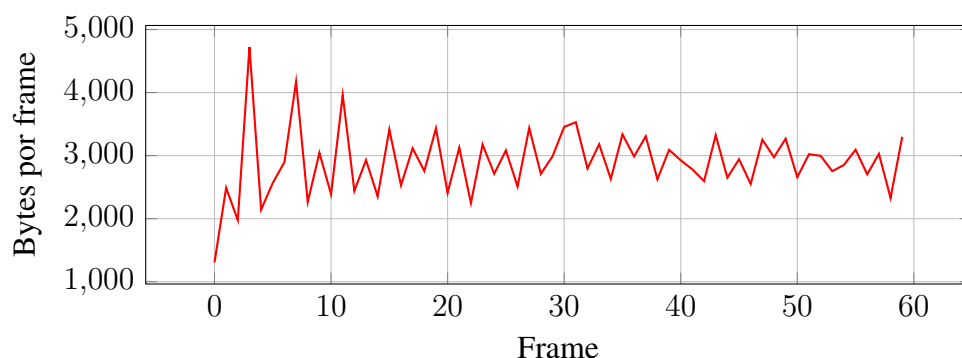


Figura 4.2 – Trace representativo da classe *Abuse*. Fonte: Autor

A classe *Arrest* exibe média de 3064 bits/frame e amplitude de 2519 bits (Figura 4.3), refletindo cenas que alternam momentos estáticos e ações rápidas de abordagem policial.

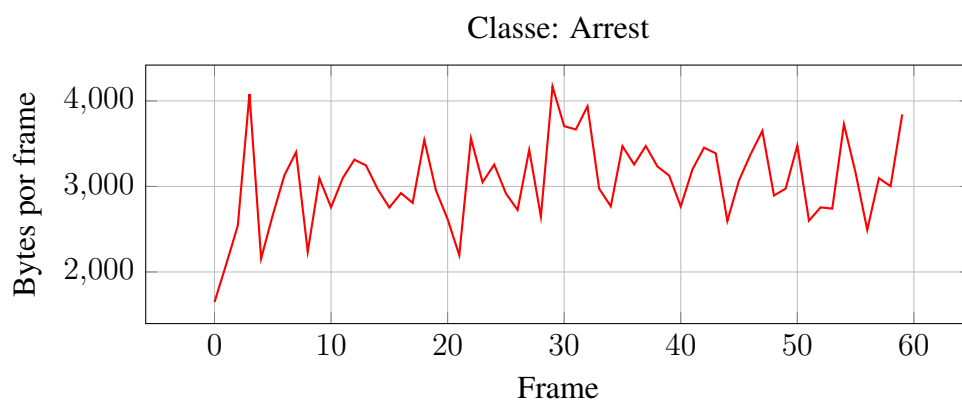


Figura 4.3 – Trace representativo da classe *Arrest*. Fonte: Autor

A classe *Arson* mantém média de 2843 bits/frame e amplitude de 3572 bits (Figura 4.4), indicando influência visual de fogo, fumaça e elementos ambientais variáveis.

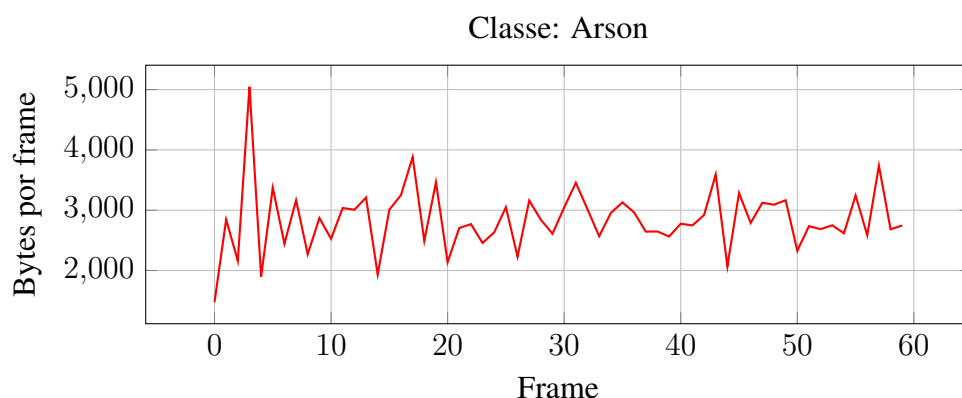


Figura 4.4 – Trace representativo da classe *Arson*. Fonte: Autor

A classe *Assault* é uma das mais dinâmicas do dataset, com média de 4407 bits/frame e amplitude extrema de 5575 bits (Figura 4.5), evidenciando agressões e movimentos bruscos.

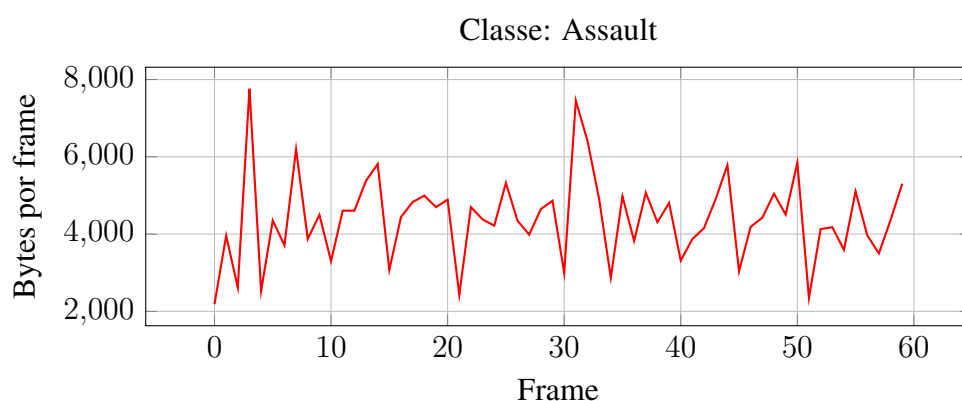


Figura 4.5 – Trace representativo da classe *Assault*. Fonte: Autor

A classe *Burglary* exibe média de 2848 bits/frame e amplitude de 4993 bits (Figura 4.6), sugerindo variação moderada típica de invasões e manipulação de objetos.

