INSTITUTO FEDERAL DE ALAGOAS

CAMPUS MACEIÓ

CURSO DE BACHAREL EM SISTEMAS DE INFORMAÇÃO

LEONARDO VICTOR FERNANDES FERREIRA

PREDIÇÃO DE CRIMES VIOLENTOS CONTRA O PATRIMÔNIO EM MACEIÓ E ARAPIRACA USANDO CONVLSTM

MACEIÓ, AL

2026

LEONARDO VICTOR FERNANDES FERREIRA

PREDIÇÃO DE CRIMES VIOLENTOS CONTRA O PATRIMÔNIO EM MACEIÓ E ARAPIRACA USANDO CONVLSTM

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação no Instituto Federal de Alagoas, *Campus* Maceió, como requisito parcial para a obtenção de grau de Bacharel(a) em Sistemas de Informação.

Orientador(a): Prof. Dr. Leonardo Melo de Medeiros.

MACEIÓ, AL

2026

Dados Internacionais de Catalogação na Publicação
Instituto Federal de Alagoas
***Campus* Maceió**
Biblioteca Benevides Monte

006.32
F383p Ferreira, Leonardo Victor Fernandes.

Predição de crimes violentos contra o patrimônio em Maceió e Arapiraca usando o ConvLSTM [recurso eletrônico] / Leonardo Victor Fernandes Ferreira. – Dados eletrônicos (1 arquivo : 1,42 MB). – 2026.

Sistema requerido: Adobe Acrobat Reader.
Modo de acesso: Internet.
Orientação: Prof. Dr. Leonardo Melo de Medeiros.
Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Instituto Federal de Alagoas, *Campus* Maceió, Maceió, 2026.

1. Predição criminal. 2. ConvLSTM. 3. Crimes contra o patrimônio. 4. Segurança pública – Maceió. 5. Segurança pública – Arapiraca. I. Título.

Franciane Monick Gomes de França
Bibliotecária – CRB 4/1831

INSTITUTO FEDERAL
Alagoas

BRASIL
PAÍS RICO É PAÍS SEM POBREZA

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE ALAGOAS – IFAL
DIRETORIA DE ENSINO - CAMPUS MACEIÓ
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ATA DE DEFESA DO TCC
Anexo à Portaria Nº 1483/GR, de 19/09/2012

Aos 23 dia(s) do mês de MAIO do ano de 2016, às 2030 foi realizada no no CAMPUS MACEIÓ, a solenidade de defesa de TCC de LEONARDO VICTOR FERNANDES FERREIRA matrícula 20170022096, com o tema PREDIÇÃO DE CRIMES VIOLENTOS USANDO CONV LSTM como pré-requisito para a conclusão do Curso BACHARELADO EM SISTEMAS DE INFORMAÇÃO.

PARECER FINAL
APROVADO SEM RESTRIÇÃO.

ALUNO(S)

ASSINATURA DA BANCA EXAMINADORA
Orientador(a)/Presidente da Banca
LEONARDO MELO DE MEDEIROS
/ Avaliador 1
JOILTON OLIVEIRA DE PAULA
BRUNO JACINTO DUARTE DA COSTA

AGRADECIMENTOS

Agradeço ao meu orientador, Dr. Leonardo Melo de Medeiros, pela orientação e pelo suporte ao longo do desenvolvimento deste trabalho.

À minha esposa, minha companheira e minha maior fortaleza: obrigado por estar ao meu lado em cada etapa desta jornada. Nos momentos de cansaço e incerteza, sua presença, seu carinho e sua fé em mim foram o que me deram força para continuar. Este trabalho também é seu.

À minha família, pelo apoio incondicional, pelas palavras de incentivo e por compreender as ausências que a vida acadêmica exige. Vocês são a base sobre a qual tudo o que construo está alicerçado.

Agradeço também ao Prof. Msc. Breno Jacinto Duarte da Costa, por me fazer enxergar a educação de uma forma completamente diferente, proporcionando uma compreensão mais profunda sobre o seu funcionamento e o seu papel transformador.

Ao Instituto Federal de Alagoas – IFAL, *Campus* Maceió, por oferecer um ambiente de ensino de qualidade e por proporcionar as condições necessárias para o desenvolvimento desta pesquisa. Estendo o agradecimento a todos os professores e servidores que contribuíram para a minha formação.

Por fim, agradeço à Polícia Militar de Alagoas pela disponibilização dos dados utilizados neste trabalho. Sua colaboração foi imprescindível para que a pesquisa fosse conduzida com base em informações reais, contribuindo para que os resultados possam efetivamente apoiar a segurança pública no estado.

RESUMO

A predição espaço-temporal de crimes violentos contra o patrimônio é um problema relevante para o planejamento de ações preventivas e para a alocação de recursos de segurança pública. Este trabalho apresenta uma replicação metodológica, no contexto das cidades de Maceió e Arapiraca, do pipeline de previsão criminal em microescala proposto por Albors Zumel, Tizzoni e Campedelli (2025), baseado em uma arquitetura ConvLSTM. A partir de registros georreferenciados de roubos violentos contra o patrimônio cedidos pela Polícia Militar de Alagoas, cobrindo o período de 2012 a 2022, o trabalho organiza os dados em *grids* regulares com células de área-alvo aproximada de 0,2 km^2, agrega as ocorrências em janelas de 12 h, binariza o sinal célula a célula, e treina uma ConvLSTM com tratamento explícito do desbalanceamento via `BCEWithLogitsLoss` ponderada. A avaliação é feita por varredura do limiar de classificação, reportando *precision*, *recall* e F1, tanto na variante tradicional (célula a célula) quanto com tolerância espacial (vizinhança de Chebyshev ≤ 1). Os resultados mostram que, em ambas as cidades, o modelo atinge desempenho na variante espacial significativamente superior à variante tradicional (F1 espacial de aproximadamente 0,24 em Maceió e 0,21 em Arapiraca, contra valores inferiores a 0,05 na variante tradicional), evidenciando que a rede aprende a localizar corretamente vizinhanças de risco mesmo quando erra a célula exata. O trabalho discute, ainda, as principais limitações da abordagem, em particular a esparsidade extrema dos dados, e propõe direções para desenvolvimentos futuros, incluindo a incorporação de canais sociodemográficos e de mobilidade humana e a comparação contra modelos *baseline*.

Palavras-chave: predição criminal; aprendizado profundo; ConvLSTM; segurança pública; Maceió; Arapiraca.

ABSTRACT

Spatiotemporal forecasting of violent property crimes is a relevant problem for the planning of preventive actions and for the allocation of public security resources. This work presents a methodological replication, in the context of the cities of Maceió and Arapiraca, of the fine-grained crime forecasting pipeline proposed by Albors Zumel, Tizzoni, and Campedelli (2025), based on a ConvLSTM architecture. From georeferenced records of violent property crimes provided by the Military Police of Alagoas, covering the period from 2012 to 2022, the work organizes the data into regular grids with cells of approximately 0.2 km^2, aggregates occurrences in 12-hour windows, binarizes the cell-level signal, and trains a ConvLSTM with explicit imbalance treatment via weighted `BCEWithLogitsLoss`. Evaluation is performed by sweeping the classification threshold, reporting *precision*, *recall*, and F1 both in the traditional variant (cell-by-cell) and with spatial tolerance (Chebyshev neighborhood ≤ 1). The results show that, in both cities, the model achieves performance in the spatial variant significantly higher than in the traditional variant (spatial F1 of approximately 0.24 in Maceió and 0.21 in Arapiraca, against values below 0.05 in the traditional variant), evidencing that the network learns to correctly locate risk neighborhoods even when missing the exact cell. The work also discusses the main limitations of the approach, particularly the extreme sparsity of the data, and proposes directions for future developments, including the incorporation of sociodemographic and human mobility channels and comparison against *baseline* models.

Keywords: crime forecasting; deep learning; ConvLSTM; public security; Maceió; Arapiraca.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Os crimes violentos contra o patrimônio (CVP), como roubos, constituem um fenômeno urbano complexo que afeta a sensação de segurança (Camacho Doyle; GERELL; ANDERSHED, 2022). Este fenômeno possui uma dinâmica espaço-temporal própria, na qual as ocorrências não se distribuem de forma aleatória, mas tendem a se concentrar de forma persistente em uma pequena parcela do território. Essa regularidade é chamada, na criminologia, de Lei da Concentração do Crime, proposta por David Weisburd (2015), que demonstra que cerca de 50% dos crimes de uma cidade costumam ocorrer em apenas 4% a 6% de seus segmentos de rua.

Nesse sentido, a análise e a predição do crime podem ser tratadas como um problema de previsão espaço-temporal. Onde o espaço urbano é representado por células de área fixa (grids) e o tempo é discretizado em janelas regulares (dias ou semanas) produzindo sequências de mapas de ocorrência. Entretanto, ao adotar a unidade célula–tempo, surge um desafio técnico significativo: a esparsidade de dados. Como a maioria das células não apresenta ocorrências na maior parte do tempo, os registros tornam-se dominados por zeros, caracterizando a predição, na prática, como uma tarefa de previsão de eventos raros sob alto desbalanceamento (WANG et al., 2019)

Diante desse cenário, este trabalho propõe a predição de crimes violentos contra o patrimônio em Arapiraca e Maceió por meio de uma arquitetura de aprendizado profundo do tipo ConvLSTM, adequada para capturar simultaneamente padrões espaciais (no grid) e dependências temporais (na sequência de mapas). Usando como base metodológica, estudo prévio realizado por Albors Zumel, Tizzoni e Campedelli (2025) onde foi empregado ConvLSTM para previsão de crime em múltiplas cidades dos Estados Unidos (incluindo Baltimore, Chicago, Los Angeles e Philadelphia) com dados agregados em grids, buscando indicar a viabilidade dessa abordagem para cenários urbanos distintos.

1.1 MOTIVAÇÃO

Crimes violentos contra o patrimônio afetam a sensação de segurança (Camacho Doyle; GERELL; ANDERSHED, 2022) e a atividade econômica, além de pressionarem recursos públicos destinados à prevenção e à resposta (FE; SANFELICE, 2022). Parte fundamental dessa prevenção esta relacionada a coleta e análise de dados relacionados a esses eventos. O Fórum Brasileiro de Segurança Pública divulga anualmente os dados de segurança a nível nacional, tendo o do ano de 2025 apontado transformações importantes nas dinâmicas criminais, com redução de diversos crimes ocorridos no espaço público e, em sentido oposto, aumento de registros ligados a fraudes e golpes (crimes patrimoniais associados ao ambiente virtual). Apesar dessa inversão, os crimes cometidos no espaço físico, especialmente os crimes violentos contra o patrimônio, continuam a

exercer impacto social e econômico em centros urbanos (FE; SANFELICE, 2022), fenômeno relevante também para contextos como Maceió (Fórum Brasileiro de Segurança Pública, 2025)

Nesse cenário, é relevante investigar alternativas tecnológicas que apoiem medidas preventivas, principalmente as capazes de estimar o risco futuro por área e por período, oferecendo subsídios para decisões mais precisas para alocação de recursos e direcionamento de ações. Considerando que a ocorrência criminal apresenta dependências no tempo e no espaço e pode se concentrar em hotspots (WEISBURD, 2015), abordagens de modelagem espaço-temporal tendem a ser mais adequadas do que análises puramente temporais ou apenas descritivas. Assim, avaliar o uso de ConvLSTM com representações em grids, com validação apropriada e discussão de limitações, pode contribuir para a literatura aplicada e para iniciativas locais de planejamento e prevenção.

