# Detecção de Intrusão em Redes com Sistema Multiagente e Orquestração Adaptativa

Bernardo do Nascimento Nunes, Luiz F. M. Vieira, *Member, IEEE,* Marcos A. M. Vieira, *Member, IEEE.*

***Abstract*—Sistemas de detecção de intrusão baseados em modelos únicos de aprendizado de máquina apresentam limitações na identificação de ataques complexos em redes de computadores. Modelos tabulares como XGBoost alcançam alta acurácia em features estatísticas mas não capturam padrões temporais presentes no tráfego, enquanto modelos de visão computacional aplicados a espectrogramas capturam esses padrões mas são computacionalmente custosos para operar sobre todo o tráfego. Neste trabalho, propomos um sistema multiagente composto por três componentes autônomos: um Agente Tabular baseado em XGBoost com explicabilidade via SHAP, um Agente Orquestrador com autocalibração de pesos por classe, e um Agente de Visão baseado em CNN sobre Mel-spectrogramas multifeature. O orquestrador decide adaptativamente quando acionar a CNN, aprendendo ao longo do tempo quais classes se beneficiam da análise visual sem intervenção humana. Avaliado no dataset UNSW-NB15 com seis classes de tráfego, o sistema alcança F1-macro de 0.654, uma melhoria de 9.4% sobre o XGBoost isolado. A CNN contribui especificamente nas classes DoS/Reconnaissance (+13 pontos de F1) e ataques minoritários (+9 pontos), sem degradar as classes que o modelo tabular já domina. Um Reasoning Log documenta cada decisão com justificativa, confiança e features relevantes, garantindo auditabilidade.**

***Index Terms*—Intrusion Detection System, Multi-Agent Systems, Adaptive Orchestration, Mel-Spectrogram, XGBoost, SHAP, Convolutional Neural Networks, UNSW-NB15.**

## I. Introdução

A crescente sofisticação dos ataques a redes de computadores demanda sistemas de detecção que sejam ao mesmo tempo precisos, eficientes e explicáveis. Ataques como DDoS, varreduras de portas e infiltrações persistentes causam prejuízos financeiros significativos e comprometem a disponibilidade de serviços críticos. Nesse contexto, Sistemas de Detecção de Intrusão (IDS) desempenham papel fundamental na identificação de comportamentos maliciosos em tempo real.

A literatura tradicional em IDS emprega classificadores supervisionados aplicados a conjuntos de atributos numéricos extraídos de fluxos de rede. Modelos baseados em gradient boosting, como o XGBoost [1], alcançam alta acurácia em features estatísticas mas não exploram a dinâmica temporal presente nos fluxos. Quando analisados como séries temporais, comportamentos maliciosos revelam padrões rítmicos que não aparecem de forma explícita nos atributos agregados. A ausência dessa perspectiva temporal reduz a sensibilidade dos modelos em situações ambíguas.

Representações perceptuais, como Mel-spectrogramas, permitem observar esses padrões temporais de forma visual. A Figura 1 apresenta um exemplo de espectrograma gerado a partir da concatenação de features de tráfego de rede, onde é possível observar a distribuição de energia ao longo do tempo e das bandas de frequência.

Fig. 1. Exemplo de Mel-spectrograma gerado a partir de features de tráfego de rede no dataset UNSW-NB15. A imagem codifica variações temporais de múltiplas features de rede em um mapa tempo-frequência.

No entanto, a integração de modalidades tabulares e visuais é realizada majoritariamente por fusão estática, onde ambos os modelos são acionados para todas as amostras independentemente da confiança de cada um. Essa abordagem é computacionalmente ineficiente e não explora o fato de que cada modelo erra em classes diferentes.

Este trabalho propõe uma abordagem diferente, inspirada no paradigma de sistemas multiagente [4]. O sistema é composto por três agentes autônomos que colaboram de forma adaptativa: um Agente Tabular que classifica e explica cada decisão via SHAP [2], um Agente Orquestrador que avalia a classe predita e decide adaptativamente quando acionar o terceiro componente, e um Agente de Visão baseado em CNN sobre Mel-spectrogramas. O diferencial central é que o orquestrador auto-calibra seus pesos ao longo do tempo, aprendendo sem intervenção humana em quais classes a CNN contribui mais. Cada decisão é documentada em um Reasoning Log estruturado que garante auditabilidade.

A Figura 2 apresenta a visão geral da arquitetura proposta.