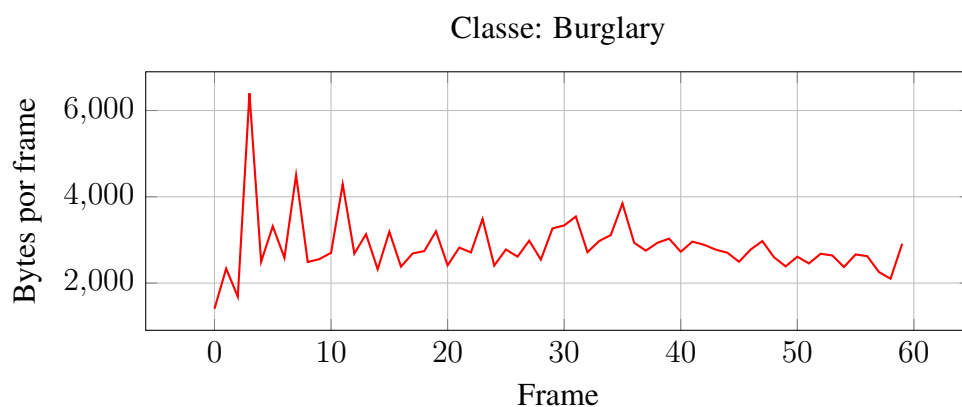


Figura 4.6 – Trace representativo da classe *Burglary*. Fonte: Autor

A classe *Explosion* tem média alta (3436 bits/frame) e amplitude ampla de 7928 bits (Figura 4.7), efeito de chamas, fragmentos e variação de luz intensa.

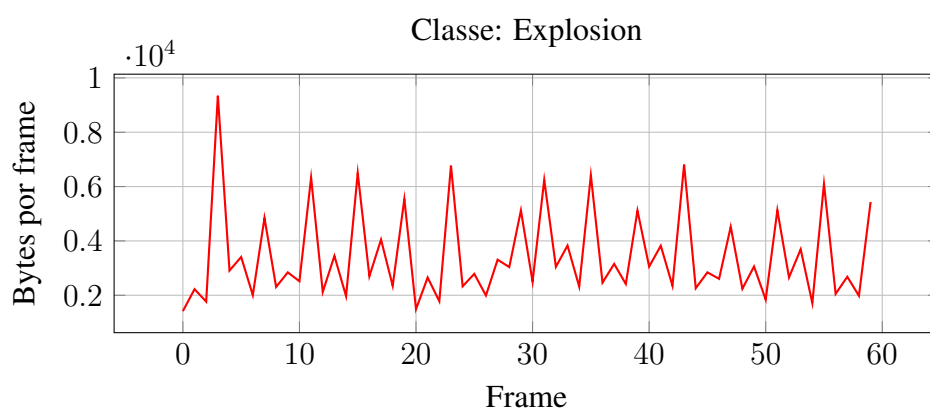


Figura 4.7 – Trace representativo da classe *Explosion*. Fonte: Autor

A classe *Fighting* apresenta média de 7574 bits/frame e amplitude de 8238 bits (Figura 4.8), resultante de movimentos físicos intensos e contínuos.



Figura 4.8 – Trace representativo da classe *Fighting*. Fonte: Autor

A classe *RoadAccidents* possui média de 3278 bits/frame e amplitude superior a 4665 bits (Figura 4.9), associada a colisões e movimento intenso de veículos.

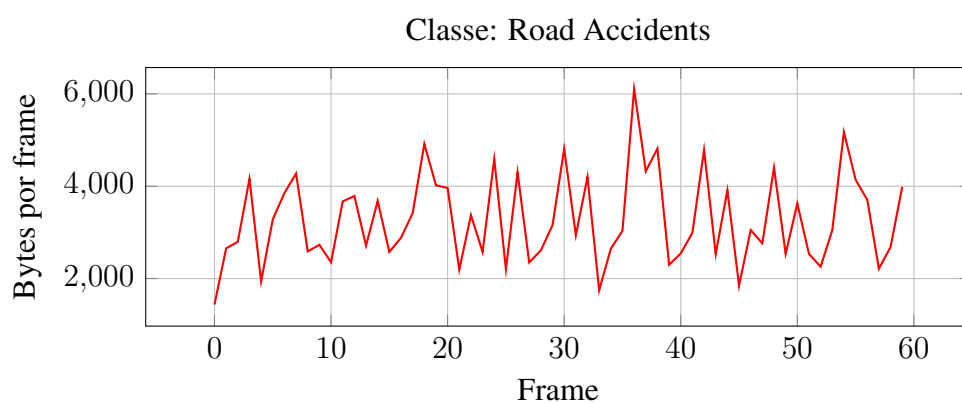


Figura 4.9 – Trace representativo da classe *Road Accidents*. Fonte: Autor

A classe *Robbery* apresenta média de 3888 bits/frame e amplitude de 5817 bits (Figura 4.10), com oscilações moderadas associadas a ações rápidas, interação entre indivíduos e deslocamentos bruscos.



Figura 4.10 – Trace representativo da classe *Robbery*. Fonte: Autor

A classe *Shooting* exibe média de 2727 bits/frame e amplitude de 1905 bits (Figura 4.11), alternando períodos estáveis com picos relacionados a disparos e transições rápidas.

