

Predição Antecipada de Congestionamento TCP com Telemetria de Transporte do Kernel Linux

Everton Vilhena Cardoso, Adrian A. C. Alanes, Samuel Baraldi Mafra, Felipe A. P. de Figueiredo

Resumo—Este artigo investiga a predição antecipada de congestionamento TCP a partir de telemetria da pilha de transporte do kernel Linux observada no emissor. Em vez de detectar se uma conexão já está congestionada, a tarefa é formulada como problema causal de alerta antecipado: dadas medições recentes do lado do emissor a partir de uma janela não congestionada, prever se o congestionamento ocorrerá em um horizonte futuro próximo. A metodologia filtra amostras atualmente congestionadas, remove variáveis com vazamento futuro, constrói sequências temporais estritas e divide os dados por execuções experimentais. Cinco famílias de modelos são comparadas sob o mesmo protocolo: MLP com última janela, GRU, CNN 1D, LSTM e HistGradientBoosting com resumos temporais. Os resultados mostram que a telemetria TCP recente contém sinal preditivo e que resumos temporais podem ser competitivos com modelos neurais temporais no regime atual de dados limitados. Os achados indicam a viabilidade inicial de preditores de alerta antecipado no lado do emissor, embora validação mais ampla seja necessária antes da implantação operacional.

Palavras-Chave—Congestionamento TCP, predição antecipada, telemetria de transporte, métricas do kernel Linux.

Abstract—This paper investigates early prediction of TCP congestion using Linux transport-stack telemetry observed at the sender. Instead of detecting whether a connection is already congested, the task is formulated as a causal early-warning problem: given recent sender-side measurements from a non-congested window, predict whether congestion will occur within a short future horizon. The methodology filters currently congested samples, removes future-leaking variables, builds strict temporal sequences, and splits data by experimental runs. Five model families are compared under the same protocol: last-window MLP, GRU, CNN 1D, LSTM, and HistGradientBoosting with temporal summaries. Results show that recent TCP telemetry contains predictive signal and that temporal summaries can be competitive with neural temporal models in the current limited-data regime. The findings support the feasibility of sender-side early-warning predictors, while broader validation remains necessary before operational deployment.

Keywords—TCP congestion, early prediction, transport telemetry, Linux kernel metrics.

Everton Vilhena Cardoso, Adrian A. C. Alanes, Samuel Baraldi Mafra e Felipe A. P. de Figueiredo, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, Brasil. E-mails: {everton.vilhena, adrian}@mtel.inatel.br, {samuelbmafra, felipe.figueiredo}@inatel.br.

Este trabalho foi apoiado pelo CNPq, processos 31470/2021-1, 403827/2021-3 e 306199/2025-4; pelos projetos XGM-AFCCT-2024-2-5-1, XGM-AFCCT-2024-5-1-2 e XGM-AFCCT-2024-9-1-1, apoiados pelo xGMobile – EMBRAPII-Inatel Competence Center on 5G and 6G Networks, com recursos financeiros do PPI IoT/Manufatura 4.0 do MCTI, concedidos pela RNP, com recursos do MCTIC, processo 01245.020548/2021-07; pelo projeto Brazil 6G do Radiocommunication Reference Center do Inatel; pelo projeto Semear, apoiado pela FAPESP, processo 22/09319-9; pelo projeto SAMURAI, apoiado pela FAPESP, processo 20/05127-2, e pela FAPEMIG, processos APQ-04523-23, APQ-05305-23, APQ-03162-24, APQ-01558-24 e RED-00194-23; pelo xGMobile, processo Grant PPE-00124-23; e pela FINEP, contrato 01.25.0883.00.

I. INTRODUÇÃO

O controle de congestionamento TCP é essencial para a estabilidade de redes IP. Algoritmos clássicos e modernos, como CUBIC [1] e BBR [2], ajustam janela, taxa de envio, *pacing* e temporizadores a partir de sinais observados durante a transmissão [3], como RTT, perdas, retransmissões e taxas de entrega [4], [5], [6]. Assim, a atuação do emissor é predominantemente reativa: as decisões são tomadas após a manifestação de evidências de degradação do caminho.