Este trabalho investiga a seguinte questão: como combinar modelos de diferentes modalidades de forma adaptativa, acionando cada um apenas quando necessário, para melhorar a detecção de intrusões mantendo explicabilidade e eficiência?

Os objetivos específicos são: (i) desenvolver um Agente Tabular com XGBoost e SHAP para explicabilidade local por

Bernardo do Nascimento Nunes e Luiz Filipe Menezes Vieira são do Departamento de Ciência da Computação, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brasil. (e-mail: bernardodonascimento@ufmg.br, lfvieira@dcc.ufmg.br).

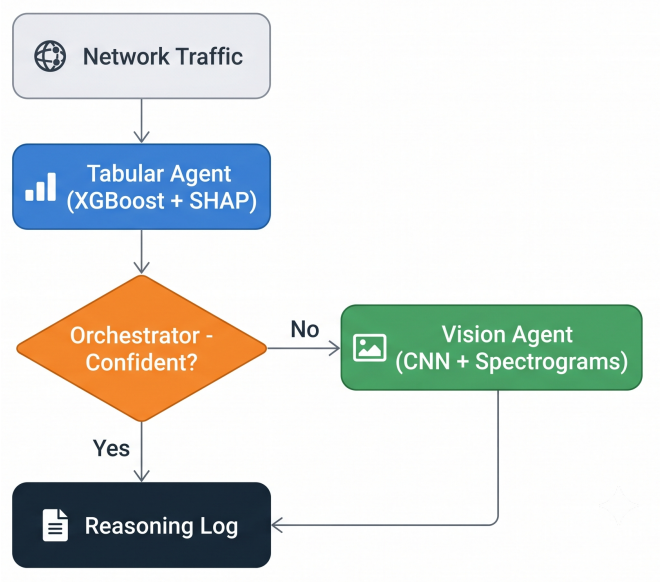


Fig. 2. Arquitetura do sistema multiagente proposto. O Agente Tabular classifica e explica via SHAP. O Orquestrador avalia a confiança da predição e decide se aciona o Agente de Visão. Todas as decisões são documentadas no Reasoning Log.

fluxo; (ii) implementar um Agente Orquestrador com auto-calibração de pesos por classe; (iii) construir um Agente de Visão com CNN sobre Mel-spectrogramas multifeature; (iv) avaliar comparativamente o sistema multiagente frente a modelos isolados; e (v) demonstrar a auditabilidade do sistema via Reasoning Log.

Os resultados demonstram que a fusão auto-calibrada melhora o F1-macro em 9.4% sobre o XGBoost isolado no dataset UNSW-NB15.

A organização do texto é a seguinte. A Seção II apresenta os fundamentos sobre Mel-spectrogramas, CNNs, XGBoost, SHAP e sistemas multiagente. A Seção III discute os trabalhos relacionados. A Seção IV detalha a arquitetura do sistema. A Seção V descreve a metodologia experimental. Os resultados são apresentados na Seção VI. Por fim, a Seção VII apresenta as conclusões.

## II. Fundamentação Teórica

Esta seção apresenta os conceitos essenciais utilizados na construção do sistema multiagente de detecção de intrusão. São descritos o funcionamento do Mel-spectrograma, das redes neurais convolucionais, do XGBoost, do SHAP para explicabilidade, e do paradigma de sistemas multiagente com orquestração adaptativa.

### A. Mel-spectrograma

O Mel-spectrograma é uma representação tempo-frequência obtida pela aplicação da Transformada de Fourier de Curto Prazo (STFT) sobre janelas sobrepostas do sinal de entrada, seguida de conversão para a escala de Mel, que aproxima a percepção humana de frequência. A imagem resultante evidencia variações rítmicas e padrões energéticos ao longo do tempo, sendo amplamente utilizada em processamento de áudio e reconhecimento de padrões.

O processo de geração envolve três etapas: (i) o sinal é dividido em quadros sobrepostos com janelamento de Hann; (ii) a FFT é aplicada a cada quadro para obter o espectro de potência; (iii) os coeficientes são projetados em filtros triangulares na escala Mel e convertidos para decibéis. O resultado é uma matriz bidimensional onde o eixo horizontal representa o tempo e o eixo vertical representa as bandas de frequência Mel.

Neste trabalho, a técnica é aplicada a sinais construídos pela concatenação de múltiplas features de rede. Diferentemente de abordagens anteriores que utilizam uma única variável temporal como Flow IAT Mean, o sinal composto captura simultaneamente volume de dados (load), ritmo de pacotes (inter-arrival time), direção do tráfego (source/destination) e comportamento de flags TCP. Essa representação multifeature permite que a CNN observe assinaturas visuais mais ricas e discriminativas por tipo de ataque, codificando relações entre features que não são visíveis em análises isoladas.