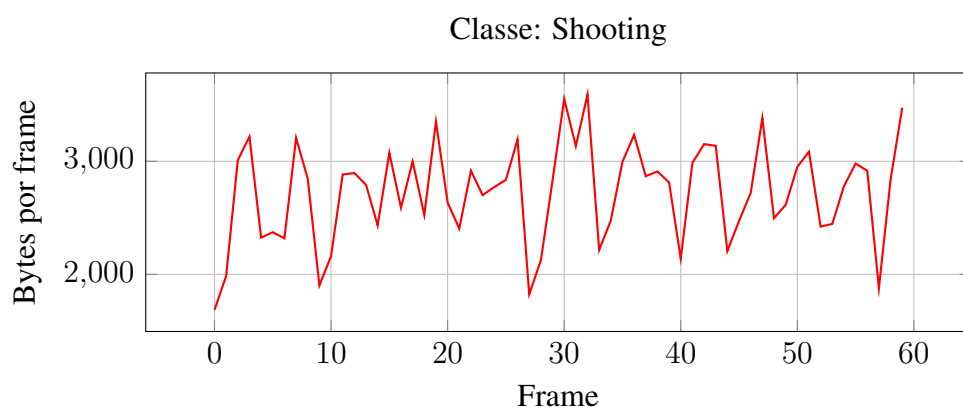


Figura 4.11 – Trace representativo da classe *Shooting*. Fonte: Autor

A classe *Shoplifting* apresenta média elevada (5873 bits/frame) e amplitude extrema (5952 bits), refletindo cenas com variação abrupta e irregularidades intensas (Figura 4.12).

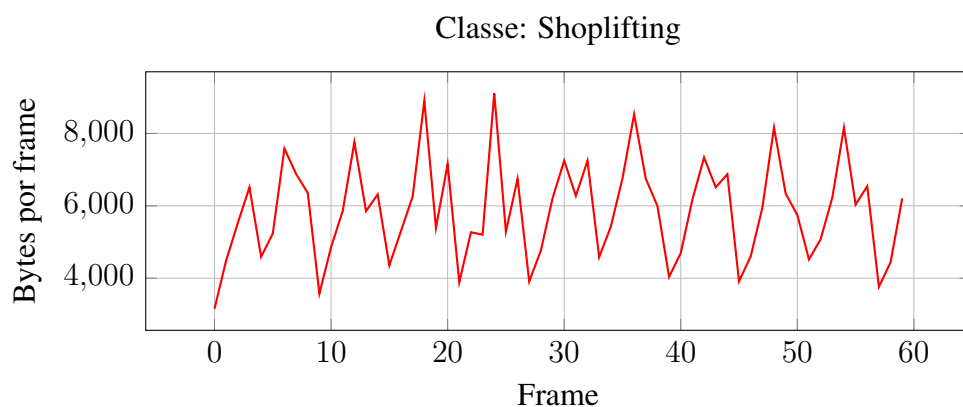


Figura 4.12 – Trace representativo da classe *Shoplifting*. Fonte: Autor

A classe *Stealing* tem média baixa (2889 bits/frame) e amplitude moderada (3352 bits), sugerindo cenas estáveis com pequenas flutuações (Figura 4.13).

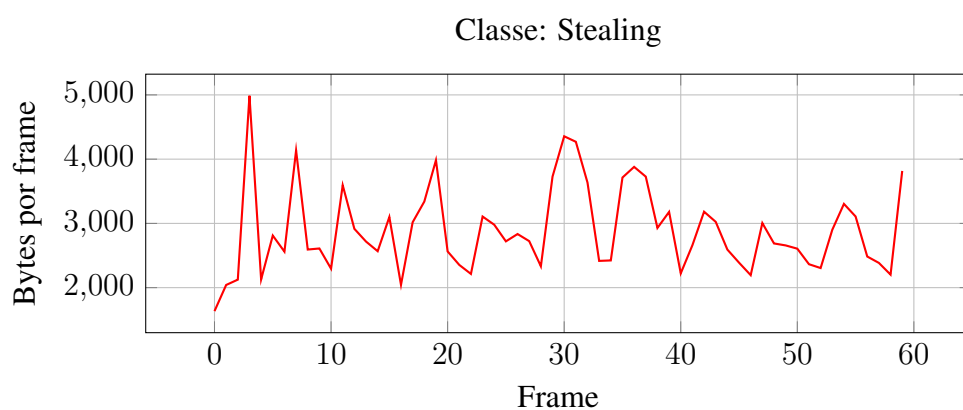


Figura 4.13 – Trace representativo da classe *Stealing*. Fonte: Autor

A classe *Vandalism* apresenta média baixa (2648 bits/frame) e amplitude de 2746 bits (Figura 4.14), refletindo cenas pouco dinâmicas e relativamente estáveis.

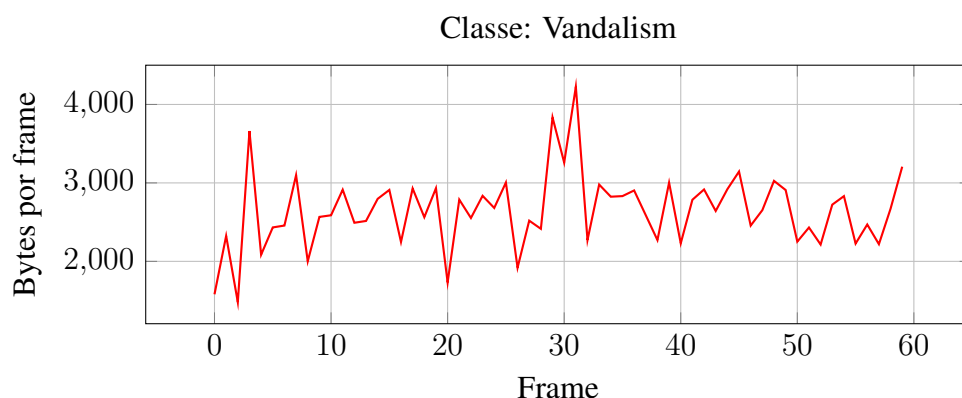


Figura 4.14 – Trace representativo da classe *Vandalism*. Fonte: Autor

A análise detalhada das treze classes revelou padrões temporais distintos e consistentes, evidenciados pelas curvas médias de bits por frame extraídas diretamente dos *traces*. As diferenças observadas entre as classes demonstram que cada tipo de cena possui uma assinatura temporal própria, refletindo sua complexidade visual e dinâmica interna. Enquanto algumas classes exibem comportamento predominantemente estável, caracterizado por baixas amplitudes e variações suaves — como *Abuse*, *Stealing* e *Vandalism* — outras apresentam flutuações acentuadas, picos abruptos e grande variabilidade temporal, como *Explosion*, *Assault* e *Shoplifting*.