Este artigo investiga uma formulação complementar: prever antecipadamente a ocorrência de congestionamento TCP a partir da telemetria disponível no emissor. A lacuna científica explorada é distinguir predição prospectiva de detecção do estado corrente. Para isso, o modelo utiliza apenas observações disponíveis até o instante de decisão e estima se ocorrerá congestionamento em um horizonte futuro curto. Essa formulação evita que o modelo aprenda apenas sintomas já manifestos e aproxima o problema de um sistema de alerta antecipado.

A hipótese avaliada é que métricas da pilha TCP do kernel Linux, incluindo RTT, variação de RTT, janela de congestionamento, taxa de entrega, *pacing*, retransmissões e indicadores de limitação, contém sinal preditivo para antecipar congestionamento em cenário controlado. Neste estudo inicial, considera-se horizonte futuro de 1 s, equivalente a quatro passos de amostragem de 250 ms. Esse horizonte foi escolhido por representar uma antecedência curta, porém operacionalmente útil, antes da consolidação do evento de congestionamento.

As contribuições deste trabalho são: (i) formulação causal de alerta antecipado com filtragem de janelas já congestionadas; (ii) separação entre congestionamento corrente e ocorrência futura com rótulos construídos a partir de evidências de fila no gargalo; (iii) comparação entre MLP, GRU, CNN 1D, LSTM e HistGradientBoosting com resumos temporais; e (iv) avaliação por métricas por janela e orientadas a eventos.

II. FUNDAMENTAÇÃO TÉCNICA E TRABALHOS RELACIONADOS

A. Relação com controle e aprendizado em redes

Trabalhos como TCP ex Machina [7], PCC Vivace [8] e abordagens de aprendizado por reforço para controle de congestionamento [9] buscam aprender diretamente políticas de envio ou de escolha de taxa. A proposta aqui é distinta: o TCP permanece inalterado e o modelo não atua sobre o controle. Em vez disso, estima a probabilidade de congestionamento futuro a partir de telemetria causal observada no emissor. A lacuna abordada é a ausência de uma avaliação isolada de predição antecipada: antes de modificar o controle de congestionamento,

é necessário verificar se há sinal preditivo disponível antes da consolidação do evento.

Seja y_t^{cur} o rótulo binário que indica congestionamento no instante corrente. Na construção do conjunto supervisionado, são mantidas apenas janelas com $y_t^{cur} = 0$, isto é, janelas ainda não congestionadas. O rótulo de saída indica a ocorrência de congestionamento em um horizonte futuro de 1 s. Dessa forma, o modelo é avaliado como mecanismo de alerta antecipado, e não como detector do estado atual ou como política de controle.

B. Congestionamento TCP e sinais observáveis

O congestionamento em redes TCP emerge quando a taxa agregada de envio se aproxima ou excede a capacidade efetiva do caminho, levando ao acúmulo de filas, aumento de atraso, variação de RTT, perdas, retransmissões e redução da taxa útil de entrega. Do ponto de vista do emissor, parte desses fenômenos é indiretamente observável por meio de métricas da pilha de transporte do kernel Linux, que refletem tanto propriedades do caminho quanto decisões internas do controle de congestionamento [1], [2], [10].

Entre os indicadores potencialmente informativos estão as estatísticas de RTT, medidas de variação temporal, estimativas de taxa de entrega, taxa de *padding*, janela de congestionamento, limiares de crescimento, bytes ainda não enviados, limitação por janela de recepção e variáveis relativas à trajetória recente. Em conjunto, tais métricas representam uma visão operacional da conexão no nível do sistema operacional, sem exigir instrumentação explícita no núcleo da rede [1], [2].