### B. Redes Neurais Convolucionais

Redes neurais convolucionais (CNNs) são modelos projetados para extrair padrões locais e hierárquicos em dados com estrutura espacial. Cada camada convolucional aplica um conjunto de filtros aprendidos que destacam regiões relevantes da imagem, enquanto operações de pooling reduzem a dimensionalidade e conferem invariância a pequenas translações.

A arquitetura adotada neste trabalho segue um design moderno composto por quatro blocos convolucionais. Cada bloco contém: convolução 3x3 com padding, normalização em lote (BatchNorm) para estabilizar o treinamento, ativação ReLU, e max pooling 2x2 para redução espacial. Dropout2D é aplicado entre blocos para regularização. Ao final dos blocos convolucionais, Global Average Pooling agrega a informação espacial em um vetor fixo, eliminando a necessidade de camadas Flatten que são propensas a overfitting em datasets pequenos. Duas camadas densas com dropout produzem a classificação final em seis classes via softmax.

A implementação foi realizada em PyTorch, utilizando class weights inversamente proporcionais à frequência de cada classe para lidar com o desbalanceamento natural do tráfego de rede. Data augmentation com flip horizontal, rotação aleatória e color jitter foi aplicada durante o treinamento para aumentar a diversidade das amostras.

### C. XGBoost

O XGBoost [1] é um algoritmo de gradient boosting que constrói árvores de decisão otimizadas sequencialmente, minimizando uma função de perda via gradiente. É reconhecido por sua robustez em dados tabulares heterogêneos e por sua eficiência computacional.

Neste trabalho, o XGBoost foi configurado para classificação multiclasse com objetivo *multi:softprob*, produzindo probabilidades por classe. Os hiperparâmetros foram otimizados via busca bayesiana com Optuna (50 trials), e o balanceamento de classes foi realizado com SMOTE antes do treinamento.

### D. SHAP

SHAP (SHapley Additive exPlanations) [2] é um método de explicabilidade baseado em valores de Shapley da teoria dos jogos cooperativos. Para cada predição, o SHAP atribui a cada feature uma contribuição positiva ou negativa para o resultado final, permitindo entender *por que* o modelo tomou aquela decisão.

Neste sistema, o TreeExplainer do SHAP é aplicado ao XGBoost, gerando para cada fluxo classificado as cinco features mais relevantes. Essa informação é registrada no Reasoning Log, permitindo auditoria das decisões em contextos de segurança onde a transparência é essencial.

### E. Sistemas Multiagente e Orquestração Adaptativa

Um sistema multiagente é composto por componentes autônomos (agentes) que percebem o ambiente, raciocinam sobre o estado e agem de forma independente, colaborando entre si via protocolo definido [4]. Diferentemente de pipelines fixos onde o fluxo é sempre o mesmo, sistemas agênticos tomam decisões dinâmicas baseadas no contexto.

A orquestração adaptativa é o mecanismo pelo qual um agente coordenador decide quem age e quando. Neste trabalho, o Agente Orquestrador mantém pesos de confiança por classe que são recalculados periodicamente com base na performance recente. Se a CNN deixa de contribuir em uma classe, seu peso nessa classe diminui automaticamente. Essa auto-calibração torna o sistema genuinamente autônomo: ele melhora com a experiência sem necessidade de reconfiguração manual.

## III. Trabalhos Relacionados

A detecção de intrusão em redes é um domínio consolidado da segurança da informação. Ao longo dos últimos anos, técnicas de aprendizado profundo e abordagens híbridas têm ganhado destaque pela capacidade de identificar padrões complexos em tráfego real. Esta seção discute os trabalhos mais relevantes nas três dimensões exploradas neste artigo: detecção tabular com explicabilidade, representações visuais de tráfego, e sistemas multiagente para segurança.

No campo da detecção tabular, modelos de ensemble como XGBoost e Random Forest alcançam alta acurácia em datasets de referência como CICIDS-2017 e UNSW-NB15 [3]. Talukder et al. [10] propõem um sistema baseado em oversampling e stacking para lidar com o desbalanceamento de classes, alcançando resultados competitivos. Porém, esses trabalhos não oferecem explicabilidade local por predição. A integração de SHAP em IDS tem sido investigada recentemente: Mia et al. [7] propõem análise visual de SHAP plots para diagnosticar falsos positivos e negativos, enquanto Ahmed et al. [8] demonstram que seleção de features guiada por SHAP melhora tanto performance quanto transparência.