1.2 OBJETIVO

Replicar metodologicamente, no contexto das cidades de Maceió e Arapiraca, o pipeline de predição espaço-temporal de crimes baseado em ConvLSTM proposto por Albors Zumel, Tizzoni e Campedelli (2025), a partir de dados históricos georreferenciados de roubos violentos contra o patrimônio (2012–2022) cedidos pela Polícia Militar de Alagoas, organizados em uma representação por grids de área-alvo $\sim$0,2 km^2, de modo a verificar a viabilidade da abordagem em um contexto urbano brasileiro distinto do estudo de referência e a caracterizar suas limitações em cenários de alta esparsidade.

1.2.1 Objetivos Específicos

- Organizar e preparar os dados históricos de ocorrências (padronização de datas, conversão de coordenadas, remoção de registros inválidos, filtragem para as oito naturezas de roubo violento contra o patrimônio e separação por cidade);
- Construir, para cada cidade, uma representação espacial em grade regular $N \times N$ com células de área-alvo $\sim$0,2 km^2 — seguindo a metodologia de Albors Zumel, Tizzoni e Campedelli (2025) —, gerando também a máscara de validade que delimita o polígono municipal a partir de dados do OpenStreetMap;
- Definir a granularidade temporal e a formulação do alvo, agregando as ocorrências em janelas de 12 horas e binarizando as contagens (presença/ausência de qualquer roubo violento contra o patrimônio na célula durante o período);
- Construir as sequências de entrada por meio de janela deslizante de 29 passos (14 dias de *lookback* mais 1 passo-alvo de 12 h) e extração de sub-grids 16×16 condicionados a apresentar sinal positivo no alvo, caracterizando o desbalanceamento resultante e justificando a escolha do alvo binário;

- Implementar a arquitetura ConvLSTM em PyTorch, com tratamento explícito do desbalanceamento via `BCEWithLogitsLoss` ponderada por `pos_weight`, e treinar o modelo de forma cronológica (90% treino / 10% validação, com *buffer* entre as partições) para evitar vazamento temporal;
- Avaliar o desempenho da ConvLSTM em Maceió e Arapiraca por meio de varredura do limiar de classificação, reportando *precision*, *recall* e F1 tanto na variante tradicional (célula a célula) quanto com tolerância espacial (vizinhança de Chebyshev ≤ 1), conforme proposto na literatura de referência;
- Discutir comparativamente os resultados das duas cidades e o seu posicionamento frente ao estudo de Albors Zumel, Tizzoni e Campedelli (2025), explicitando limitações metodológicas (ausência de *baselines* e de conjunto de teste, uso de canal único de crime e de uma única semente) e indicando direções para trabalhos futuros.

2 REVISÃO DE LITERATURA

2.1 PREDIÇÃO ESPACIAL DO CRIME

A predição espacial do crime parte do entendimento de que esses eventos se distribuem de forma não aleatória no espaço, mas tendem a se concentram em um número pequeno de lugares, formando hotspots. Esse padrão direciona parte da literatura em criminologia espacial e indica que a dinâmica criminal deve ser analisada em escalas espaciais finas, nas quais se tornam mais visíveis os padrões das ocorrências (WEISBURD, 2015). Nesse contexto, as abordagens tradicionais como a Kernel Density Estimatio (KDE), modelos de processo pontual e modelos autorregressivos espaço-temporais foram amplamente empregados para identificar áreas críticas e antecipar a evolução do crime. Embora essas estratégias tenham sido importantes para a análise espacial do fenômeno criminal, sua capacidade preditiva tende a ser limitada em cenários urbanos complexos(WANG et al., 2019; Albors Zumel; TIZZONI; CAMPEDELLI, 2025).

A partir desse conceito, a predição do crime passou a ser formulada como um problema espaço-temporal. Em geral, o espaço urbano é descrito como uma grade regular de células com área fixa, enquanto o tempo é dividido em janelas sucessivas, como horas, dias ou blocos temporais maiores. A partir disso, cada observação passa a representar a intensidade criminal ou a presença/ausência de crime em uma célula durante um determinado intervalo de tempo. Dessa forma, a entrada do modelo corresponde a uma sequência de mapas criminais passados, eventualmente combinados com variáveis contextuais, e a saída corresponde a um mapa futuro de risco ou ocorrência criminal. Essa estrutura permite explorar simultaneamente dependências temporais entre períodos consecutivos e dependências espaciais entre áreas vizinhas, sendo particularmente útil para ambientes urbanos em que o crime apresenta persistência local e difusão territorial (WANG et al., 2019; Albors Zumel; TIZZONI; CAMPEDELLI, 2025)

Entretanto, essa formulação espaço temporal traz um dos principais desafios: a esparsidade dos dados. Em escalas espaciais muito finas, a maior parte das células não registra crime na maior parte dos períodos observados, produzindo bases desbalanceadas. Em termos práticos, isso significa que o modelo precisa aprender os padrões relevantes a partir de uma quantidade pequena de eventos positivos, o que dificulta o aumento da precisão e torna frequente a presença de falsos positivos. Esse problema aparece de forma recorrente na literatura (WANG et al., 2019; ROTARU et al., 2022; Albors Zumel; TIZZONI; CAMPEDELLI, 2025) e ajuda a explicar por que, mesmo quando os modelos apresentam bom recall ou desempenho global superior aos métodos clássicos, ainda persistem limitações importantes para uso operacional direto (WANG et al., 2019; PAPACHRISTOS, 2022; Albors Zumel; TIZZONI; CAMPEDELLI, 2025).

Nesse cenário, métodos de deep learning passaram a ganhar destaque por sua capacidade de aprender relações espaço-temporais complexas diretamente dos dados. (WANG et al., 2019)

adaptaram arquiteturas profundas para previsão criminal quase em tempo real, mostrando desempenho superior a métodos de referência, como ARIMA, k-NN e médias históricas, ao combinar padrões históricos do crime com variáveis externas. Em outra direção, (ROTARU et al., 2022) avançaram da predição agregada em grades para a predição em nível de evento, argumentando que a modelagem em resolução mais fina permitiria captar interações espaciais e temporais relevantes entre ocorrências urbanas. Mais recentemente, (Albors Zumel; TIZZONI; CAMPEDELLI, 2025) reforçaram essa ideia ao investigar previsões em microescala com redes do tipo ConvLSTM, incorporando não apenas o histórico criminal, mas também a mobilidade humana e características sociodemográficas. Seus resultados indicam que a combinação dessas informações pode melhorar a capacidade preditiva, sobretudo em tarefas realizadas em grades finas e janelas temporais curtas.

Apesar desses avanços, a literatura também tem enfatizado limites metodológicos, sociais e éticos da predição criminal. Do ponto de vista técnico, a baixa precisão em contextos altamente esparsos impõe cautela na interpretação dos resultados, pois alertas incorretos podem induzir respostas desproporcionais ou ineficientes. Do ponto de vista normativo, (PAPACHRISTOS, 2022) destaca que modelos preditivos não são automaticamente benéficos: uma vez implantados, tendem a ser utilizados pelas polícias de formas que contrariam a intenção dos pesquisadores, sem que haja marco regulatório adequado para limitar esse uso. Em sentido complementar, (ROTARU et al., 2022) argumentam que modelos de previsão precisos podem expor distorções sistêmicas no policiamento, como a realocação de recursos de áreas mais pobres para áreas mais ricas sob pressão institucional, e assim funcionar como instrumentos de responsabilização do Estado. Assim, a predição espacial do crime deve ser compreendida como um problema simultaneamente analítico e social, exigindo modelos robustos, validação crítica e reflexão ética sobre seus usos possíveis.

2.2 APRENDIZADO DE MÁQUINA

O aprendizado de máquina (*machine learning*) é um campo da inteligência artificial que estuda algoritmos capazes de aprender padrões diretamente a partir de dados, em vez de depender de regras explicitamente programadas. Na programação tradicional, o desenvolvedor descreve manualmente o conjunto de instruções que transforma uma entrada na saída desejada. No aprendizado de máquina, ao contrário, fornece-se ao algoritmo um conjunto de exemplos e deixa-se que ele ajuste seus próprios parâmetros de modo a inferir a relação entre entradas e saídas (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa abordagem é especialmente vantajosa em problemas nos quais a regra que liga entrada e resposta é complexa, desconhecida ou difícil de formalizar, como ocorre na identificação de padrões espaço-temporais de criminalidade.

De forma geral, os métodos de aprendizado de máquina costumam ser organizados em três paradigmas. No aprendizado supervisionado, cada exemplo de treinamento é acompanhado de um rótulo que indica a resposta correta, e o objetivo é aprender uma função que associe novas

entradas a saídas adequadas. No aprendizado não supervisionado, não há rótulos, e o algoritmo busca estruturas latentes nos dados, como agrupamentos ou representações compactas. Já no aprendizado por reforço, um agente aprende por tentativa e erro a partir de recompensas obtidas ao interagir com um ambiente (GOODFELLOW; BENGIO; COURVILLE, 2016; LECUN; BENGIO; HINTON, 2015). O problema tratado neste trabalho enquadra-se no aprendizado supervisionado, uma vez que cada célula da grade espacial, em cada intervalo de tempo, possui um rótulo conhecido que indica a presença ou ausência de ocorrência criminal.

Dentro do aprendizado supervisionado, distinguem-se duas tarefas principais. Na regressão, a saída é uma grandeza contínua, como prever a quantidade de eventos em uma região. Na classificação, a saída é uma categoria discreta, como decidir entre duas ou mais classes possíveis. A previsão criminal adotada aqui é formulada como um problema de classificação binária célula-tempo, no qual o modelo deve decidir, para cada célula e instante futuro, entre as classes *ocorrência* e *ausência de ocorrência* (GOODFELLOW; BENGIO; COURVILLE, 2016).

O processo de aprendizado consiste em ajustar os parâmetros do modelo de modo a minimizar uma função de perda, que quantifica a discrepância entre as previsões geradas e os rótulos reais. Esse ajuste é tipicamente realizado por métodos iterativos baseados em gradiente, que atualizam os parâmetros na direção que reduz o erro. Para avaliar se o modelo de fato aprendeu padrões úteis, e não apenas memorizou os exemplos vistos, os dados são divididos em conjuntos distintos de treinamento, validação e teste. O conjunto de treinamento é usado para ajustar os parâmetros, o de validação para escolher hiperparâmetros e monitorar o aprendizado, e o de teste para estimar o desempenho final em dados nunca vistos (GOODFELLOW; BENGIO; COURVILLE, 2016). A capacidade de manter bom desempenho fora dos dados de treinamento é chamada de generalização e constitui o objetivo central do aprendizado de máquina.

Dois comportamentos opostos comprometem essa generalização. No subajuste (*underfitting*), o modelo é simples demais para capturar a estrutura dos dados e apresenta erro elevado mesmo no treinamento. No sobreajuste (*overfitting*), o modelo é flexível a ponto de se ajustar a ruídos e particularidades dos exemplos de treinamento, alcançando erro baixo no treinamento, mas desempenho ruim em dados novos. O equilíbrio entre esses extremos depende da capacidade do modelo e da quantidade de dados disponíveis, e é justamente para controlar o sobreajuste que se empregam técnicas de regularização, discutidas na Seção 2.7 (GOODFELLOW; BENGIO; COURVILLE, 2016; LECUN; BENGIO; HINTON, 2015).