A predição antecipada é mais difícil que a detecção do estado corrente porque os sintomas mais evidentes do congestionamento podem ainda não estar presentes. O desafio científico está em identificar padrões de transição: pequenas elevações de atraso, mudanças de inclinação, instabilidade na taxa de entrega, redução relativa de janela ou acúmulo gradual de fila podem anteceder eventos futuros, mas também podem ocorrer em situações benignas. Por isso, a formulação experimental deve evitar vazamento de informação e deve avaliar se os modelos aprendem sinais precursores reais, e não atalhos estatísticos [11], [9].

C. Predição baseada em telemetria de transporte

Embora métricas de fluxo, atraso e estatísticas agregadas sejam amplamente usadas em aprendizado de máquina para redes [12], muitos estudos tratam inferência, detecção ou otimização de desempenho a partir de sinais já observados. A contribuição aqui está na formulação causal da tarefa: as entradas incluem apenas métricas disponíveis no emissor até o instante de decisão, enquanto o rótulo descreve a ocorrência futura do evento [9], [11].

D. Modelos temporais e resumo de trajetória

Séries temporais de métricas TCP podem ser modeladas de diferentes formas. Redes recorrentes, como Redes Neurais Gated Recurrent Unit (GRU) e *Long Short-Term Memory* (LSTM), processam diretamente sequências de medições e podem aprender dependências temporais de curta duração [13],

[14]. Redes Neurais Convolucionais (CNNs) unidimensionais exploram padrões locais de transição por meio de filtros temporais, oferecendo uma alternativa não recorrente para a representação de sequências [9]. Modelos tabulares como, por exemplo, *HistGradientBoosting* (HGB), por outro lado, não processam a sequência bruta, mas podem receber estatísticas agregadas da trajetória recente, como último valor, média, desvio, mínimo, máximo, variação e inclinação [15].

A comparação entre essas famílias é cientificamente relevante porque testa se a estrutura temporal precisa ser aprendida por arquiteturas sequenciais ou se resumos simples capturam informações suficientes. Em regimes de dados pequenos, modelos tabulares fortes podem apresentar melhor generalização do que redes neurais mais expressivas, sobretudo quando o sinal preditivo está concentrado em tendências e estatísticas de curta duração [15], [13], [14].

III. METODOLOGIA EXPERIMENTAL

A. Topologia e ambiente experimental

Os experimentos foram conduzidos em topologia dumbbell sobre Mininet [16] (kernel Linux 5.19.0-46, Mininet 2.3.1b4), com dois fluxos TCP compartilhando um enlace de gargalo, conforme a Fig. 1. O fluxo A liga o emissor h1 ao receptor h2; o fluxo B liga h3 a h4. O gargalo opera a 100 Mbit/s com disciplina pfifo [17] (tail-drop) e ECN desativado, sem mecanismos ativos de gerenciamento de filas. O tamanho de fila variou entre 25 e 200 pacotes; atraso e perda foram configurados por netem nos enlaces de acesso.

O fluxo A foi configurado, em execuções experimentais distintas, com CUBIC [1] ou LEDBAT [10] como algoritmo de controle de congestionamento. O fluxo B permaneceu sempre configurado com CUBIC, atuando como fluxo concorrente de referência. Essa configuração permite comparar o comportamento do fluxo A sob diferentes algoritmos de controle e, em particular, observar o comportamento de *yield* do LEDBAT frente a um concorrente CUBIC sob diferentes condições de fila e atraso.

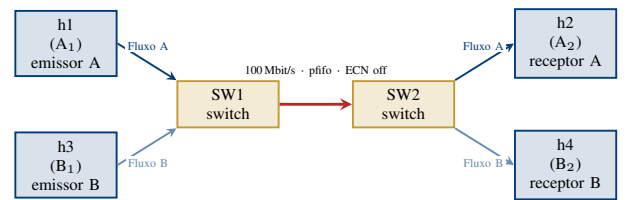


Fig. 1. Topologia *dumbbell* utilizada. Dois fluxos TCP fim a fim compartilham o enlace central: fluxo A entre h1 e h2 e fluxo B entre h3 e h4. O enlace central opera a 100 Mbit/s, com pfifo e ECN desativado.