Essas curvas reforçam a adequação das métricas estatísticas utilizadas para caracterizar a movimentação nos vídeos, uma vez que a variabilidade, a amplitude e o comportamento temporal capturado pelas séries se alinham às descrições semânticas das classes e ao nível de complexidade associado a cada uma. Além disso, a distinção clara entre os padrões temporais confirma a separabilidade intrínseca entre as classes no espaço de atributos, o que dá suporte à etapa posterior de classificação automática por meio de aprendizado de máquina.

Em síntese, a caracterização temporal dos fluxos de vídeo evidencia que as treze classes capturam diferentes níveis de dinamismo, variabilidade e complexidade visual, justificando o emprego de um modelo multiclasse e a subsequente integração com o

FITPATH para adaptação da taxa de bits em cenários reais de transmissão. Essa etapa consolida o entendimento sobre o comportamento dos vídeos e prepara o terreno para a avaliação quantitativa do classificador e sua aplicação prática em um sistema adaptativo de rede.

4.3 Performance do MAPE utilizando o Classificador

Por fim, para demonstrar a performance do MAPE ao considerar o classificador de vídeo para determinar o tipo de tráfego que será analisado para escolhas de rotas, foram realizados 3 cenários de experimentos considerando uma topologia de rede mesh apresentada na Figura 4.15. Essa topologia é representada por um grafo, onde os vértices são os nós da rede e as arestas são as conexões entre os nós. O peso das arestas representam a qualidade do sinal medida em Contagem de Transmissão Esperada (Expected Transmission Count) (ETX) métrica utilizada para avaliar a qualidade de um enlace, indicando o número esperado de transmissões até que um pacote seja recebido com sucesso no destino.

Os valores médios empregados como referência encontram-se na Tabela 4.1. Tais valores foram calculados a partir das taxas de bits observadas nos fluxos de tráfego médio, conforme detalhado na figura 4.1.

Para a avaliação, foi considerado um cenário com dois fluxos de dados, cada um utilizando rotas distintas para minimizar a interferência direta entre eles. Foram configurados três saltos para cada caminho, definidos como: Fluxo 0: $1 \rightarrow 3 \rightarrow 5 \rightarrow 0$ e Fluxo 1: $2 \rightarrow 4 \rightarrow 5 \rightarrow 0$. A inclusão da estimativa de tráfego no processo de seleção de rotas é necessária para otimizar o desempenho do sistema, uma vez que permite identificar gargalos e distribuir a carga de forma equilibrada entre os enlaces. Dessa forma, busca-se reduzir a latência, evitar perdas de pacotes e melhorar a vazão, aspectos diretamente relacionados à Qualidade de Serviço (QoS) [27,46].

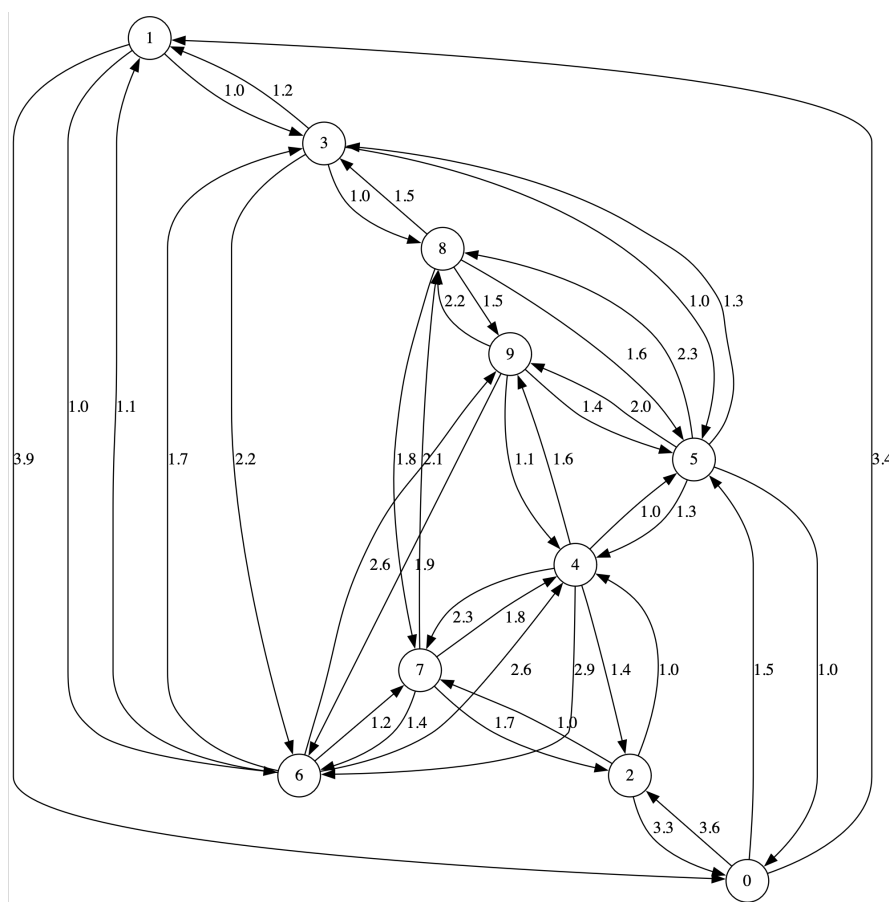


Figura 4.15 – Topologia Rede Mesh, Fonte: Autor.