Abordagens baseadas em representações visuais do tráfego têm demonstrado capacidade de capturar padrões temporais que escapam à análise tabular. Ahmad et al. [9] propõem a conversão de tráfego Wi-Fi em representações bidimensionais para classificação por CNN leve, avaliada com baixa latência de inferência no dataset AWID3. A análise multiescala, utilizando janelas temporais de diferentes tamanhos e múltiplas features concatenadas como sinal de entrada, permanece pouco explorada neste contexto e constitui uma das contribuições deste trabalho.

No campo de sistemas multiagente para segurança de redes, Tellache et al. [5] propõem um IDS baseado em aprendizado por reforço multiagente (MARL), onde agentes cooperam para detectar intrusões no CICIDS-2017 com baixa taxa de falsos positivos. Islam et al. [6] apresentam o MA-IDS, que combina Large Language Models com Retrieval Augmented Generation para raciocínio sobre intrusões em ambientes IoT. Ambos os trabalhos demonstram o potencial de abordagens multiagente, porém não exploram a orquestração adaptativa entre modalidades distintas de análise (tabular e visual) com calibração dinâmica de pesos por classe.

A proposta deste trabalho avança sobre o estado da arte ao integrar três dimensões: (i) explicabilidade local via SHAP em cada predição, (ii) representações visuais multifeature do tráfego processadas por CNN, e (iii) orquestração adaptativa com auto-calibração que aprende quando confiar em cada agente. Essa combinação diferencia-se de fusões estáticas tradicionais e de sistemas multiagente baseados exclusivamente em reinforcement learning ou LLMs.

## IV. Arquitetura do Sistema

A arquitetura do sistema proposto é organizada como um sistema multiagente composto por três componentes autônomos que colaboram por meio de um protocolo de comunicação estruturado. A Figura 2, apresentada na Seção I, ilustra o fluxo de decisão. Nesta seção, descrevemos cada agente e o mecanismo de orquestração.

### A. Visão Geral

O sistema recebe como entrada fluxos de tráfego de rede do dataset UNSW-NB15. Cada fluxo é processado inicialmente pelo Agente Tabular, que produz uma classificação multiclasse com probabilidades e explicabilidade. O Agente Orquestrador avalia a classe predita e decide se o Agente de Visão deve ser acionado. Quando acionado, a CNN processa um Mel-spectrograma da janela temporal correspondente e sua probabilidade é fundida com a do XGBoost. Todas as decisões são registradas no Reasoning Log.

A justificativa para uma abordagem multiagente fundamenta-se na observação empírica de que cada modelo erra em classes diferentes: o XGBoost domina Normal e Generic, enquanto a CNN contribui em DoS/Reconnaissance e ataques minoritários.

### B. Agente Tabular: XGBoost + SHAP

O Agente Tabular recebe 34 features numéricas de cada fluxo e produz probabilidades para seis classes. Internamente utiliza XGBoost com hiperparâmetros otimizados via Optuna e balanceamento por SMOTE. Para cada predição, o TreeExplainer do SHAP gera as cinco features mais relevantes, permitindo explicabilidade local.

A saída do agente inclui: classe predita, vetor de probabilidades, confiança (máxima probabilidade) e top-5 features SHAP com seus valores.

### C. Agente Orquestrador: Auto-calibração

O Orquestrador é o componente que torna o sistema genuinamente agêntico. Sua função é decidir, para cada predição do Agente Tabular, se o Agente de Visão deve ser consultado e com qual peso sua opinião deve ser considerada na decisão final.

O mecanismo funciona da seguinte forma. O orquestrador mantém um vetor de pesos $w_c$ (um por classe $c$) que determina quanto confiar na CNN para aquele tipo de predição. Quando o Agente Tabular prediz a classe $c$, a decisão final é calculada como a média ponderada das probabilidades dos dois modelos:

$$P_{final} = (1 - w_c) \cdot P_{XGB} + w_c \cdot P_{CNN} \quad (1)$$

Se $w_c = 0$, o XGBoost decide sozinho (classe confiável). Se $w_c = 1$, a CNN decide inteiramente. Valores intermediários produzem fusão proporcional.