Um aspecto que ajuda a entender a transição dos métodos clássicos para as redes neurais diz respeito à forma como as características (*features*) usadas pelo modelo são obtidas. Em abordagens tradicionais de aprendizado de máquina, o desempenho depende fortemente de uma etapa manual de engenharia de atributos, na qual especialistas definem quais variáveis extrair dos dados brutos para que o algoritmo consiga aprender. O aprendizado profundo (*deep learning*) propõe um caminho alternativo: por meio de múltiplas camadas de processamento, o próprio modelo aprende representações hierárquicas dos dados, combinando características simples em representações progressivamente mais abstratas, sem necessidade de defini-las manualmente

(LECUN; BENGIO; HINTON, 2015). Essa capacidade de aprendizado de representações é particularmente útil para dados de alta dimensionalidade e estrutura complexa, como imagens, sequências e mapas espaço-temporais, e é o que motiva o uso das redes neurais artificiais, apresentadas a seguir.

2.3 REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais (RNAs) são sistemas computacionais baseados em modelos matemáticos inspirados no sistema nervoso, buscando aproximar, de forma abstrata, mecanismos de processamento e aprendizagem associados ao cérebro humano (FURTADO, 2019). Um marco histórico ocorreu em 1943, quando Warren McCulloch e Walter Pitts propuseram um dos primeiros modelos formais de neurônio artificial através de um modelo matemático que estabeleceu as premissas fundamentais para futuras implementações computacionais (MCCULLOCH; PITTS, 1943). Essas redes operam por processamento paralelo e distribuído, no qual múltiplas unidades simples atuam simultaneamente e de forma interconectada.

Em termos de funcionamento, uma RNA organiza essas unidades em camadas. Cada neurônio recebe valores de entrada, realiza uma combinação ponderada por pesos sinápticos, adiciona um viés e aplica uma função de ativação não linear, gerando um sinal que é propagado para os neurônios da camada seguinte. Essa composição em camadas permite o aprendizado de representações hierárquicas, onde camadas iniciais tendem a capturar padrões mais simples, enquanto camadas posteriores combinam esses padrões para modelar relações mais complexas (GUPTA, 2018).

Além da estrutura em camadas, o desempenho das RNAs depende do processo de treinamento, no qual os pesos e vieses são ajustados para minimizar o erro entre as saídas previstas e os valores reais. Em geral, esse ajuste é realizado por métodos de otimização baseados em gradiente, utilizando o algoritmo de retropropagação do erro (backpropagation) para calcular, de forma eficiente, como cada parâmetro da rede contribui para a função de perda (RUMELHART; HINTON; WILLIAMS, 1986). Os diferentes arranjos dessas camadas, bem como a forma como os parâmetros são conectados e compartilhados, dão origem a arquiteturas adequadas a diferentes tipos de dados e tarefas. Redes feedforward processam a informação em um fluxo unidirecional, sendo empregadas em problemas gerais de classificação e regressão.

Quando os dados apresentam estrutura espacial, as Redes Neurais Convolucionais (CNNs) exploram a organização em grade por meio de operações de convolução e compartilhamento de pesos, tornando-se eficazes para identificar padrões locais e construir representações hierárquicas em imagens e mapas (LECUN; BENGIO; HINTON, 2015). Já nos cenários em que a informação é sequencial, as Redes Neurais Recorrentes (RNNs) incorporam um estado interno que é atualizado ao longo do tempo, permitindo a modelagem de dependências temporais entre observações. Contudo, RNNs tradicionais podem apresentar limitações para aprender dependências de longo prazo, motivando o desenvolvimento de arquiteturas com mecanismos de memória e portas

de controle, como as Long Short-Term Memory (LSTM) (LIPTON; BERKOWITZ; ELKAN, 2015).

Essas distinções tornam-se especialmente relevantes em problemas espaço-temporais, nos quais é necessário capturar simultaneamente padrões no espaço e dinâmicas de tempo. Nesse contexto, arquiteturas híbridas como a ConvLSTM combinam convoluções (para modelar as correlações espaciais em grids) com recorrência (para representar sequências de mapas), sendo, portanto, adequadas para tarefas de previsão em dados espaço-temporais (SHI et al., 2015).

2.4 LONG SHORT-TERM MEMORY (LSTM)

As redes neurais recorrentes (RNNs) foram propostas para modelar dados sequenciais, permitindo que informações observadas em instantes anteriores influenciem a saída produzida em instantes posteriores. Apesar dessa característica, o treinamento dessas redes por retropropagação no tempo (Backpropagation Through Time - BPTT) encontra dificuldades quando as dependências temporais são longas. Isso ocorre porque, ao ser propagado por muitos passos de tempo, o gradiente pode decair ou crescer exponencialmente, fenômenos conhecidos como vanishing gradient e exploding gradient. Em termos práticos, isso faz com que as RNNs tradicionais tenham dificuldade para aprender relações entre eventos muito distantes na sequência, já que o sinal de erro pode se tornar pequeno demais para promover atualizações úteis dos pesos, ou grande demais a ponto de tornar o treinamento instável (LIPTON; BERKOWITZ; ELKAN, 2015; HOCHREITER; SCHMIDHUBER, 1997).

Foi nesse contexto que Hochreiter e Schmidhuber (1997) propuseram a arquitetura Long Short-Term Memory (LSTM), com o objetivo de contornar as limitações das RNNs convencionais no aprendizado de dependências de longo prazo. Ou seja a ideia central da LSTM é substituir o neurônio recorrente simples por uma célula de memória, com capcidade de controlar de forma mais estável o fluxo de informação ao longo do tempo. Diferentemente de uma RNN tradicional, a LSTM possui um estado celular que funciona como a memória interna dessa célula. Esse estado é atualizado de maneira controlada por mecanismos de portas, permitindo decidir quais informações devem ser armazenadas, descartadas e quais devem ser disponibilizadas na saída. Essa estrutura ajuda a preservar o fluxo do erro durante a retropropagação, reduzindo o problema do gradiente desvanecente e favorecendo o aprendizado de dependências temporais mais longas (HOCHREITER; SCHMIDHUBER, 1997; LIPTON; BERKOWITZ; ELKAN, 2015).

De forma geral, a célula LSTM é composta por quatro elementos principais: a porta de esquecimento, a porta de entrada, a atualização do estado celular e a porta de saída. Considerando a entrada no instante t, representado por x_t, o estado oculto anterior h_t-1 e o estado celular anterior c_t -1 a dinâmica da LSTM pode ser descrita pelas seguintes equações:

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{2.1}$$

$$i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right) \tag{2.2}$$

$$g_t = \tanh\left(W_g[h_{t-1}, x_t] + b_g\right) \tag{2.3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.4}$$

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) \tag{2.5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.6}$$

Nessas expressões, σ representa a função sigmoide, tanh representa a tangente hiperbólica e $\odot$ denota o produto elemento a elemento. A porta de esquecimento f_t controla quanto da memória anterior c_{t-1} deve ser mantido. A porta de entrada i_t define quanto da nova informação candidata g_t será incorporada ao estado celular. Em seguida, o estado celular c_t é atualizado pela combinação entre a memória anterior preservada e a nova informação selecionada. Por fim, a porta de saída o_t controla qual parcela do conteúdo armazenado em c_t será exposta no estado oculto h_t, que é repassado para os próximos passos da sequência e também pode ser utilizado para gerar a saída da rede (LIPTON; BERKOWITZ; ELKAN, 2015).

Um aspecto particularmente importante da LSTM é que o estado celular atua como um caminho mais estável para a propagação da informação ao longo do tempo. No trabalho original, esse mecanismo é descrito a partir da ideia de constant error carousel, isto é, uma conexão recorrente interna que favorece a passagem do erro sem que ele desapareça rapidamente ao longo de muitos passos temporais (HOCHREITER; SCHMIDHUBER, 1997). Em termos intuitivos, isso significa que a rede consegue reter informações relevantes por intervalos maiores, ao mesmo tempo em que utiliza as portas para filtrar o que deve entrar, permanecer ou sair da memória. Dessa forma, a LSTM consegue equilibrar retenção e atualização da informação, algo que é muito mais difícil de alcançar em RNNs recorrentes simples.

Na formulação original de Hochreiter e Schmidhuber (1997), a arquitetura enfatizava principalmente a célula de memória e os mecanismos de controle de entrada e saída. Posteriormente, Gers et al. introduziram a forget gate, que passou a ser incorporada de forma quase padrão nas implementações modernas, pois permite que a rede aprenda também quando deve apagar o conteúdo armazenado no estado celular (LIPTON; BERKOWITZ; ELKAN, 2015). Além disso, foram propostas variantes com peephole connections, nas quais o estado celular passa a alimentar diretamente algumas portas, permitindo um controle ainda mais refinado em tarefas sensíveis a intervalos temporais precisos (LIPTON; BERKOWITZ; ELKAN, 2015).

Em síntese, a LSTM representa um avanço importante em relação às RNNs tradicionais porque foi projetada especificamente para lidar com dependências de longo prazo em dados

sequenciais. Sua arquitetura baseada em células de memória e portas de controle tornou possível um fluxo mais estável de informação e gradiente ao longo do tempo, o que explica seu amplo uso em tarefas envolvendo séries temporais, linguagem natural, reconhecimento de fala, vídeo e, mais recentemente, em arquiteturas derivadas como a ConvLSTM. Nesse sentido, compreender a LSTM é fundamental, pois a ConvLSTM preserva a mesma lógica de controle da memória temporal, substituindo apenas as transformações totalmente conectadas por operações convolucionais, de modo a incorporar explicitamente a estrutura espacial dos dados.

2.5 CONVLSTM

A ConvLSTM (Convolutional Long Short-Term Memory) é uma arquitetura derivada das redes LSTM, proposta por Shi et al. (2015), como já citado, para lidar com sequências espaço-temporais, ou seja, que possuem simultaneamente padrões espaciais e dinâmicas de tempo. Diferentemente da LSTM, em que as transformações entre entrada e estado são realizadas por operações totalmente conectadas, a ConvLSTM substitui essas operações por convoluções, preservando a estrutura espacial dos dados ao longo do tempo. Isso permite que o modelo capture tanto a evolução temporal quanto as correlações espaciais.

As convoluções consistem na aplicação de pequenos filtros na entrada, também chamados de kernels, que analisam partes pequenas da grade de dados. Esses filtros identificam a concentração, mudanças e semelhanças entre a região analisada e as regiões vizinhas. À medida que percorrem a entrada, as características capturadas são organizadas em diferentes posições do espaço, formando representações que destacam onde determinados padrões ocorrem (LECUN; BENGIO, 1998).

Na ConvLSTM, cada observação da sequência temporal é representada em uma estrutura organizada sobre uma grade espacial, preservando a posição relativa das informações ao longo do tempo. Diferentemente de abordagens que transformam cada instante em um vetor, essa arquitetura incorpora convoluções nas transições entre a entrada, o estado oculto e a memória. Com isso, a atualização de cada região passa a considerar não apenas o histórico temporal cumulativo, mas também a influência exercida pelas regiões vizinhas, favorecendo a modelagem de correlações espaço-temporais em dados estruturados em mapas ou grades (SHI et al., 2015, 2017).