Inicialmente, foram realizados experimentos utilizando um tráfego CBR, considerando a taxa média de cada classe. Os resultados apresentados na Figura 4.16 demonstram um comportamento de rede altamente determinístico e estável, independente da categoria do vídeo. Embora haja uma variação expressiva na demanda de tráfego — com a categoria Fighting exigindo mais que o dobro da vazão (3.635 kbps) em relação a categorias Vandalism (1.271 kbps), essa demanda não provocou congestionamento. Isso é evidenciado pela uniformidade das métricas de atraso e perdas, onde todas as categorias apresentaram o atraso médio menores que 100 ms e perdas de pacotes inferiores a 1%. Isso indica que, no modo CBR, a ausência de rajadas abruptas de tráfego permite um dimensionamento de fila mais eficiente, onde o impacto da complexidade do vídeo recai

Tabela 4.1 – Bitrate médio estimado para cada uma das 13 classes, calculado a partir dos valores médios de bits por frame do consolidado. Fonte: Autor.

Classe	Bitrate Médio (kbps)
Abuse	697.04
Arrest	735.44
Arson	682.48
Assault	1057.68
Burglary	683.52
Explosion	824.72
Fighting	1817.92
RoadAccidents	786.88
Robbery	933.12
Shooting	654.64
Shoplifting	1409.68
Stealing	693.52
Vandalism	635.76

puramente sobre o consumo de largura de banda, sem degradar a latência ou a integridade dos pacotes, desde que a capacidade do canal não seja excedida.

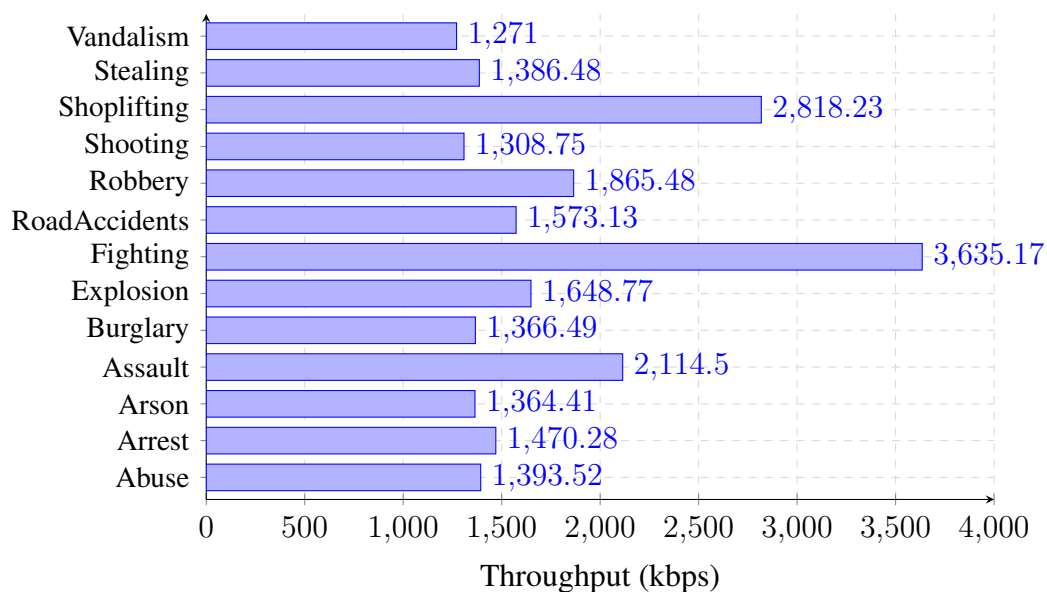


Figura 4.16 – Comparação da vazão agregada (2 fluxos, 3 saltos) entre transmissões CBR das diferentes classes. Fonte: Autor

Para as simulações usando tráfego VRB, os resultados evidenciam que a natureza

do conteúdo de vídeo e a consequente variação de suas taxas de bits exercem um impacto determinante sobre o desempenho da rede, demonstrando a importância de classificar a variação das taxas. Observa-se nas Figuras 4.17, 4.18 e 4.19, uma distinção entre categorias: cenários de alta dinamicidade, como Fighting e Shoplifting, geram padrões de tráfego instáveis (rajadas) que, embora resultem em uma vazão média agregada inferior (1.160 kbps) devido à degradação do canal, saturam os buffers da rede, provocando as maiores taxas de perda (acima de 48%) e elevação do atraso. Em contrapartida, categorias como Shooting e Vandalism apresentam um comportamento de tráfego mais estável, permitindo que a rede mantenha perdas próximas a zero (0,04%) e a menor latência média registrada (131 ms). Essa variabilidade da taxa de bits, inerente à complexidade de cada vídeo, pode gerar o congestionamento momentâneo dos enlaces, influenciando diretamente na qualidade do serviço.