A escolha dessa topologia se justifica por concentrar a disputa por recursos em um enlace compartilhado, permitindo induzir eventos de congestionamento controlados e reprodutíveis. Como a estrutura da rede permanece fixa, as variações observadas nas métricas do emissor podem ser relacionadas à evolução temporal da conexão e à ocorrência futura de congestionamento.

B. Construção do conjunto de dados supervisionado

O conjunto de dados foi construído a partir de uma cadeia automatizada de coleta, pré-processamento e rotulagem das

execuções experimentais. Durante cada execução, os emissores TCP foram monitorados periodicamente, gerando séries temporais de métricas de transporte associadas a cada fluxo. Em seguida, rotinas de processamento consolidaram os registros por execução, fluxo, papel da conexão e algoritmo de controle de congestionamento, alinharam as amostras no tempo, removeram inconsistências e derivaram janelas temporais rotuladas.

A partir dessa base, foi definido um subconjunto supervisionado de alerta antecipado. A tarefa é formulada como classificação binária com duas classes: *No Future Congestion* e *Future Congestion*. A classe *No Future Congestion* representa janelas sem congestionamento futuro no horizonte de 1s, enquanto a classe *Future Congestion* representa janelas nas quais o congestionamento ocorre dentro desse horizonte. Foram consideradas apenas janelas nas quais o fluxo ainda não está rotulado como congestionado no instante corrente.

Para cada instante t , seja $\mathbf{x}_t \in \mathbb{R}^d$ o vetor de atributos extraído da janela temporal correspondente. Nos modelos temporais, a entrada é composta por uma sequência de L janelas consecutivas:

$$\mathbf{X}_t^{(L)} = [\mathbf{x}_{t-L+1}, \dots, \mathbf{x}_t], \quad \hat{p}_t = f(\mathbf{X}_t^{(L)}). \quad (1)$$

A saída $\hat{p}_t$ representa a probabilidade estimada de congestionamento futuro, e a decisão binária é obtida por um limiar τ escolhido no conjunto de validação. Os rótulos de congestionamento corrente e futuro são provenientes da etapa de geração e rotulagem do conjunto de dados; neste trabalho, eles são usados para construir a tarefa supervisionada e comparar os modelos sob a mesma definição de evento.

C. Particionamento experimental

As amostras foram particionadas por execução experimental, e não por amostragem aleatória de linhas. Como havia cinco execuções disponíveis, adotou-se uma divisão 3/1/1: três execuções para treino, uma para validação e uma para teste. Essa escolha preserva execuções independentes para seleção de hiperparâmetros e avaliação final, ao mesmo tempo em que mantém a maior parte dos dados para treinamento. Essa estratégia reduz o risco de vazamento temporal, pois janelas próximas no tempo ou provenientes da mesma execução tendem a compartilhar condições de rede semelhantes.

A Tabela I resume a distribuição do subconjunto supervisionado após a filtragem de alerta antecipado. Na tabela, “Sem evento” corresponde à classe negativa e “Com evento” corresponde à classe positiva. Em termos de janelas rotuladas, as partições correspondem a aproximadamente 60,0% para treino, 21,2% para validação e 18,8% para teste. As classes permanecem aproximadamente balanceadas no treino e na validação, enquanto o teste apresenta leve predominância de janelas com evento. Adotou-se $L = 8$ como configuração principal por compromisso entre contexto temporal e tamanho efetivo da base; nessa configuração, as sequências úteis correspondem a aproximadamente 61,3%, 22,1% e 16,6%, respectivamente.

D. Atributos de entrada e prevenção de vazamento

A seleção de atributos teve dois objetivos: manter apenas informações disponíveis ao emissor TCP e evitar qualquer

TABELA I
DISTRIBUIÇÃO DAS CLASSES E SEQUÊNCIAS ÚTEIS PARA $L = 8$.