No contexto da predição criminal, essa característica é especialmente relevante, pois a ocorrência de eventos em uma célula po de estar associada tanto ao comportamento observado em períodos anteriores quanto à dinâmica espacial das células vizinhas. Assim, a ConvLSTM oferece uma estrutura adequada para modelar, de forma integrada, dependências temporais e relações espaciais locais em problemas representados por grades espaciais (SHI et al., 2015).

2.6 EQUAÇÕES FORMAIS DA CONVLSTM

A ConvLSTM pode ser entendida como uma extensão da LSTM para dados espaço-temporais. Enquanto na LSTM tradicional as entradas, os estados ocultos e o estado celular são tratados como vetores, na ConvLSTM essas grandezas passam a ser representadas como tensores tridimensionais, preservando a organização espacial da informação. Em geral, X_t, H_t e C_t possuem a forma canais $\times$ altura $\times$ largura, de modo que as duas últimas dimensões correspondem à grade espacial observada. Com isso, a arquitetura passa a modelar, de forma conjunta, dependências temporais e correlações espaciais locais (SHI et al., 2015).

A principal diferença em relação à LSTM convencional é que as multiplicações matriciais das transições entrada-estado e estado-estado são substituídas por operações de convolução. Assim, o estado futuro de cada posição da grade não depende apenas do valor nessa posição, mas também das informações presentes em sua vizinhança local. Seguindo a formulação original proposta por Shi et al. (SHI et al., 2015), a dinâmica da ConvLSTM pode ser descrita pelas seguintes equações:

$$i_t = \sigma\left(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i\right) \tag{2.7}$$

$$f_t = \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f\right) \tag{2.8}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh\left(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c\right) \tag{2.9}$$

$$o_t = \sigma\left(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o\right) \tag{2.10}$$

$$H_t = o_t \odot \tanh(C_t) \tag{2.11}$$

Nessas expressões, $*$ denota a operação de convolução, $\odot$ representa o produto de Hadamard e σ é a função sigmoide. A porta de entrada i_t controla a incorporação de novas informações ao estado celular, a porta de esquecimento f_t determina quanto da memória anterior deve ser preservado, e a porta de saída o_t controla a parcela do conteúdo armazenado em C_t que será exposta no estado oculto H_t. Dessa forma, a ConvLSTM mantém a lógica de memória temporal da LSTM, porém substitui as operações totalmente conectadas por convoluções, o que a torna mais adequada para problemas nos quais a estrutura espacial da entrada precisa ser mantida ao longo do tempo (SHI et al., 2015).

Essa modificação é particularmente importante em tarefas de previsão espaço-temporal, pois permite explorar explicitamente a continuidade espacial entre regiões vizinhas. No trabalho original, os autores mostram que a ConvLSTM captura correlações espaço-temporais melhor do que a FC-LSTM em problemas de nowcasting de precipitação. Posteriormente, Shi et al. (SHI et

al., 2017) observam que, embora a ConvLSTM represente um avanço importante, sua estrutura convolucional recorrente ainda é invariável à localização, o que pode limitar a modelagem de movimentos mais complexos, como rotações e transformações que variam conforme a posição espacial.

Além disso, a formulação da ConvLSTM pode ser compreendida dentro de uma lógica mais ampla de modelos de sequência para sequência, em que uma rede codifica a sequência observada em uma representação latente e outra a utiliza para gerar a sequência futura. Essa ideia geral foi consolidada por Sutskever, Vinyals e Le (SUTSKEVER; VINYALS; LE, 2014) e passou a influenciar também arquiteturas voltadas à previsão de sequências visuais. Nesse contexto, trabalhos como o de Srivastava, Mansimov e Salakhutdinov (SRIVASTAVA; MANSIMOV; SALAKHUDINOV, 2015) ajudam a situar a ConvLSTM dentro do avanço mais amplo da modelagem temporal com redes recorrentes aplicadas a sequências de imagens e vídeo.

2.7 TÉCNICAS DE REGULARIZAÇÃO E OTIMIZAÇÃO

O treinamento de modelos profundos para dados espaço-temporais envolve desafios como sobreajuste, instabilidade na propagação dos gradientes e sensibilidade à escolha dos hiperparâmetros. Nesse contexto, técnicas de regularização e estratégias de otimização tornam-se fundamentais para favorecer a convergência do modelo, melhorar sua capacidade de generalização e tornar o processo de treinamento mais estável. Entre os principais recursos utilizados neste trabalho destacam-se o *Dropout*, a *Batch Normalization*, as funções de ativação, o otimizador Adam e o escalonamento da taxa de aprendizado por *Cosine Annealing*.

O *Dropout* é uma técnica de regularização proposta por Srivastava et al. (SRIVASTAVA et al., 2014), baseada no desligamento aleatório de neurônios durante o treinamento. Em cada iteração, uma fração das ativações é temporariamente anulada, o que impede a rede de depender excessivamente de um subconjunto específico de unidades. Como consequência, reduz-se a coadaptação entre neurônios e favorece-se a aprendizagem de representações mais robustas. Em termos práticos, o *Dropout* atua como uma forma eficiente de regularização, contribuindo para diminuir o sobreajuste e melhorar a capacidade de generalização do modelo em dados não vistos.

Outra técnica amplamente empregada é a *Batch Normalization*, proposta por Ioffe e Szegedy (IOFFE; SZEGEDY, 2015). Essa abordagem consiste em normalizar as ativações intermediárias da rede ao longo dos mini-lotes de treinamento, reduzindo a variação estatística entre camadas durante a aprendizagem. Com isso, o treinamento tende a se tornar mais estável, menos sensível à inicialização dos pesos e mais rápido em termos de convergência. Além de acelerar o processo de otimização, a *Batch Normalization* também pode exercer efeito regularizador, contribuindo indiretamente para melhorar o desempenho do modelo.

As funções de ativação também desempenham papel central no comportamento da rede, pois introduzem não linearidade e permitem que o modelo aprenda relações complexas entre entrada e saída. Entre as funções mais utilizadas estão a sigmoide, a tangente hiperbólica e a ReLU. A

função sigmoide é definida por

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.12}$$

e produz saídas no intervalo $[0,1]$, sendo particularmente útil em portas de controle, como ocorre em arquiteturas recorrentes do tipo LSTM e ConvLSTM, nas quais é necessário decidir quanto de informação deve ser mantido, esquecido ou liberado. Já a função tangente hiperbólica é dada por

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.13}$$

e gera valores no intervalo $[-1,1]$, o que a torna conveniente para representar estados internos centrados em zero. Em arquiteturas recorrentes, essa função é frequentemente usada para atualizar o estado celular e gerar representações intermediárias. Por sua vez, a função ReLU (*Rectified Linear Unit*) é definida como

$$\mathrm{ReLU}(x) = \max(0,x) \tag{2.14}$$

e se destaca por sua simplicidade computacional e por atenuar, em certa medida, problemas associados ao gradiente desvanecente, favorecendo o treinamento de redes profundas (NAIR; HINTON, 2010). Em geral, sigmoid e *tanh* aparecem com maior frequência nas portas e mecanismos internos de redes recorrentes, enquanto a ReLU é muito utilizada entre camadas convolucionais e blocos mais profundos da rede.

No processo de otimização, um dos algoritmos mais empregados é o Adam (*Adaptive Moment Estimation*), proposto por Kingma e Ba (KINGMA; BA, 2014). Esse método combina ideias do *Momentum* e do RMSProp, adaptando a taxa de aprendizado de cada parâmetro a partir de estimativas dos primeiros e segundos momentos dos gradientes. Dessa forma, o Adam tende a apresentar boa eficiência computacional, rápida convergência e robustez em problemas com grande número de parâmetros, sendo por isso amplamente adotado no treinamento de redes neurais profundas.

Além da escolha do otimizador, o ajuste da taxa de aprendizado ao longo do treinamento também é um fator importante. Uma estratégia bastante utilizada é o *Cosine Annealing*, introduzido por Loshchilov e Hutter (LOSHCHILOV; HUTTER, 2016). Nessa abordagem, a taxa de aprendizado decresce segundo uma função cosseno ao longo das épocas, permitindo passos maiores no início do treinamento e ajustes mais finos nas etapas finais. Esse mecanismo ajuda a melhorar a convergência e pode contribuir para que o modelo encontre soluções de melhor qualidade no espaço de parâmetros.

Em conjunto, essas técnicas cumprem papéis complementares. O *Dropout* e a *Batch Normalization* ajudam a reduzir o sobreajuste e estabilizar o treinamento; as funções de ativação determinam como a informação é transformada e propagada ao longo da rede; o Adam fornece

uma estratégia eficiente de atualização dos parâmetros; e o *Cosine Annealing* ajusta dinamicamente a taxa de aprendizado ao longo das épocas. Assim, a combinação desses elementos constitui parte importante do desempenho de modelos profundos aplicados à modelagem espaço-temporal.

2.8 MÉTRICAS DE AVALIAÇÃO PARA CLASSIFICAÇÃO BINÁRIA

Na avaliação de modelos de classificação binária, é comum utilizar a matriz de confusão como ponto de partida para a definição das principais métricas de desempenho. Nessa matriz, as previsões do modelo são comparadas com os rótulos reais, resultando em quatro quantidades fundamentais: verdadeiros positivos (*True Positives* – TP), falsos positivos (*False Positives* – FP), verdadeiros negativos (*True Negatives* – TN) e falsos negativos (*False Negatives* – FN). Os verdadeiros positivos correspondem aos exemplos positivos corretamente classificados como positivos, enquanto os falsos positivos representam exemplos negativos incorretamente classificados como positivos. De forma análoga, os verdadeiros negativos são os exemplos negativos corretamente classificados, e os falsos negativos correspondem aos exemplos positivos que o modelo deixou de identificar (MAXWELL; WARNER, 2020).

A partir dessas quantidades, podem ser definidas métricas amplamente utilizadas neste trabalho. A *precision* mede a proporção de predições positivas que de fato pertencem à classe positiva, sendo dada por

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.15}$$

Já o *recall*, também chamado de sensibilidade, mede a proporção de exemplos positivos corretamente recuperados pelo modelo:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2.16}$$

Enquanto a *precision* penaliza o excesso de alarmes falsos, o *recall* penaliza a perda de exemplos positivos. Como essas duas métricas podem entrar em conflito, é comum utilizar o *F1-score*, que corresponde à média harmônica entre *precision* e *recall*:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.17}$$

Essa métrica é particularmente útil quando se deseja um equilíbrio entre identificar corretamente a classe positiva e evitar classificações positivas incorretas (SOKOLOVA; LAPALME, 2009).

Para que um modelo produza uma decisão binária, é necessário definir um limiar de classificação (*threshold*). Em geral, a rede produz um escore contínuo, frequentemente interpretado como *logit*. Aplicando-se a função sigmoide, obtém-se uma probabilidade estimada para a

classe positiva. A previsão final é então obtida comparando essa probabilidade com um limiar τ. Assim, se $\hat{p} \geq \tau$, o exemplo é classificado como positivo; caso contrário, é classificado como negativo. Quando $\tau = 0.5$, tem-se a configuração mais usual, mas esse valor pode ser ajustado de acordo com o custo relativo entre falsos positivos e falsos negativos. Como consequência, *precision*, *recall* e F1-score dependem diretamente do limiar adotado (scikit-learn developers, 2025; PyTorch contributors, 2025).