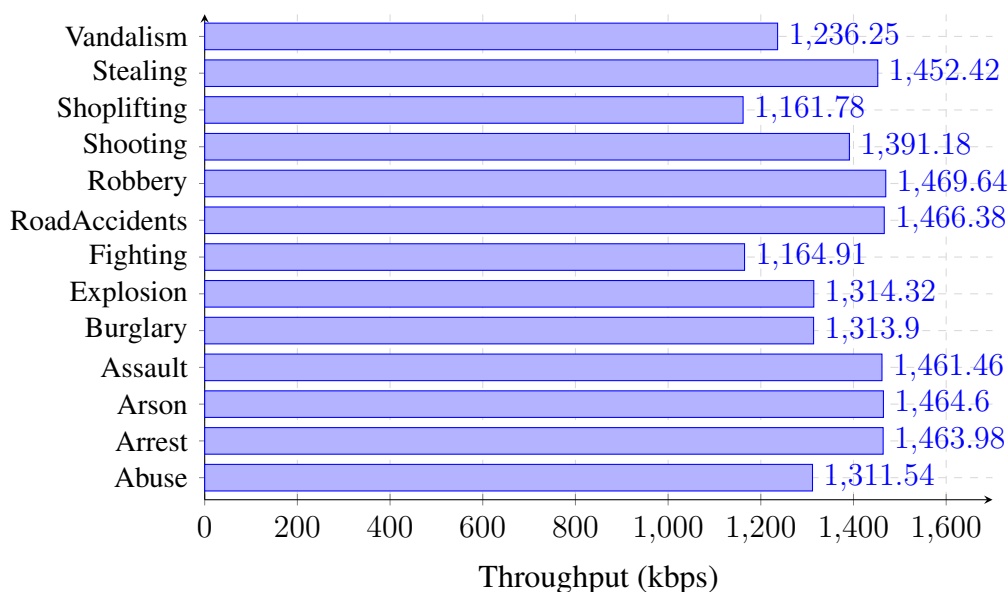


Figura 4.17 – Comparação da vazão agregada (2 fluxos, 3 saltos) entre transmissões VBR das diferentes classes. Fonte: Autor

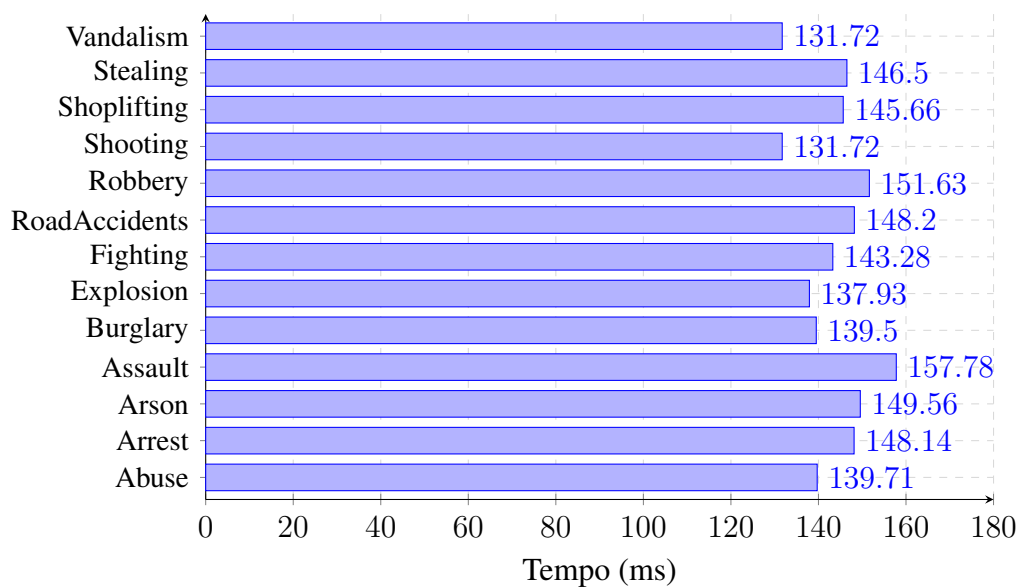


Figura 4.18 – Comparação do delay médio (2 fluxos, 3 saltos) entre transmissões VBR das diferentes classes. Fonte: Autor.

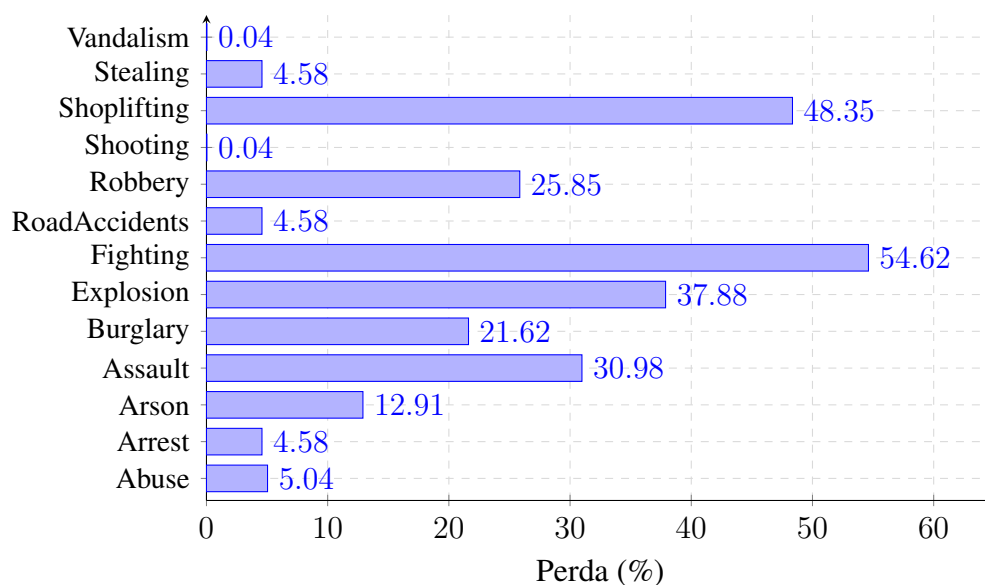


Figura 4.19 – Comparação da perda de pacotes média (2 fluxos, 3 saltos) entre transmissões VBR das diferentes classes. Fonte: Autor

5 CONCLUSÃO

A conclusão deste trabalho apresenta uma estratégia para o roteamento adaptativo de fluxos de vídeo em redes de Internet das Coisas (IoT), com foco na otimização da transmissão de vídeo em tempo real. O principal objetivo foi desenvolver e validar experimentalmente, por meio de aprendizado de máquina, um modelo de estimativa da variação da taxa de bits que poderá futuramente ser integrado ao framework FITPATH, permitindo decisões de roteamento mais precisas e adaptativas conforme a complexidade do vídeo transmitido e as condições da rede.

A contribuição central da proposta reside na demonstração de que é possível ajustar dinamicamente as rotas de transmissão a partir da detecção de variações abruptas na taxa de bits, especialmente durante transições de cena. Com isso, o FITPATH poderá, futuramente, selecionar rotas mais adequadas para cada fluxo, otimizando a utilização dos recursos da rede e proporcionando melhorias tanto na qualidade de serviço (QoS) quanto na qualidade de experiência do usuário (QoE).