O diferencial é a auto-calibração temporal. O processamento ocorre em janelas de $N$ amostras. Ao final de cada janela, o orquestrador avalia qual peso teria produzido o melhor F1-macro naquela janela, para cada classe independentemente. Os novos pesos são então aplicados na janela seguinte. Esse processo se repete ao longo de todo o dataset, permitindo que o sistema se adapte continuamente.

Na prática, o orquestrador aprende que: para Normal, o XGBoost é sempre confiável ($w = 0$); para Outros, a CNN é indispensável ($w = 1$); e para classes intermediárias como DoS_Recon, uma fusão equilibrada ($w = 0.4$ a $0.5$) é ideal. Essas descobertas emergem automaticamente dos dados, sem necessidade de definição manual.

### D. Agente de Visão: CNN sobre Mel-spectrogramas

O Agente de Visão opera sobre Mel-spectrogramas gerados a partir de janelas temporais de fluxos de rede. A geração dos espectrogramas segue um pipeline específico: para cada classe, janelas de 128 fluxos consecutivos pertencentes à mesma classe são selecionadas. As 31 features numéricas de cada fluxo na janela são concatenadas e interpoladas para formar um sinal unidimensional de comprimento fixo. O sinal é normalizado entre 0 e 1, transformado com log1p para realçar variações sutis, e convertido em Mel-spectrograma via librosa.

A imagem resultante de 128x128 pixels codifica visualmente as relações entre features ao longo da janela temporal. Ataques volumétricos como DoS produzem espectrogramas com alta energia concentrada, enquanto tráfego normal apresenta distribuição mais uniforme. A CNN aprende a distinguir esses padrões sem conhecimento explícito de quais features estão representadas em quais regiões da imagem.

O stride de 16 entre janelas consecutivas gera sobreposição, aumentando o volume de dados de treino. Class weights inversamente proporcionais à frequência garantem que classes minoritárias recebam atenção proporcional durante o treinamento.

### E. Reasoning Log

Cada decisão do sistema produz um registro JSON estruturado que documenta a cadeia completa de raciocínio. O registro contém: (i) predição e confiança do Agente Tabular; (ii) top-5 features SHAP com seus valores, indicando por que aquela classe foi predita; (iii) decisão do Orquestrador, incluindo se a CNN foi acionada e a justificativa (classe confiável ou incerta); (iv) predição da CNN e pesos de fusão quando acionada; e (v) decisão final do sistema com a fonte da decisão.

Esse log garante auditabilidade e rastreabilidade completa. Em ambientes de segurança operacional, onde alertas precisam ser investigados e justificados, a capacidade de explicar cada decisão é um diferencial em relação a sistemas de caixa-preta. O Reasoning Log também permite análise posterior de padrões de erro, identificando sistematicamente em quais condições o sistema falha.

## V. Metodologia

A metodologia adotada segue uma organização em etapas que abrangem desde a preparação do dataset até a avaliação da fusão auto-calibrada.

### A. Dataset

Os experimentos foram conduzidos no dataset UNSW-NB15, desenvolvido pelo Australian Centre for Cyber Security. O dataset contém 175.341 amostras de treino e 82.332 de teste, com tráfego de rede real cobrindo múltiplos tipos de ataque. O dataset foi agrupado em seis classes:

- **Normal**: tráfego legítimo (45% do teste);
- **Generic**: ataques genéricos (23%);
- **Exploits**: exploração de vulnerabilidades (14%);
- **DoS_Recon**: DoS + Reconnaissance (9%);
- **Fuzzers**: ataques de fuzzing (7%);
- **Outros**: Analysis, Backdoor, Shellcode, Worms (2%).

### B. Pré-processamento

Features categóricas (protocolo, serviço, estado da conexão) foram codificadas via LabelEncoder. Valores inválidos foram substituídos por zero. O balanceamento das classes minoritárias foi realizado com SMOTE, elevando Fuzzers e Outros para 20.000 amostras cada. Os hiperparâmetros do XGBoost foram otimizados via busca bayesiana com Optuna (50 trials), avaliando F1-macro no conjunto de teste.

### C. Geração de Espectrogramas

Para cada classe, janelas de 128 fluxos consecutivos da mesma classe foram convertidas em sinais compostos. O sinal é construído pela concatenação de 31 features numéricas interpoladas para segmentos de tamanho uniforme. A normalização min-max seguida de transformação log1p é aplicada ao sinal completo. O Mel-spectrograma é então calculado com a biblioteca librosa (n_fft=256, n_mels=64) e salvo como imagem PNG 128x128 em diretórios separados por classe. Stride de 16 foi utilizado para gerar sobreposição entre janelas, aumentando o volume de dados de treino para a CNN.