Partição	Exec.	No Future Cong.	Future Cong.	Seq. úteis
Treino	1–3	3 985	3 984	2 397
Validação	4	1 419	1 403	864
Teste	5	1 160	1 334	650

forma de vazamento de informação futura. A partir de 49 métricas candidatas, foram removidos atributos não aplicáveis ao regime experimental avaliado, variáveis diretamente associadas ao tempo até o próximo evento de congestionamento e colunas constantes no conjunto de treino. Esse processo resultou nos 33 atributos finais resumidos na Tabela II.

Os atributos selecionados representam sinais complementares da evolução recente da conexão. Em vez de depender de uma única métrica, como perda ou RTT, a entrada combina informações sobre atraso, enfileiramento, reação da janela TCP, taxa efetiva de entrega, ritmo de envio, desvios em relação ao histórico recente e possíveis limitações do emissor ou receptor.

A padronização dos atributos foi ajustada exclusivamente no conjunto de treino e posteriormente aplicada aos conjuntos de validação e teste. Esse procedimento impede que estatísticas do teste influenciem a transformação dos dados e mantém a comparação entre modelos sob a mesma representação de entrada.

E. Representação temporal das amostras

Após a seleção, cada janela foi representada por 33 atributos derivados das métricas de transporte. Para os modelos temporais, foram construídas sequências estritas de comprimento L , formadas por janelas consecutivas do mesmo fluxo e da mesma partição experimental, espaçadas em 250 ms. O rótulo da sequência é definido pela última janela, preservando a relação causal entre histórico observado e evento futuro.

Adotou-se $L = 8$ como compromisso entre contexto temporal e tamanho efetivo da base. Como cada janela agrega 1 s e o deslocamento é de 250 ms, uma sequência cobre aproximadamente 2,75 s antes do instante de decisão. Esse intervalo permite observar tendências curtas em RTT, sinais indiretos de enfileiramento, taxa de entrega, janela de congestionamento e pacing antes do horizonte futuro de 1 s, mantendo 2 397 sequências de treino, 864 de validação e 650 de teste, conforme a Tabela I. Valores menores de L reduzem o histórico disponível; valores maiores reduzem o conjunto útil no regime atual de dados.

IV. ESTRATÉGIA DE MODELAGEM E AVALIAÇÃO

A. Modelos e representações

Os modelos foram escolhidos para comparar três formas de explorar a trajetória recente da conexão: última janela, sequência bruta e resumos temporais. O MLP usa apenas os 33 atributos da última janela e serve como baseline não temporal. GRU e LSTM avaliam o uso de recorrência para dependências temporais curtas, enquanto a CNN 1D avalia padrões locais de

TABELA II
GRUPOS FUNCIONAIS DOS 33 ATRIBUTOS DE ENTRADA USADOS PELOS MODELOS.

Grupo	Atributos	Observação
Atraso	rtt, rttvar, min_rtt, rtt_ratio, rttvar_ratio, queue_delay_proxy	atraso e variação causal
Janela	snd_cwnd, snd_ssthresh, snd_wnd, rcv_space, cwnd_ratio	reação do emissor/receptor
Entrega/pacing	delivery_rate, pacing_rate, max_pacing_rate, delivery_rate_ratio, pacing_rate_ratio	taxa efetiva e ritmo
Pacotes/perdas	unacked, sacked, lost, retrans, retrans_out, total_retrans, unacked_ratio, lost_delta, retrans_delta	pressão e perdas observadas
Limitações/estado	notsent_bytes, busy_time, rwnd_limited, sndbuf_limited, app_limited, ca_state, advmss, send_limited_ratio	limites locais e estado

transição ao longo da sequência. O HGB representa um modelo tabular forte aplicado a resumos temporais da trajetória.