A etapa prática deste trabalho envolveu a criação e avaliação de um sistema de classificação de complexidade de movimentação baseado em aprendizado de máquina, cujo potencial de integração ao FITPATH foi analisado por meio de experimentos com captura, processamento e análise de vídeos em tempo real. Os resultados evidenciaram a eficácia do modelo em identificar diferentes padrões de movimentação e sugerem que, se integrado ao processo de roteamento, pode contribuir para maior eficiência no uso dos recursos da rede. Além disso, abrem caminho para futuras implementações de sistemas auto-adaptativos e de autoaprendizagem no contexto de redes inteligentes para IoT.

5.1 Limitações do Trabalho

Apesar dos avanços alcançados na modelagem, classificação e integração preliminar com o FITPATH, algumas limitações estruturais do presente estudo devem ser consideradas.

A primeira delas diz respeito ao conjunto de vídeos utilizado: embora abrangente e diversificado, o dataset analisado não contempla toda a variabilidade possível de cenas reais, especialmente em ambientes urbanos de IoT, que podem envolver oclusões, iluminação variável, câmeras com diferentes taxas de compressão e situações imprevisíveis. Assim, a modelagem obtida a partir das treze classes pode não generalizar de forma ideal para fluxos de vídeo capturados em condições extremas ou não controladas.

Outra limitação relevante refere-se ao método de obtenção e processamento dos *traces*. Os valores de bits por frame foram derivados a partir de transmissões determinísticas utilizando `mp4trace`, o que fornece consistência experimental, mas não reproduz integralmente efeitos de rede reais, como jitter, congestionamento, interferência, enlaces instáveis ou perda aleatória de pacotes. Consequentemente, as curvas médias calculadas para cada classe refletem apenas o comportamento intrínseco dos vídeos, e não a interação completa entre conteúdo, codificação e rede — o que tende a ocorrer em ambientes IoT de produção.

Da mesma forma, a extração das métricas temporais foi limitada aos 60 primeiros quadros de cada sequência, obedecendo ao padrão definido para o classificador. Embora esse recorte seja adequado para criar uma janela representativa, ele não captura eventos mais longos ou padrões cíclicos que podem ocorrer em vídeos extensos. Além disso, a divisão em janelas fixas pode não ser suficientemente sensível a transições abruptas ou mudanças de contexto que acontecem fora do intervalo analisado.

O próprio modelo de classificação apresenta limitações inerentes. Embora o *ExtraTreesClassifier* tenha demonstrado desempenho consistente em um problema multiclasse complexo, a acurácia global de 62%, aliada a variações de precisão e recall entre as categorias, indica que algumas classes continuam sendo desafiadoras, especialmente aquelas com maior variabilidade interna ou menor número de amostras. A falta de balanceamento perfeito entre as classes e a ausência de técnicas de treinamento incremental também podem restringir a capacidade do modelo de generalizar para novos

padrões de movimentação que não estejam suficientemente representados nos dados.

No campo da integração com o FITPATH, a maior limitação está no fato de que o ajuste adaptativo de taxa de bits não foi aplicado em uma rede real ou num ambiente de simulação completo. A utilização de um ambiente determinístico impede a avaliação de como a classificação influenciaria rotas sob métricas reais de rede, como ETX, latência, perda, ocupação de buffers ou variação temporal da capacidade do enlace. Como consequência, ainda não é possível quantificar, de forma rigorosa, o impacto da classificação nas decisões de roteamento, nem medir efetivamente o ganho esperado de QoS ou QoE.

Também não foram considerados cenários com nós heterogêneos ou topologias variadas. Do ponto de vista da escalabilidade, não há evidências de como o sistema se comportaria ao classificar dezenas ou centenas de fluxos em tempo real.

Por fim, não foi realizada comparação direta com protocolos de roteamento estabelecidos, tais como AODV, OLSR, RPL ou variantes multipath. Essa ausência restringe a interpretação quantitativa da eficácia do FITPATH como solução de seleção de rotas orientada por conteúdo, limitando o estudo a uma prova de conceito de integração e não a uma validação comparativa completa.

5.2 Trabalhos Futuros

Diante das limitações identificadas, uma série de extensões e aprimoramentos podem ser explorados em trabalhos futuros. A primeira linha de evolução envolve a integração completa do modelo ao FITPATH em um ambiente de rede real ou em simuladores como, que permitiriam avaliar o impacto do classificador em cenários sujeitos a congestionamento, interferência, mobilidade e compartilhamento de recursos entre múltiplos fluxos.

Também se destaca a necessidade de realizar medições experimentais com

repetição estatística, variação de topologias e cenários com múltiplos fluxos concorrentes, permitindo avaliar escalabilidade, estabilidade e impacto global da solução proposta. Tais experimentos podem revelar gargalos, limitações de processamento e cenários de saturação que não são perceptíveis em ambientes determinísticos.

Por fim, sugerem-se estudos comparativos para mensurar ganhos reais da integração do classificador com o FITPATH em termos de desempenho, QoS e QoE. A partir dessas evidências experimentais, seria possível consolidar o uso de classificadores de vídeo como parâmetros adicionais para sistemas de roteamento adaptativo em redes IoT, ampliando a aplicabilidade da abordagem proposta.