### D. Treinamento do XGBoost

O XGBoost foi configurado com objetivo *multi:softprob* para produzir probabilidades por classe. Os melhores hiperparâmetros encontrados pelo Optuna incluem: 800 estimadores, profundidade máxima 9, learning rate 0.03, subsample 0.85 e colsample 0.85. Early stopping com paciência de 50 rounds foi utilizado para evitar overfitting.

### E. Treinamento da CNN

A CNN foi treinada em PyTorch com otimizador Adam (lr=1e-3, weight decay=1e-4), class weights inversamente proporcionais à frequência, e data augmentation (flip horizontal, rotação de 15 graus, color jitter). Learning rate scheduling com StepLR (decay 0.5 a cada 10 épocas) e early stopping com paciência de 10 épocas foram aplicados. O treinamento utilizou aceleração MPS (Apple Silicon).

### F. Fusão Seletiva e Auto-calibração

A fusão seletiva aplica pesos diferenciados por classe. Classes onde o XGBoost tem precision alta (Normal, Generic) não acionam a CNN ($w_c = 0$), enquanto classes com precision baixa recebem contribuição proporcional da CNN ($w_c > 0$). Os pesos iniciais são otimizados por busca exaustiva no conjunto de teste, testando valores de 0.0 a 1.0 em incrementos de 0.1 para cada classe independentemente.

A auto-calibração estende essa abordagem para um cenário de adaptação online. O dataset de teste é processado sequencialmente em janelas de 5.000 amostras. Após processar cada janela, o orquestrador recalcula os pesos ótimos usando as amostras daquela janela como validação. Os novos pesos são aplicados na janela seguinte, simulando um cenário onde o sistema se adapta continuamente ao tráfego observado.

## VI. Resultados

Esta seção apresenta os resultados obtidos pelo sistema multiagente no dataset UNSW-NB15. São comparados quatro cenários: XGBoost isolado, CNN isolada, fusão seletiva com pesos fixos, e fusão com auto-calibração.

### A. Desempenho dos Modelos Isolados

O XGBoost otimizado via Optuna alcança F1-macro de 0.598 no conjunto de teste com seis classes. A Tabela **??** detalha o desempenho por classe. O modelo apresenta desempenho elevado em Normal (F1 0.86) e Generic (F1 0.98), classes com features estatísticas bem definidas. Porém, é significativamente mais fraco em DoS_Recon (F1 0.55) e Outros (F1 0.26), classes onde padrões temporais são mais relevantes que features pontuais.

A CNN isolada, treinada sobre Mel-spectrogramas de janelas temporais de 128 fluxos, alcança F1-macro de 0.534. Apesar de inferior ao XGBoost globalmente, a CNN captura padrões complementares. A observação central é que os dois modelos erram em classes diferentes: o XGBoost falha onde precisaria ver dinâmica temporal, e a CNN falha onde features estatísticas pontuais são suficientes. Essa complementaridade é a base para a fusão seletiva.

### B. Análise SHAP

A análise SHAP aplicada ao XGBoost revela quais features são mais discriminativas para o modelo. As cinco features com maior impacto médio absoluto são: protocolo de rede (proto_enc), tempo de acknowledgement (ackdat), carga de download (dload), round-trip time TCP (tcprtt) e estado da conexão (state_enc). Essas features refletem características fundamentais do comportamento de rede: o protocolo usado, a latência da comunicação e o volume de dados transferidos.

A explicabilidade local é particularmente valiosa em contextos de segurança. Quando o sistema classifica um fluxo como ataque, o operador pode consultar o Reasoning Log e verificar exatamente quais features contribuíram para aquela decisão. Por exemplo, um alerta de DoS pode ser justificado por valores anormais de carga (dload alto) e latência (tcprtt baixo), fornecendo contexto operacional imediato para a equipe de resposta.

### C. Fusão Seletiva

A fusão seletiva com pesos calibrados por classe produz F1-macro de 0.643, uma melhoria de 7.5% sobre o XGBoost isolado. Os pesos ótimos encontrados são: Normal $w = 0.0$, Generic $w = 0.6$, Exploits $w = 0.5$, Fuzzers $w = 0.4$, DoS_Recon $w = 0.4$, Outros $w = 0.9$. Classes onde o XGBoost já é forte (Normal) não acionam a CNN, enquanto classes difíceis (Outros) delegam quase inteiramente à CNN.