O MLP recebe 33 atributos; GRU, LSTM e CNN 1D recebem a sequência completa 8×33 ; e o HGB recebe 231 atributos, obtidos por sete resumos de cada atributo ao longo da sequência: último valor, média, desvio padrão, mínimo, máximo, variação e inclinação. Assim, a comparação com HGB também deve ser interpretada como comparação de representação, pois ele não processa a sequência bruta.

B. Métricas

Cada modelo estima a probabilidade de congestionamento futuro, convertida em uma decisão binária por um limiar escolhido na validação. A avaliação por janela usa acurácia, acurácia balanceada, precisão, revocação, F1 score, *Area Under the Receiver Operating Characteristic* (AUROC), *Area Under the Precision-Recall Curve* (AUPRC) e matriz de confusão. As métricas centrais de comparação são acurácia balanceada, F1 score, revocação e AUPRC, pois refletem melhor a detecção da classe positiva e o ranqueamento probabilístico.

Também foram usadas métricas orientadas a eventos. Um evento é um bloco de janelas positivas consecutivas. O *strict pre-label recall* considera alarmes antes do início do evento; o *pre-or-onset recall* também aceita alarmes no início; o *during-event recall* mede detecções durante o evento; e o *operational recall* agrega alarmes úteis antes ou durante o evento. A taxa de falsos positivos isolados mede alarmes distantes de qualquer evento real.

C. Pontos de operação e seleção

Foram avaliados três pontos de operação para GRU, CNN 1D, LSTM e HGB. O modo base usa a configuração padrão selecionada na validação; o modo balanceado prioriza métricas por janela, como acurácia balanceada e F1; e o modo sensível a eventos prioriza métricas operacionais de alerta antecipado, favorecendo maior revocação de eventos. Hiperparâmetros e limiares foram escolhidos exclusivamente na validação, e o teste foi usado apenas para avaliação final.

As redes neurais foram treinadas com entropia cruzada binária com logits, ponderação da classe positiva estimada no treino, Adam, lote de 256 amostras e semente fixa igual a 42. O HGB seguiu a mesma lógica de seleção, usando os 231 resumos temporais como entrada.

V. RESULTADOS

Todas as métricas foram calculadas no conjunto de teste, composto por 650 sequências, usando $L = 8$, 33 atributos de entrada, divisão por execuções experimentais e limiares definidos no conjunto de validação. A Tabela III reúne, em uma única comparação, métricas clássicas por janela e métricas orientadas a eventos.

A. Desempenho por janela

Nas métricas por janela da Tabela III, o HGB balanceado obteve o melhor desempenho global, com F1 de 0,7692 e acurácia balanceada de 0,7511. Esse modelo também apresentou alta revocação, 0,9150, reduzindo falsos negativos em relação à GRU balanceada, embora haja aumento de falsos positivos.

O modelo GRU sensível a eventos apresentou a maior AUPRC, 0,7812, indicando o melhor ranqueamento probabilístico entre os modelos avaliados. O HGB base, por outro lado, apresentou a maior precisão e o menor número de falsos positivos, caracterizando um ponto de operação mais conservador.

B. Desempenho orientado a eventos

Nas métricas orientadas a eventos da Tabela III, a CNN 1D sensível a eventos e o HGB balanceado alcançaram o maior *operational recall*, ambos com 0,9706. A CNN 1D sensível a eventos obteve o maior *pre-or-onset recall*, 0,9510, indicando maior sensibilidade temporal para alarmes antes ou no início dos eventos.

O HGB balanceado manteve o mesmo *operational recall* máximo, mas com melhor desempenho global por janela. Assim, os resultados mostram um compromisso entre sensibilidade operacional e desempenho discriminativo por janela.

VI. DISCUSSÃO

Os resultados indicam que não há um único modelo dominante para todos os critérios. O HGB balanceado apresentou o melhor compromisso global por janela, a GRU sensível a eventos obteve o melhor ranqueamento probabilístico e a CNN 1D sensível a eventos alcançou maior sensibilidade temporal. Assim, a escolha do modelo depende do objetivo operacional: desempenho global, calibração de limiares ou antecipação de eventos.