Um aspecto especialmente importante em problemas de classificação binária é o desbalanceamento de classes. Esse cenário ocorre quando uma das classes está muito mais representada do que a outra no conjunto de dados. Nessa situação, métricas como acurácia podem se tornar enganosas, pois um classificador pode apresentar valor elevado simplesmente por favorecer a classe majoritária. Chawla et al. (2002) observam que, em bases desbalanceadas, a acurácia pode mascarar um desempenho ruim na classe minoritária, justamente a classe que muitas vezes possui maior interesse prático. De forma complementar, Saito e Rehmsmeier (SAITO; REHMSMEIER, 2015) mostram que, em cenários desbalanceados, medidas relacionadas à recuperação da classe positiva, como *precision* e *recall*, tendem a ser mais informativas do que avaliações baseadas apenas na separação global entre positivos e negativos.

Diferentes estratégias podem ser adotadas para mitigar o efeito do desbalanceamento. Uma delas consiste em ponderar a função de perda, atribuindo maior peso à classe minoritária. No caso da `BCEWithLogitsLoss`, a documentação oficial do PyTorch define o parâmetro `pos_weight` justamente para ajustar a contribuição relativa dos exemplos positivos na perda, o que é útil quando há desequilíbrio entre positivos e negativos. Além disso, essa função combina, em uma única operação, a sigmoide e a entropia cruzada binária, sendo mais estável numericamente do que aplicar primeiro a sigmoide e depois a perda BCE separadamente.

De forma simplificada, para um logit z e um alvo binário $y \in \{0,1\}$, a entropia cruzada binária pode ser expressa como

$$\mathscr{L}_{BCE} = -\left[y \log(\sigma(z)) + (1-y)\log(1-\sigma(z))\right] \tag{2.18}$$

Na prática, a `BCEWithLogitsLoss` implementa essa ideia diretamente sobre os logits, o que evita instabilidades numéricas em valores extremos. Quando se utiliza `pos_weight`, os exemplos positivos passam a ter maior influência no valor final da perda, favorecendo o aprendizado da classe minoritária.

Outra estratégia importante é a *Focal Loss*, proposta por Lin et al. (2017). Essa função foi desenvolvida para reduzir a influência de exemplos fáceis durante o treinamento e concentrar o aprendizado em exemplos difíceis ou raros. Em termos gerais, a *Focal Loss* adiciona um fator modulador à entropia cruzada, de modo que exemplos classificados com alta confiança contribuam menos para a perda. Isso é especialmente útil em cenários com forte desbalanceamento, nos quais a grande quantidade de exemplos negativos fáceis pode dominar o treinamento. Em sua forma binária, ela pode ser escrita como

$$FL(p_t) = -\alpha(1-p_t)^{\gamma}\log(p_t) \tag{2.19}$$

em que p_t representa a probabilidade associada à classe correta, α é um fator de balanceamento entre classes e γ controla o grau de focalização nos exemplos difíceis. Quando $\gamma = 0$, a *Focal Loss* se reduz à entropia cruzada tradicional. À medida que γ aumenta, exemplos fáceis recebem peso progressivamente menor no processo de otimização.

Além das abordagens baseadas em função de perda, também é possível atuar no próprio conjunto de treinamento por meio de técnicas de reamostragem. As duas estratégias mais comuns são o *oversampling*, que aumenta a representação da classe minoritária, e o *undersampling*, que reduz a quantidade de exemplos da classe majoritária. Chawla et al. (2002) discutem essas abordagens no contexto de aprendizado com dados desbalanceados e propõem o SMOTE como alternativa ao simples balanceamento por replicação. Mesmo quando não se utiliza SMOTE especificamente, o princípio geral permanece: alterar a distribuição amostral do treinamento pode ajudar o modelo a não ficar excessivamente enviesado para a classe majoritária.

Dessa forma, a escolha das métricas e das estratégias de treinamento em problemas binários não deve se limitar à acurácia global. Em contextos com desbalanceamento de classes, torna-se mais apropriado analisar *precision*, *recall* e F1-score, definir cuidadosamente o limiar de classificação e empregar mecanismos de mitigação como ponderação da perda, *Focal Loss* e reamostragem do conjunto de treinamento.

Por fim, em problemas espaciais formulados sobre uma grade regular, é útil dispor de uma noção de distância entre células que não dependa apenas da contiguidade ortogonal. A *distância de Chebyshev*, também conhecida como *chessboard distance*, entre duas células (i_1, j_1) e (i_2, j_2) de um *grid* é definida por

$$D\big((i_1, j_1), (i_2, j_2)\big) = \max(|i_1 - i_2|,\ |j_1 - j_2|)\,, \tag{2.20}$$

e captura o deslocamento máximo, horizontal, vertical ou diagonal, entre as duas células (DEZA; DEZA, 2009). Com essa definição, a vizinhança de raio 1 ao redor de uma célula corresponde às 8 células ortogonal e diagonalmente adjacentes (totalizando 9 posições, incluindo a própria célula). Essa noção é a base do critério de tolerância espacial empregado neste trabalho: ao permitir que uma previsão seja contabilizada como acerto caso a ocorrência real esteja em uma célula a até 1 passo de Chebyshev, mitiga-se o efeito de pequenas imprecisões de localização sobre as métricas de classificação, problema particularmente relevante em *grids* de alta resolução.

2.9 TRABALHOS RELACIONADOS

A literatura recente sobre predição espaço-temporal do crime tem avançado na direção de modelos capazes de integrar dependências espaciais, temporais e contextuais em escalas cada vez mais finas. No contexto deste trabalho, essa discussão é particularmente importante porque o

presente estudo se caracteriza como uma replicação metodológica em cenário distinto, tomando como referência um modelo recente de aprendizado profundo aplicado à previsão criminal em microescala. Assim, esta seção é organizada em três blocos: o primeiro discute em profundidade o estudo base da replicação; o segundo situa esse trabalho em uma família mais ampla de abordagens de aprendizado profundo para previsão de crime; e o terceiro examina a literatura brasileira e o contexto local, de modo a justificar a relevância da aplicação em Maceió e Arapiraca.

O estudo que constitui a principal referência desta pesquisa é o de Albors Zumel, Tizzoni e Campedelli (2025). Os autores propõem um modelo de aprendizado profundo para previsão criminal em escalas espaço-temporais finas, com o objetivo de avaliar em que medida a incorporação de variáveis de mobilidade humana e características sociodemográficas melhora o desempenho preditivo. O trabalho considera quatro cidades norte-americanas, Baltimore, Chicago, Los Angeles e Philadelphia, e organiza os dados criminais em grades regulares com células de 0,2 km². A tarefa preditiva é formulada como classificação binária célula-tempo, em que o modelo deve prever a ocorrência de crime 12 horas à frente a partir de janelas históricas de 14 dias ou 2 dias. Para isso, os autores utilizam uma arquitetura ConvLSTM e comparam seus resultados com três baselines: regressão logística, random forest e LSTM padrão. Os resultados mostram que a ConvLSTM supera os modelos alternativos em *recall* e F1-score nas quatro cidades analisadas, com precisão comparável ou ligeiramente superior às alternativas. Além disso, os autores observam que a inclusão de variáveis de mobilidade melhora o desempenho sobretudo quando são utilizadas janelas temporais mais curtas, enquanto a combinação entre mobilidade e sociodemografia produz os melhores resultados gerais. Ainda assim, o estudo também explicita limites importantes: o ganho não é uniforme entre tipos criminais, e a própria utilidade de modelos profundos em escalas finas depende fortemente da qualidade, cobertura e complementaridade das fontes de dados utilizadas.

O trabalho de Albors Zumel, Tizzoni e Campedelli (2025) não é um caso isolado, mas se insere em uma agenda mais ampla de uso de aprendizado profundo para previsão criminal. Wang et al. (2019), por exemplo, abordam a previsão criminal em tempo real com uma formulação ternária e adaptam uma arquitetura do tipo *spatial-temporal residual network* para prever a distribuição do crime em Los Angeles em escala horária e em parcelas do tamanho de bairros. Seus resultados indicam superioridade em relação a métodos anteriores e mostram que arquiteturas profundas podem capturar padrões espaço-temporais mais complexos do que abordagens estatísticas tradicionais. Em outra direção, Rotaru et al. (2022) propõem uma mudança importante de granularidade ao trabalhar no nível do evento, em vez de grades agregadas, e mostram que é possível aprender dependências espaço-temporais diretamente a partir dos registros individuais de ocorrências. O estudo reporta desempenho elevado em cidades norte-americanas e argumenta que esse tipo de modelagem permite não apenas prever eventos futuros, mas também investigar distorções sistêmicas no policiamento. Comentando esse trabalho, Papachristos (2022) ressalta que ferramentas de previsão criminal podem produzir ganhos analíticos relevantes, mas tam-

bém carregam riscos éticos e institucionais importantes, sobretudo porque as polícias tendem a utilizá-las de maneiras que contrariam a intenção dos pesquisadores, sem um marco regulatório que limite esse uso. Em conjunto, esses trabalhos mostram que o campo tem avançado tanto em precisão preditiva quanto em sofisticação metodológica, mas também reforçam que desempenho estatístico não elimina o problema dos vieses e das implicações normativas do uso desses sistemas.

Além dessas aplicações diretamente ligadas ao crime, a própria literatura de previsão espaço-temporal em aprendizado profundo ajuda a contextualizar o uso da ConvLSTM. Shi et al. (2015) mostram que a ConvLSTM preserva explicitamente a estrutura espacial dos dados ao substituir multiplicações matriciais por convoluções nas transições entrada-estado e estado-estado. Posteriormente, Shi et al. (2017) destacam que, embora a recorrência convolucional capture melhor correlações locais do que estruturas totalmente conectadas, sua natureza invariável à localização pode limitar a modelagem de dinâmicas mais complexas. Isso é relevante para a previsão criminal, uma vez que os padrões urbanos não são homogêneos e podem variar conforme a posição na cidade, a conectividade entre áreas e os fluxos populacionais. Em estudos mais recentes, outras arquiteturas também vêm sendo exploradas, como modelos baseados em grafos e atenção. No caso brasileiro, Hassan et al. (2024) aplicam Graph Neural Networks à cidade de São Paulo, integrando dados de crime, malha viária e informações urbanas, o que indica que a agenda metodológica nacional já começa a dialogar com abordagens contemporâneas de previsão espaço-temporal, ainda que por caminhos distintos da ConvLSTM.

No Brasil, entretanto, a literatura ainda parece mais consolidada em análises espaciais do crime e em modelos agregados baseados em indicadores urbanos do que em previsões finas com aprendizado profundo. Beato, Silva e Tavares (2008) já argumentava que a compreensão do crime em espaços urbanos exige atenção às características ecológicas da cidade e à distribuição espacial das ocorrências. Na mesma linha, Soares e Saboya (2019) propõem um modelo estruturador para discutir fatores espaciais associados à ocorrência criminal, reforçando a importância da dimensão urbana na explicação da violência.Alves, Ribeiro e Rodrigues (2018) avançam em direção à predição ao utilizar aprendizado de máquina, especificamente random forest, para prever homicídios a partir de métricas urbanas em cidades brasileiras, identificando desemprego e analfabetismo entre os fatores mais relevantes. Esses estudos mostram que há produção nacional importante sobre crime, espaço urbano e modelagem quantitativa, mas não configuram ainda um corpo robusto de trabalhos com resolução célula-tempo, janelas curtas e arquiteturas profundas recorrentes comparáveis ao desenho de Albors Zumel, Tizzoni e Campedelli (2025).