REFERÊNCIAS

- [1] Advanced video coding for generic audiovisual services. Technical Report H.264, ITU-T, 2019.
- [2] Samira Afzal, Vanessa Testoni, Christian Esteve Rothenberg, Prakash Kolan, and Imed Bouazizi. A holistic survey of multipath wireless video streaming. *Journal of Network and Computer Applications*, 212:103581, 2023.
- [3] Charu C Aggarwal et al. *Data mining: the textbook*, volume 1. Springer, 2015.
- [4] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 2005.
- [5] Lamine Amour, Sami Souihi, Abdelhamid Mellouk, and S M Mushtaq. Q2ABR: QoE-aware adaptive video bit rate solution. *Int. J. Commun. Syst.*, 33(10), 2020.
- [6] F Bhering, C Albuquerque, D Passos, and L Ochi. Multipath selection mechanism for wireless video-surveillance systems. *LANOMS*, 2019.
- [7] F Bhering, D Passos, C Albuquerque, and K Obraczka. Efficient multipath selection for IoT video transmission. *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, pages 73–78, 2022.
- [8] Fabiano Bhering, Diego Passos, Célio Albuquerque, and Katia Obraczka. Efficient multipath selection for iot video transmission. In *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, pages 73–78, 2022.
- [9] Fabiano Bhering, Diego Passos, Katia Obraczka, and Célio Albuquerque. Network performance estimator with applications to route selection for IoT multimedia applications. *Simulation*, 100(1):23–37, 2024.

- [10] Fabiano Bhering, Diego Passos, Katia Obraczka, and Célio Albuquerque. Network performance estimator with applications to route selection for iot multimedia applications. *SIMULATION*, 100(1):23–37, 2024.
- [11] Fabiano Bhering, Diego Passos, Luiz Satoru Ochi, Katia Obraczka, and Célio Albuquerque. Wireless multipath video transmission: when IoT video applications meet networking—a survey. *Multimed. Syst.*, 28(3):831–850, 2022.
- [12] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- [13] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). Technical Report 3626, RFC Editor, October 2003. [Online; acessado em 2024].
- [14] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.
- [15] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.
- [16] Adele Cutler, D Richard Cutler, and John R Stevens. Random forests. *Ensemble machine learning: Methods and applications*, pages 157–175, 2012.
- [17] Matthias Feurer, Aaron Klein, and Katharina et al. Eggenberger. Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [18] F-A Fortin, F-M De Rainville, and M-A et al. Gardner. Deap: Evolutionary algorithms made easy. In *Journal of Machine Learning Research*, 2012.
- [19] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

- [20] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. "O'Reilly Media, Inc.", 2022.
- [21] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.
- [22] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automated machine learning: State-of-the-art and open challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [23] David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied logistic regression*. Wiley, 2013.
- [24] Yu-Wen Huang, Ching-Yeh Chen, Chen-Han Tsai, Chun-Fu Shen, and Liang-Gee Chen. Survey on block matching motion estimation algorithms and architectures with new results. *Journal of VLSI signal processing systems for signal, image and video technology*, 42(3):297–320, 2006.
- [25] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.
- [26] Rasool Jamal Kolaib and Jumana Waleed. Crime activity detection in surveillance videos based on developed deep learning approach. *Diyala Journal of Engineering Sciences*, pages 98–114, 2024.
- [27] James F Kurose and Keith W Ross. Computer networking: A top-down approach edition. *Addision Wesley*, 12, 2007.
- [28] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

- [29] Jun Li, Zhen Wu, and Qi Wang. Classification of video sequences by motion complexity using deep learning. *IEEE International Conference on Image Processing (ICIP)*, pages 2455–2459, 2016.
- [30] Y. Li, X. Zhang, et al. Video traffic modeling and prediction: A survey. *Journal of Network and Computer Applications*, 2020.
- [31] Ze-Nian Li, Mark S Drew, and Jiangchuan Liu. Modern video coding standards: H.264, h.265, and h.266. In *Texts in Computer Science*, pages 423–478. Springer International Publishing, Cham, 2021.
- [32] Christian Lottermann, Alexander Machado, Damien Schroeder, Yang Peng, and Eckehard Steinbach. Bit rate estimation for h.264/avc video encoding based on temporal and spatial activities. pages 3195–3199, 2014.
- [33] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 2017.
- [34] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.
- [35] Randal S Olson and Jason H Moore. Automating biomedical data science through tree-based pipeline optimization. *Applications in Genetics and Molecular Biology*, 2016.
- [36] Randal S. Olson, Ryan J. Urbanowicz, Peter C. Andrews, Nicole A. Lavender, La Creis Kidd, and Jason H. Moore. Tpot: A tree-based pipeline optimization tool for automating machine learning. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 251–260, 2016.

- [37] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python, 2011.
- [38] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.
- [39] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [40] Iain E. G. Richardson. *The H.264 Advanced Video Compression Standard*. Wiley, 2010.
- [41] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
- [42] E.M. Royer and C.E. Perkins. An implementation study of the aodv routing protocol. In *2000 IEEE Wireless Communications and Networking Conference. Conference Record (Cat. No.00TH8540)*, volume 3, pages 1003–1008 vol.3, 2000.
- [43] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.
- [44] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.
- [45] Mohammad A Talaat, Magdi A Koutb, and Hoda S Sorour. Psnr evaluation of media traffic over tfrc. *International Journal of Computer Networks and Communications*, 1(3):71–76, 2009.
- [46] Andrew S Tanenbaum. Network protocols. *ACM Computing Surveys (CSUR)*, 13(4):453–489, 1981.

- [47] Kai Ming Ting and Ian H. Witten. Issues in multi-class classification. *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1016–1021, 1999.
- [48] Yao Wang, Jörn Ostermann, and Ya-Qin Zhang. *Video processing and communications*, volume 1. Prentice hall Upper Saddle River, 2002.
- [49] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.
- [50] Kazushi Yamagishi and Tatsuya Hayashi. Parametric packet-layer model for monitoring video quality of iptv services. *IEEE Journal of Selected Areas in Communications*, 26(6):756–766, 2008.
- [51] Yang Yang, Liquan Shen, Hao Yang, and Ping An. A content-based rate control algorithm for screen content video coding. *Journal of Visual Communication and Image Representation*, 60:328–338, 2019.
- [52] Kaiqun Zhu, Zidong Wang, Guoliang Wei, and Xiaohui Liu. Adaptive set-membership state estimation for nonlinear systems under bit rate allocation mechanism: A neural-network-based approach. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8337–8348, 2023.