O desempenho do HGB deve ser interpretado à luz da representação usada. Esse modelo não processa a sequência bruta,

TABELA III
COMPARAÇÃO GLOBAL DOS MODELOS NO CONJUNTO DE TESTE.

Modelo	Modo	Métricas por janela						Métricas orientadas a eventos					
		F1	BACC	Prec.	Rec.	AUROC	AUPRC	FP	FN	Op. rec.	Pré/início	Pré estrito	FP isol.
MLP	Últ. janela	0,6903	0,6930	0,6531	0,7320	0,7629	0,7150	119	82	0,8627	0,8137	0,2843	0,0092
GRU	Base	0,7245	0,7283	0,6882	0,7647	0,8004	0,7614	106	72	0,8627	0,8235	0,2647	0,0062
GRU	Balanceado	0,7543	0,7448	0,6762	0,8529	0,8052	0,7661	125	45	0,9118	0,8627	0,2647	0,0077
GRU	Sens. eventos	0,7251	0,6968	0,6170	0,8791	0,8094	0,7812	167	37	0,9314	0,8922	0,3431	0,0077
CNN 1D	Base	0,7145	0,6808	0,6018	0,8791	0,7865	0,7445	178	37	0,9314	0,9118	0,3333	0,0169
CNN 1D	Balanceado	0,7289	0,7195	0,6579	0,8170	0,7969	0,7541	130	56	0,9216	0,8922	0,2745	0,0123
CNN 1D	Sens. eventos	0,7328	0,7012	0,6156	0,9052	0,8048	0,7567	173	29	0,9706	0,9510	0,3431	0,0169
LSTM	Base	0,7337	0,7279	0,6703	0,8105	0,7984	0,7681	122	58	0,8725	0,8235	0,2451	0,0046
LSTM	Balanceado	0,7258	0,6970	0,6164	0,8824	0,7924	0,7580	168	36	0,9412	0,9216	0,3235	0,0154
LSTM	Sens. eventos	0,7247	0,6833	0,5958	0,9248	0,8003	0,7663	192	23	0,9608	0,9412	0,3431	0,0154
HGB	Base	0,6724	0,7013	0,7036	0,6438	0,8000	0,7697	83	109	0,7843	0,6961	0,2157	0,0077
HGB	Balanceado	0,7692	0,7511	0,6635	0,9150	0,8061	0,7584	142	26	0,9706	0,9314	0,3235	0,0092
HGB	Sens. eventos	0,7548	0,7331	0,6472	0,9052	0,8094	0,7659	151	29	0,9608	0,9216	0,3333	0,0092

mas 231 resumos temporais da trajetória recente, incluindo o último valor, média, dispersão, variação e inclinação. Portanto, seu bom desempenho não invalida modelos sequenciais; indica que, no regime atual de dados, tendências curtas das métricas de transporte já concentram sinal preditivo relevante.

A avaliação orientada a eventos também mostra que nem todo falso positivo tem o mesmo significado: alarmes próximos de eventos reais podem ser úteis como alerta antecipado, enquanto falsos positivos isolados são mais problemáticos.

Os resultados devem ser tratados como evidência inicial em ambiente controlado. Para $L = 8$, há 2 397 sequências de treino, 864 de validação e 650 de teste, o que limita a robustez da comparação entre modelos neurais e tabulares. Além disso, o teste usa uma única execução não observada, sem múltiplas sementes completas nem intervalos de confiança.

A avaliação atual também não mede custo computacional, latência de inferência, uso de memória ou consumo energético, aspectos relevantes para implantação no emissor TCP.

Avaliações futuras devem incluir mais execuções, múltiplos horizontes, validação por execução, análise estatística e maior detalhamento da regra de rotulagem.