Essa lacuna se torna ainda mais relevante quando se considera o contexto de Alagoas. Fontes institucionais como o Anuário Brasileiro de Segurança Pública (Fórum Brasileiro de Segurança Pública, 2025) mostram que a violência letal e a transformação recente das dinâmicas patrimoniais permanecem como temas centrais no país. Ao mesmo tempo, a literatura localizada sobre Maceió e Arapiraca ainda é mais fortemente orientada para descrição espacial, epidemiologia da violência ou discussão urbana do que para replicações metodológicas em aprendizado pro-

fundo. Nesse sentido, aplicar uma abordagem inspirada em Albors Zumel, Tizzoni e Campedelli (2025) ao contexto de Maceió e Arapiraca não apenas permite testar a portabilidade de um modelo recente para uma realidade urbana distinta, mas também contribui para preencher uma lacuna empírica e metodológica na literatura brasileira. Assim, a principal contribuição deste trabalho está menos em propor uma arquitetura inédita e mais em avaliar, criticamente, o que ocorre quando uma estratégia recente de previsão criminal em microescala é transferida para um contexto urbano nordestino marcado por dinâmicas sociais, espaciais e criminais próprias.

3 METODOLOGIA

Esta seção detalha as etapas do pipeline experimental implementado neste trabalho, desde a obtenção dos dados brutos até a avaliação do modelo treinado. Todas as etapas foram desenvolvidas em Python, utilizando Jupyter Notebooks e bibliotecas como pandas, geopandas, osmnx, shapely, folium, PyTorch e scikit-learn.

3.1 DADOS DE ORIGEM

Os dados de ocorrências criminais foram obtidos da Polícia Militar de Alagoas por meio de parceria institucional, mediante assinatura de termo de confidencialidade. Em razão dessa restrição, a base de dados original não pode ser disponibilizada publicamente. Os dados correspondem a registros georreferenciados (latitude e longitude) de crimes violentos contra o patrimônio nas cidades de Maceió e Arapiraca, cobrindo o período de 2012 a 2022. O conjunto original contém 137.300 registros com 30 variáveis, incluindo natureza do fato, data/hora, coordenadas geográficas e cidade.

Embora os dados brutos e suas derivações que permitam reidentificar ocorrências (CSVs intermediários por célula, datasets pré-processados, predições e alvos do conjunto de validação, índices de sub-grids) não possam ser divulgados, todo o código-fonte utilizado para o pré-processamento, a construção dos grids, a preparação das sequências e o treinamento do modelo está disponível publicamente em repositório no GitHub[1], permitindo a reprodutibilidade metodológica do trabalho. Acompanham o código apenas artefatos agregados ou de geometria pública, a saber: o polígono municipal extraído do OpenStreetMap em formato GeoJSON, as curvas de *loss* por época e as tabelas de métricas por limiar de classificação.

3.2 PRÉ-PROCESSAMENTO DOS DADOS

O pré-processamento foi realizado em múltiplas etapas encadeadas:

3.2.1 Carregamento e limpeza inicial

Os arquivos CSV de entrada foram carregados e concatenados. As colunas de data foram convertidas para o formato datetime (formato `dd/mm/yyyy HH:MM:SS`) e as coordenadas de latitude e longitude, originalmente com separador decimal em vírgula, foram convertidas para tipo numérico. Registros com data ou coordenadas inválidas (valores ausentes) foram removidos.

[1] <https://github.com/leonardovff/crime-forecasting>

3.2.2 Filtragem por tipo de crime

Das 23 naturezas de fato distintas presentes nos dados, foram selecionadas 8 naturezas de roubo violento contra o patrimônio, escolhidas por concentrarem a maior parte das ocorrências e por compartilharem dinâmica espaço-temporal análoga (predominantemente vias públicas e estabelecimentos urbanos):

- Roubo a transeunte;
- Roubo a residência;
- Roubo de veículo (moto);
- Roubo de veículo (de passeio);
- Roubo de veículo (outros);
- Roubo a transporte coletivo rodoviário;
- Roubo a transporte coletivo urbano;
- Roubo a casa comercial.

Foram excluídas naturezas com descrição genérica (*roubo outros*, $\approx 11\%$ do total mas sem detalhamento que permita análise) ou com baixo volume e dinâmica espacial distinta dos demais roubos urbanos (*roubo a correspondente bancário* e *roubo a correios*, juntos $< 0{,}1\%$ do total). Após essa filtragem, restaram 72.724 registros em Maceió e 17.406 em Arapiraca, que foram separados em arquivos distintos por cidade. Como o objetivo é prever a presença de *qualquer* crime de roubo violento contra o patrimônio na célula do grid, e não distinguir por subcategoria, todas as ocorrências selecionadas foram tratadas como uma única classe binária a partir desta etapa.

3.2.3 Filtragem espacial com polígono municipal

Para cada cidade, o polígono do limite municipal foi obtido via OpenStreetMap (usando a biblioteca osmnx). Em seguida, realizou-se um *spatial join* entre os pontos de ocorrência e o polígono, mantendo apenas os registros cujas coordenadas caem efetivamente dentro do município. Para Maceió, dos 72.724 registros, 72.649 foram mantidos (75 descartados por estarem fora do polígono); para Arapiraca, dos 17.406 registros, 17.356 foram mantidos (50 descartados). Os resultados foram visualizados em mapas interativos (folium) para verificação.

3.3 CONSTRUÇÃO DA REPRESENTAÇÃO EM GRIDS

3.3.1 Geração da grade espacial

A grade espacial foi construída sobrepondo uma matriz regular $N \times N$ ao *bounding box* do polígono municipal obtido via OpenStreetMap. O valor de N não foi escolhido arbitrariamente: seguindo a metodologia de Albors Zumel, Tizzoni e Campedelli (2025), fixou-se a *área-alvo* de cada célula do grid em aproximadamente 0,2 km^2 (equivalente a 0,077 milhas quadradas, unidade espacial empregada no estudo de referência nas quatro cidades norte-americanas analisadas, Baltimore, Chicago, Los Angeles e Filadélfia). Essa escolha representa um compromisso entre granularidade espacial fina o suficiente para capturar padrões micro-geográficos de concentração criminal (*law of crime concentration*) e densidade de eventos suficiente para evitar esparsidade extrema que inviabilizaria o aprendizado.

Dado o alvo de área por célula, o número de divisões N é derivado automaticamente do tamanho do *bounding box* da cidade pela relação

$$N = \text{round}\left(\sqrt{\frac{A_{\text{bbox}}}{A_{\text{célula}}}}\right), \tag{3.1}$$

em que A_{bbox} é a área do *bounding box* do polígono municipal (em km^2), calculada convertendo graus para quilômetros com correção de latitude, e $A_{\text{célula}} = 0{,}2$ km^2. Essa formulação torna o procedimento reutilizável para qualquer cidade, bastando substituir o polígono de entrada, e reproduz, por construção, os valores de N reportados em Albors Zumel, Tizzoni e Campedelli (2025) para as cidades estudadas (por exemplo, $N = 39$ para Baltimore e $N = 85$ para Chicago).

Aplicando a Equação 3.1 aos polígonos municipais obtidos via OpenStreetMap, obtiveram-se os seguintes valores:

- **Maceió**: *bounding box* de aproximadamente $28{,}1 \times 37{,}8$ km (≈ 1.063 km^2), resultando em $N = 73$ e uma grade de 73×73 células (5.329 células no total);
- **Arapiraca**: *bounding box* de aproximadamente $23{,}4 \times 27{,}3$ km (≈ 638 km^2), resultando em $N = 57$ e uma grade de 57×57 células (3.249 células no total).

Cada célula possui área nominal de $\approx 0{,}2$ km^2, preservando a mesma resolução espacial do estudo de referência. Cada ocorrência criminal foi atribuída à célula correspondente por meio de verificação de contenção geométrica (*point-in-polygon*).

3.3.2 Máscara de validade

Para cada cidade, uma máscara binária de $N \times N$ foi gerada, indicando quais células do grid possuem interseção com o polígono municipal. Para Maceió, das 5.329 células totais, 2.714 ($\approx 50{,}9\%$) são válidas; para Arapiraca, das 3.249 células totais, 1.896 ($\approx 58{,}4\%$) são válidas.

Essa máscara é utilizada na etapa de avaliação para ignorar células que estão fora dos limites da cidade.

3.3.3 Agregação temporal

As ocorrências das naturezas selecionadas foram contabilizadas conjuntamente por célula e por hora do ano, formando uma matriz esparsa *célula* $\times$ *hora* para cada ano. Diferentemente de uma estratificação por subtipo, a agregação produz, desde esta etapa, um único indicador binário de ocorrência criminal por célula e período, alinhando o pipeline ao objetivo de prever a presença de *qualquer* roubo violento contra o patrimônio. Em seguida, essa matriz horária foi agregada em janelas de 12 horas (granularidade de 12h, totalizando 2 passos diários), resultando em aproximadamente 730 passos por ano. As contagens foram binarizadas (presença/ausência de crime: $\geq 1 \rightarrow 1$, caso contrário $\rightarrow 0$), produzindo um tensor final de dimensão $T \times N \times N \times 1$ (passos $\times$ linhas $\times$ colunas $\times$ canais) para cada cidade, com $T \approx 8.036$ passos de 12h abrangendo 2012–2022.