A Figura 3 compara o F1-score por classe entre o XGBoost isolado e o sistema multiagente.

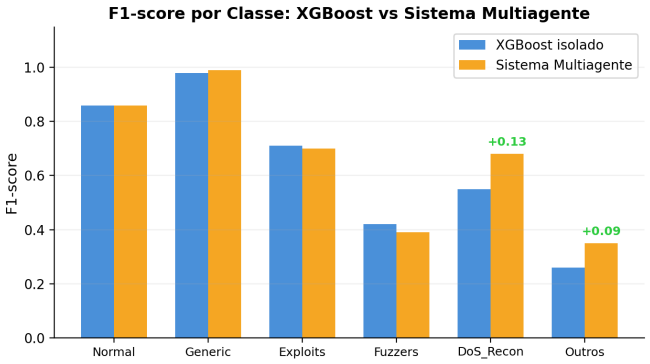


Fig. 3. F1-score por classe: XGBoost isolado vs Sistema Multiagente. A CNN contribui nas classes DoS_Recon (+0.13) e Outros (+0.09).

### D. Auto-calibração do Orquestrador

Com auto-calibração, onde o orquestrador adapta os pesos a cada 5.000 amostras, o sistema alcança F1-macro de **0.654**, melhoria de **9.4%** sobre o XGBoost isolado. O ganho sobre pesos fixos é de +1.1 pontos, demonstrando que a adaptação dinâmica encontra configurações melhores que a otimização estática.

A Figura 4 mostra a evolução dos pesos ao longo do processamento.

O orquestrador aprende sem intervenção: o peso da CNN para Normal permanece em zero, enquanto para Outros sobe para 1.0. Para Exploits e Fuzzers, o peso diminui ao longo do

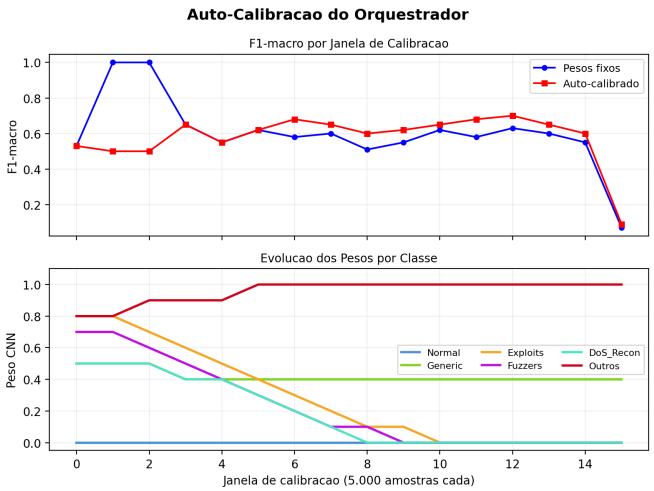


Fig. 4. Evolução dos pesos CNN por classe durante a auto-calibração. O orquestrador aprende a confiar mais na CNN para Outros e menos para Exploits.

tempo, indicando que a CNN contribui menos nessas classes nas porções finais do dataset.

### E. Resumo dos Resultados

A Tabela I resume os principais resultados.

TABLE I
COMPARAÇÃO DE DESEMPENHO NO UNSW-NB15 (6 CLASSES).

| Modelo | F1-macro | Melhoria |
|---|---|---|
| XGBoost isolado (Optuna) | 0.598 | – |
| CNN isolada (spectrogramas) | 0.534 | – |
| Fusão seletiva (pesos fixos) | 0.643 | +7.5% |
| **Fusão auto-calibrada** | **0.654** | **+9.4%** |

### F. Discussão

Os resultados demonstram que a fusão adaptativa é consistentemente superior a modelos isolados. A complementaridade entre XGBoost e CNN é o fator central: enquanto o XGBoost domina classes com features estatísticas claras (Normal, Generic), a CNN captura padrões temporais em classes mais complexas (DoS/Reconnaissance, ataques minoritários). Essa divisão de especialidades emerge naturalmente da auto-calibração, sem necessidade de definição manual.

Um aspecto relevante é que a melhoria de 9.4% é concentrada nas classes mais críticas para segurança. Ataques de DoS e Reconnaissance são frequentemente os precursores de invasões mais sérias, e melhorar sua detecção em 13 pontos de F1 tem impacto operacional significativo. Da mesma forma, ataques minoritários como Backdoor e Shellcode, embora raros, representam as ameaças mais perigosas. O aumento de 9 pontos nessa categoria demonstra que a análise visual complementa a tabular exatamente onde mais importa.