VII. CONCLUSÃO E TRABALHOS FUTUROS

Este artigo formulou a predição antecipada de congestionamento TCP como uma tarefa supervisionada, baseada em telemetria de transporte do kernel Linux observada no emissor. A metodologia filtrou janelas já congestionadas, removeu atributos com vazamento de informação futura, construiu sequências temporais estritas e adotou divisão por execuções experimentais.

Os resultados indicam que o histórico recente das métricas de transporte contém sinal preditivo para o horizonte de 1 s avaliado. No conjunto atual, o HGB balanceado obteve o melhor desempenho global por janela, enquanto a GRU sensível a eventos apresentou o melhor ranqueamento probabilístico, e a CNN 1D sensível a eventos alcançou maior sensibilidade temporal. Esses achados sugerem que resumos temporais simples podem competir com modelos neurais temporais em regimes de dados limitados.

Como trabalhos futuros, pretende-se ampliar a campanha experimental com mais execuções, cenários, algoritmos de

controle e condições de rede. Também serão avaliados múltiplos horizontes, validação por execução, intervalos de confiança, interpretabilidade, combinações em *ensemble*, custo computacional, latência, memória, consumo energético e políticas de controle proativo.

REFERÊNCIAS

- [1] S. Ha, I. Rhee e L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” *ACM SIGOPS Operating Systems Review*, vol. 42, n. 5, pp. 64–74, 2008.
- [2] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh e V. Jacobson, “BBR: Congestion-Based Congestion Control,” *Communications of the ACM*, vol. 60, n. 2, pp. 58–66, 2017.
- [3] V. Jacobson, “Congestion avoidance and control,” in *Proc. ACM SIGCOMM Symposium*, pp. 314–329, 1988.
- [4] M. Allman, V. Paxson e E. Blanton, “TCP Congestion Control,” RFC 5681, IETF, 2009.
- [5] V. Paxson, M. Allman, J. Chu e M. Sargent, “Computing TCP’s Retransmission Timer,” RFC 6298, IETF, 2011.
- [6] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert e R. Scheffenegger, “CUBIC for Fast and Long-Distance Networks,” RFC 9438, IETF, 2023.
- [7] K. Winstein e H. Balakrishnan, “TCP ex Machina: computer-generated congestion control,” in *Proc. ACM SIGCOMM Conference*, pp. 123–134, 2013.
- [8] M. Dong *et al.*, “PCC Vivace: Online-learning congestion control,” in *Proc. USENIX NSDI*, 2018.
- [9] N. Jay, N. Rotman, P. Godfrey, M. Schapira e A. Tamar, “A deep reinforcement learning perspective on Internet congestion control,” in *Proc. International Conference on Machine Learning*, 2019.
- [10] S. Shalunov, G. Hazel, J. Iyengar e M. Kuehlewind, “Low Extra Delay Background Transport,” RFC 6817, IETF, 2012.
- [11] F. Y. Yan *et al.*, “Pantheon: the training ground for Internet congestion-control research,” in *Proc. USENIX Annual Technical Conference*, pp. 731–743, 2018.
- [12] Y. Sun, M. Peng, Y. Zhou, Y. Huang e S. Mao, “Application of machine learning in wireless networks: Key techniques and open issues,” *IEEE Communications Surveys & Tutorials*, vol. 21, n. 4, pp. 3072–3108, 2019.
- [13] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proc. EMNLP*, pp. 1724–1734, 2014.
- [14] S. Hochreiter e J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, n. 8, pp. 1735–1780, 1997.
- [15] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, vol. 29, n. 5, pp. 1189–1232, 2001.
- [16] B. Lantz, B. Heller e N. McKeown, “A network in a laptop: rapid prototyping for software-defined networks,” in *Proc. ACM HotNets*, 2010.
- [17] Y. Gong, D. Rossi, C. Testa, S. Valenti e M. D. Täht, “Fighting the bufferbloat: on the coexistence of AQM and low priority congestion control,” in *Proc. 5th IEEE International Traffic Monitoring and Analysis Workshop (TMA)*, 2013.