3.4 PREPARAÇÃO PARA TREINAMENTO

3.4.1 Janela deslizante e sub-grids

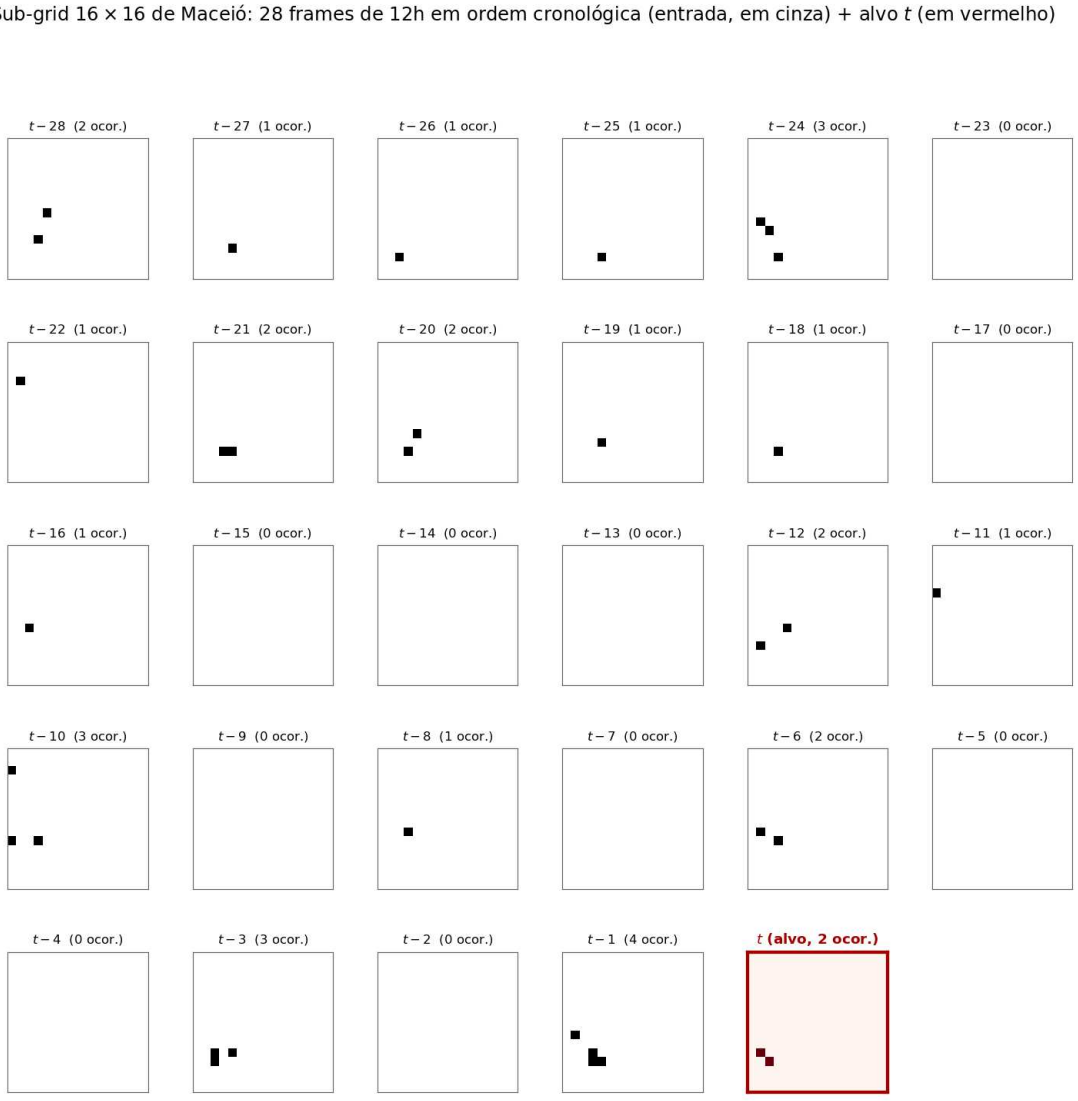
Sobre a sequência temporal de mapas binarizados de 12h, foi aplicada uma janela deslizante de 29 passos consecutivos. Os 28 primeiros passos (equivalentes a 14 dias de *lookback*, dado que cada passo cobre 12h) correspondem ao histórico utilizado como entrada do modelo, enquanto o 29º passo é o alvo a ser previsto, isto é, o mapa binário das próximas 12h. Esse tamanho de janela foi escolhido para alinhar o contexto temporal ao adotado por Albors Zumel, Tizzoni e Campedelli (2025), que reporta o melhor desempenho da ConvLSTM com *lookback* de 14 dias.

Para cada amostra temporal, em vez de utilizar o grid completo de $N \times N$, foram extraídos 5 sub-grids de 16×16 em posições aleatórias dentro da grade, condicionados a apresentar pelo menos uma ocorrência criminal no passo-alvo. A geração de cada sub-grid sorteia coordenadas (i, j) de modo que a janela 16×16 permaneça dentro da grade completa; caso, após 50 tentativas, não seja possível encontrar uma janela com sinal positivo no alvo, a amostra temporal é descartada. Essa estratégia amplia o volume de amostras de treinamento, força o modelo a observar regiões com sinal positivo, reduz o custo computacional do treino em relação ao grid completo e permite que a ConvLSTM opere sobre regiões locais da cidade, sem depender da geografia global.

A Figura 1 concretiza essa descrição com uma amostra real do conjunto de treino de Maceió. Cada um dos 29 painéis representa o mesmo sub-grid 16×16 em um passo de 12h: os 28 painéis em cinza são os frames de entrada do modelo, dispostos em ordem cronológica do passo mais antigo ($t - 28$, equivalente a 14 dias antes do alvo) até o mais recente ($t - 1$, equivalente a 12h antes do alvo); o painel em vermelho é o alvo t, ou seja, o mapa binário de ocorrências nas próximas 12h que o modelo deve prever. Duas características da tarefa ficam evidentes ao olhar

a figura. Primeiro, a esparsidade do problema: a maior parte dos passos da entrada não registra nenhuma ocorrência (painéis totalmente brancos) e, mesmo entre os passos com sinal, raramente mais do que duas ou três células do sub-grid acendem simultaneamente. Segundo, a relação entrada-alvo: a previsão exige que o modelo associe esse histórico esparso e ruidoso a um mapa binário também esparso, sem garantia de que o alvo apareça em uma posição já ativada no histórico recente.

Figura 1 – Exemplo de uma amostra do conjunto de treino de Maceió. Os 28 painéis em cinza correspondem aos frames de entrada (cada um cobrindo 12h, do passo $t-28$ ao $t-1$); o painel em vermelho é o alvo t, isto é, o mapa binário de ocorrências nas próximas 12h que o modelo deve prever. Cada célula exibe o valor binário (preto = ocorrência, branco = ausência); o número entre parênteses no título de cada painel indica a quantidade de células positivas naquele passo.



Fonte: Elaborado pelo autor (2026).

As escolhas dos três principais hiperparâmetros desta etapa (tamanho do *lookback*, dimensão do sub-grid e quantidade de sub-grids por amostra) têm justificativas distintas, descritas a seguir.

3.4.1.1 *Lookback* de 14 dias.

O *lookback* de $T = 28$ passos de 12h (equivalente a 14 dias) segue diretamente Albors Zumel, Tizzoni e Campedelli (2025), que avaliam dois valores de T (28 e 4, isto é, 14 e 2 dias) com o objetivo de identificar a quantidade ótima de histórico para a tarefa de previsão. Os autores reportam que sequências de entrada mais longas melhoram a previsão de crimes violentos, enquanto sequências mais curtas tendem a ser mais adequadas para crimes contra o patrimônio em geral. Como o presente trabalho concentra-se especificamente em *roubos violentos contra o patrimônio*, categoria que combina componente violento e patrimonial, optou-se pelo *lookback* mais longo (14 dias) como ponto de partida da replicação metodológica, alinhando-se ao melhor cenário observado no estudo de referência para a vertente violenta do fenômeno.

3.4.1.2 Tamanho do sub-grid $M \times M = 16 \times 16$.

O valor $M = 16$ é igualmente herdado de Albors Zumel, Tizzoni e Campedelli (2025), que justificam essa escolha por dois motivos: padronizar o tamanho da entrada da rede entre cidades de áreas distintas (no estudo original, N varia entre 39 em Baltimore e 85 em Chicago) e aumentar artificialmente o número de amostras de treinamento. Cada sub-grid de 16×16 corresponde, dada a área-alvo de 0,2 km^2 por célula, a aproximadamente 50 km^2 ($\sim$ 19,31 milhas quadradas no estudo original), escala compatível com uma região urbana operacionalmente coerente. No contexto desta replicação, o uso do sub-grid também atende a uma restrição prática: treinar a ConvLSTM sobre os grids completos de 73×73 (Maceió) ou 57×57 (Arapiraca) implicaria um custo computacional de memória e de tempo de treinamento incompatível com o *hardware* disponível (uma única GPU NVIDIA RTX 4070 *Laptop*). Manter $M = 16$ preserva, portanto, a comparabilidade com o estudo de referência *e* viabiliza a execução experimental neste trabalho.

3.4.1.3 Cinco sub-grids por amostra temporal.

A extração de 5 sub-grids aleatórios por amostra também é diretamente reaproveitada de Albors Zumel, Tizzoni e Campedelli (2025). A estratégia tem três efeitos complementares: aumentar em até $5\times$ o volume efetivo de amostras de treinamento, expor a rede a múltiplas regiões da cidade dentro de uma mesma amostra temporal e reduzir o impacto do sub-amostramento espacial sobre a representatividade estatística do treino.

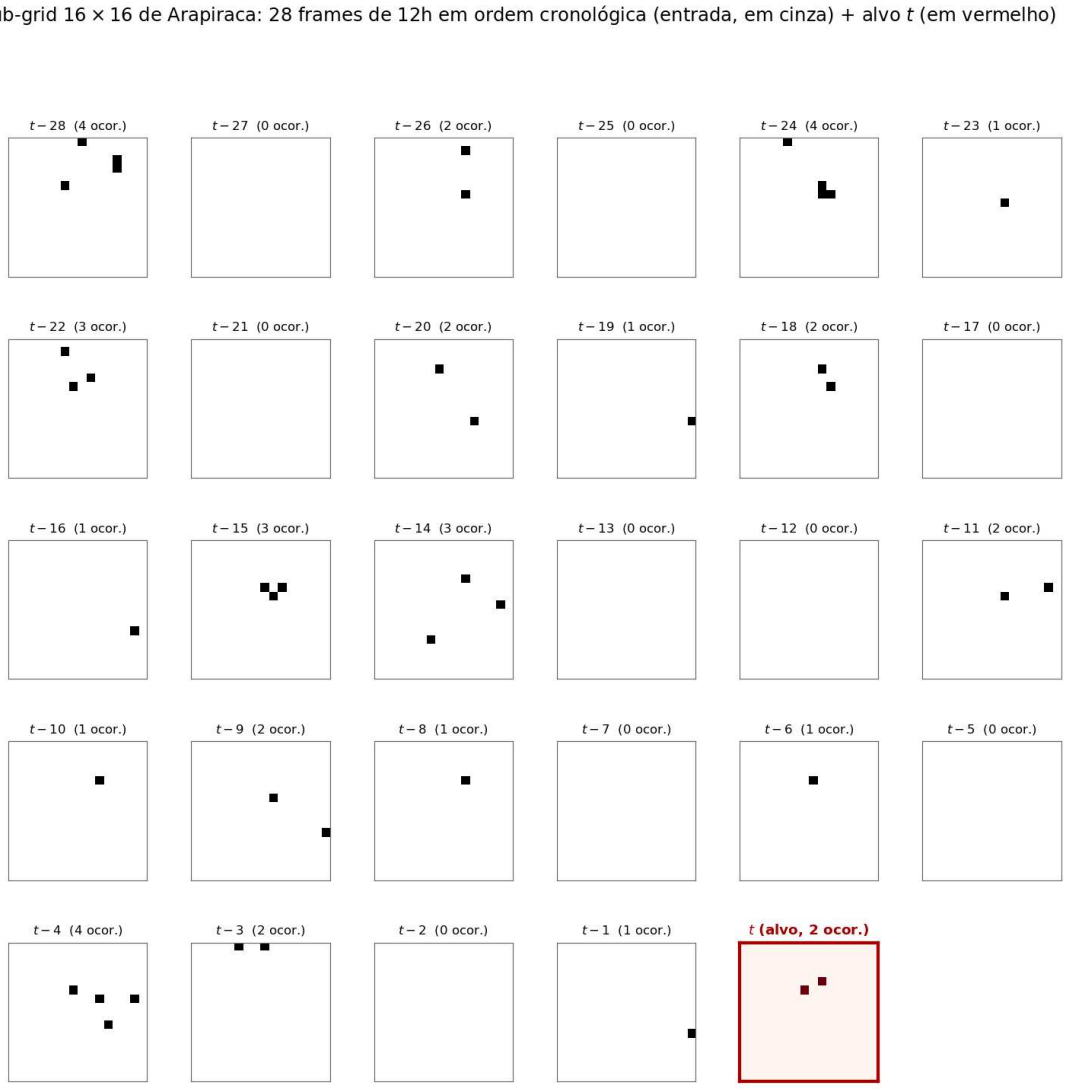
3.4.1.4 Critério de mínimo de ocorrências por sub-grid.