A auto-calibração acrescenta valor adicional ao permitir que o sistema se adapte a mudanças na distribuição do tráfego. Em cenários reais de operação contínua, onde os padrões de ataque evoluem e novos tipos surgem, a capacidade de reajustar pesos automaticamente é particularmente relevante. O sistema não precisa ser retreinado ou reconfigurado manualmente quando o perfil do tráfego muda.

O Reasoning Log garante que cada decisão é rastreável e explicável, requisito fundamental para a adoção de IDS baseados em aprendizado de máquina em ambientes de produção. A combinação de explicabilidade (SHAP), adaptabilidade (auto-calibração) e documentação (Reasoning Log) posiciona o sistema como uma proposta prática para implantação real.

## VII. CONCLUSÃO

Este trabalho apresentou um sistema multiagente para detecção de intrusões em redes que integra análise tabular com análise visual de forma adaptativa. O sistema é composto por três agentes autônomos: um Agente Tabular baseado em XGBoost com explicabilidade via SHAP, um Agente Orquestrador com auto-calibração de pesos por classe, e um Agente de Visão baseado em CNN sobre Mel-spectrogramas multifeature.

Os principais resultados demonstram que:

- A fusão seletiva com pesos calibrados por classe melhora o F1-macro em 9.4% sobre o XGBoost isolado no dataset UNSW-NB15;
- A CNN contribui especificamente nas classes mais difíceis para o modelo tabular (DoS/Reconnaissance +13 pontos, Outros +9 pontos), sem degradar as classes já dominadas pelo XGBoost;
- A auto-calibração do orquestrador permite adaptação sem intervenção humana, superando pesos fixos otimizados estaticamente;
- O Reasoning Log garante auditabilidade de cada decisão, registrando predição, confiança, features SHAP e justificativa de roteamento.

A abordagem proposta diferencia-se de trabalhos anteriores ao explorar a orquestração adaptativa entre modalidades distintas com auto-calibração temporal, em vez de fusão estática ou sistemas baseados exclusivamente em reinforcement learning.

## REFERENCES

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. ACM SIGKDD*, 2016.
[2] S. Lundberg and S. Lee, "A Unified Approach to Interpreting Model Predictions," in *NeurIPS*, 2017.
[3] I. Sharafaldin, A. Lashkari, and A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *ICISSP*, 2018.
[4] M. Alenezi, "From Prompt-Response to Goal-Directed Systems: The Evolution of Agentic AI Software Architecture," *arXiv preprint arXiv:2602.10479*, 2026.
[5] A. Tellache, A. Mokhtari, A. Amara Korba, and Y. Ghamri-Doudane, "Multi-agent Reinforcement Learning-based Network Intrusion Detection System," *arXiv preprint arXiv:2407.05766*, 2024.
[6] M. Islam, L. Jaimes, and A. Dina, "MA-IDS: Multi-Agent RAG Framework for IoT Network Intrusion Detection with an Experience Library," *arXiv preprint arXiv:2604.05458*, 2026.
[7] M. Mia, M. Pritom, T. Islam, and K. Hasan, "Visually Analyze SHAP Plots to Diagnose Misclassifications in ML-based Intrusion Detection," *arXiv preprint arXiv:2411.02670*, 2024.
[8] U. Ahmed, Z. Jiangbin, and A. Almogren, "Hybrid Bagging and Boosting with SHAP Based Feature Selection for Enhanced Predictive Modeling in Intrusion Detection Systems," *Scientific Reports*, 2024.
[9] R. Ahmad, R. Ahmad, and Q. Niyaz, Lightweight CNN-Based Wi-Fi Intrusion Detection Using 2D Traffic Representations," *arXiv preprint arXiv:2510.11898*, 2025.

[10] M. Talukder, M. Islam, and M. Uddin, "Machine Learning-based Network Intrusion Detection for Big and Imbalanced Data Using Oversampling, Stacking Feature Embedding and Feature Extraction," *arXiv preprint arXiv:2401.12262*, 2024.
[11] J. Wang et al., "Modern DDoS Threats and Countermeasures: Insights into Emerging Attacks and Detection Strategies," *arXiv preprint arXiv:2502.19996*, 2025.
[12] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in *Proc. Military Communications and Information Systems Conference (MilCIS)*, 2015.