Cada sub-grid sorteado é mantido apenas se o passo-alvo apresentar pelo menos uma ocorrência criminal. Albors Zumel, Tizzoni e Campedelli (2025) adotam um critério mais rigoroso (≥ 2 ocorrências por sub-grid) com vistas a equilibrar o conjunto de treinamento. Neste trabalho, optou-se por relaxar esse limiar para ≥ 1 ocorrência, pois Maceió e, sobretudo, Arapiraca apresentam volume absoluto de ocorrências e densidade de eventos por período substancialmente

inferiores aos das quatro cidades norte-americanas estudadas no artigo de referência; manter o critério ≥ 2 resultaria em descarte excessivo de amostras temporais e em redução do conjunto de treino disponível para a rede. Essa é uma divergência metodológica deliberada em relação ao trabalho-base e deve ser considerada na interpretação comparativa dos resultados.

A Figura 2 torna concreta essa diferença de densidade: comparada à amostra equivalente de Maceió (Figura 1), a sequência de Arapiraca apresenta menos passos com sinal positivo no histórico e um alvo com menor número de células ativas. É justamente esse tipo de cenário que justifica o relaxamento do critério para ≥ 1 ocorrência: exigir ≥ 2 ocorrências no alvo eliminaria uma fração considerável das amostras temporais de Arapiraca antes mesmo do treinamento começar.

Figura 2 – Exemplo de uma amostra do conjunto de treino de Arapiraca. A leitura segue o mesmo padrão da Figura 1: 28 frames de entrada em cinza ($t-28$ a $t-1$) e o alvo t em vermelho. A densidade média de ocorrências por passo é visivelmente menor do que em Maceió, refletindo a maior esparsidade dos dados de Arapiraca, fato que motiva o relaxamento do critério mínimo de ocorrências por sub-grid descrito acima.



Fonte: Elaborado pelo autor (2026).

3.4.2 Divisão cronológica treino/validação

A divisão dos dados respeita rigorosamente a ordem temporal para evitar vazamento de informação. Os primeiros 90% da sequência são utilizados para treinamento e os 10% finais para validação, com um *buffer* de $n_{\text{frames}} - 1 = 28$ passos descartado entre treino e validação para impedir que uma mesma janela apareça parcialmente nas duas partições. Para Maceió, isso resultou em 32.556 amostras de treinamento e 3.238 de validação; para Arapiraca, em 27.590 amostras de treinamento e 2.484 de validação. Cada amostra tem formato $28 \times 16 \times 16 \times 1$ para a entrada e $16 \times 16 \times 1$ para o alvo. A normalização *min-max* foi calibrada apenas com os valores do conjunto de treino e em seguida aplicada também à validação, evitando vazamento estatístico. As células fora do polígono municipal foram zeradas via máscara antes do salvamento, e os arquivos resultantes (`maceio_chrono.npz` e `arapiraca_chrono.npz`) também armazenam os índices (i, j) dos sub-grids extraídos, para permitir avaliação posterior por região.

É importante explicitar o papel duplo desempenhado pela partição de validação neste trabalho. Embora os 10% finais não tenham sido utilizados para ajustar os pesos da rede durante o treinamento, eles cumprem dois papéis simultaneamente: (i) servem como conjunto de *validação*, sobre o qual é feita a varredura dos 51 limiares de classificação para identificar o τ ótimo (decisão de hiperparâmetro); e (ii) servem também como conjunto de reporte das métricas finais (*precision*, *recall*, *F1*, e suas variantes espaciais). Como o limiar é, por construção, escolhido para maximizar o F1 sobre exatamente o mesmo conjunto que será reportado, há um viés otimista nos valores apresentados: o "melhor F1 sobre 51 limiares" tende a ser superior ao F1 esperado em dados verdadeiramente inéditos. Esse viés é típico de pipelines em que validação e teste são confundidos em uma única partição.

Vale destacar que a ausência de conjunto de teste independente foi adotada para preservar a comparabilidade com o estudo de referência: Albors Zumel, Tizzoni e Campedelli (2025) também utilizam um *split* cronológico $90/10$ entre treino e teste, sem partição adicional. Contudo, o estudo de referência adota limiar fixo $\tau = 0{,}5$, sem varredura de seleção, de modo que o viés otimista descrito acima é mais acentuado neste trabalho do que no estudo-base. A introdução de uma terceira partição no presente trabalho deslocaria o pipeline em relação ao baseline de comparação. Ainda assim, a substituição do esquema atual por um *split* $80/10/10$ (treino/validação/teste) com seleção de limiar exclusivamente na validação e reporte sobre o teste fica indicada como caminho metodologicamente mais robusto, e está elencada entre os trabalhos futuros (Capítulo 5).

Quanto à cobertura temporal das partições, o conjunto de treino abrange aproximadamente os primeiros 9 anos e 10 meses do período observado (de janeiro de 2012 até o entorno de outubro de 2021), enquanto a validação cobre cerca de 14 meses (do entorno de outubro de 2021 até dezembro de 2022), considerando $T \approx 8.036$ passos de 12h, *buffer* de 28 passos entre as partições e $\textit{train\_size} = \lfloor 0{,}9 \cdot \textit{num\_samples} \rfloor$.

3.5 ARQUITETURA DO MODELO E TREINAMENTO

3.5.1 Arquitetura ConvLSTM implementada

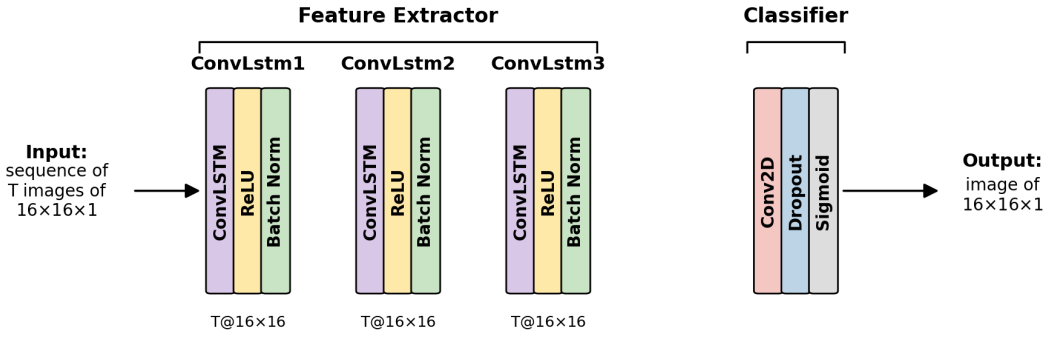
O modelo foi implementado em PyTorch e consiste em três componentes principais, conforme ilustrado na Figura 3:

1. **ConvLSTM empilhada**: 3 camadas ConvLSTM com 28 canais ocultos por camada ($\texttt{hidden\_chn} = \texttt{seq\_length} - 1 = 28$) e *kernels* de 3×3. Cada camada ConvLSTM é seguida por uma ativação ReLU e por uma normalização por lote (*BatchNorm3d*).

2. **Dropout**: camada de *dropout* com taxa $p = 0{,}8$ aplicada sobre o estado oculto da última camada.

3. **Convolução final**: uma convolução 3×3 com 1 canal de saída (*padding* "*same*"), produzindo o mapa de previsão para o passo seguinte.

Figura 3 – Arquitetura ConvLSTM empregada neste trabalho, adaptada de Albors Zumel, Tizzoni e Campedelli (2025). A entrada é uma sequência de T mapas 16×16 ($T = 28$ neste trabalho) processada por três blocos ConvLSTM (cada um com ConvLSTM + ReLU + *Batch Norm*); o classificador final é composto por uma convolução 2D, uma camada de *dropout* e uma sigmoide, produzindo um mapa 16×16 de probabilidades.



Fonte: Adaptado de Albors Zumel, Tizzoni e Campedelli (2025).

A entrada do modelo tem formato $(B, 28, 16, 16, 1)$ (*batch*, sequência temporal, altura, largura, canais) e a saída tem formato $(B, 16, 16)$, correspondente às probabilidades de ocorrência criminal em cada célula no próximo passo de 12h. Considerando todos os pesos das três camadas ConvLSTM, das três normalizações por lote e da convolução final, o modelo totaliza aproximadamente 143 mil parâmetros treináveis.

3.5.1.1 Justificativa dos hiperparâmetros

Todas as escolhas estruturais da arquitetura são herdadas, sem ajuste local, do código e do artigo de Albors Zumel, Tizzoni e Campedelli (2025), com o objetivo de preservar a comparabilidade

direta entre este trabalho e o estudo-base. Os autores reportam ter partido de uma arquitetura básica de *video frame prediction* com ConvLSTM e ajustado os hiperparâmetros até chegarem à configuração final descrita na Figura 3 (ver Apêndice G do artigo original para os valores específicos). Cada decisão é discutida a seguir.

Profundidade: 3 *camadas ConvLSTM.* A escolha de três camadas empilhadas reproduz a arquitetura final apresentada na Figura 4 de Albors Zumel, Tizzoni e Campedelli (2025). Em arquiteturas recorrentes convolucionais, o empilhamento permite que camadas iniciais capturem padrões locais (correlações imediatas entre célula e vizinhança próxima ao longo do tempo) e que camadas posteriores combinem essas representações em padrões espaço-temporais mais complexos, à semelhança da hierarquia observada em CNNs convencionais (LECUN; BENGIO; HINTON, 2015). A camada $ConvLSTM_1$ recebe como entrada o mapa de crimes original; a $ConvLSTM_2$ e a $ConvLSTM_3$ operam sobre as representações latentes produzidas pelas camadas anteriores, conforme indicado na Figura 3.

Canais ocultos: $\texttt{hidden\_chn} = \texttt{seq\_length} - 1 = 28$. O número de canais ocultos por camada não é um valor arbitrário, mas sim uma parametrização funcional adotada por Albors Zumel, Tizzoni e Campedelli (2025): define-se $\texttt{hidden\_chn} = T$, em que T é o número de passos de *lookback* ($T = 28$ aqui, equivalente a 14 dias). Essa convenção, presente no código original do paper de referência, faz a capacidade representacional da camada acompanhar a profundidade temporal da entrada.

Kernel convolucional 3×3. O *kernel* 3×3 é o tamanho canônico utilizado em Albors Zumel, Tizzoni e Campedelli (2025) para todas as transformações convolucionais (entrada-estado e estado-estado em cada *ConvLSTMCell*, e também na convolução final do classificador). *Kernels* dessa dimensão aparecem amplamente em arquiteturas recorrentes convolucionais desde Shi et al. (2015) e oferecem um bom compromisso entre capacidade de modelar dependências espaciais locais (cada célula é afetada por seus 8 vizinhos imediatos) e custo computacional. Combinados com *padding* "*same*", preservam as dimensões espaciais ao longo das camadas.

Dropout $p = 0{,}8$. O valor de *dropout* é especialmente alto (compare-se com $p \in [0{,}2; 0{,}5]$ comum em outras arquiteturas) e é igualmente herdado do código original de Albors Zumel, Tizzoni e Campedelli (2025), em que essa camada é aplicada entre a saída da última ConvLSTM e a convolução final. O valor elevado é coerente com o cenário do problema: o conjunto de treinamento é dominado por células negativas (esparsidade superior a 99%), e a ponderação por `pos_weight` amplifica a contribuição dos poucos exemplos positivos. Essa combinação aumenta o risco de o modelo memorizar configurações específicas dessas amostras, e um *dropout* forte funciona como regularização agressiva, forçando a rede a aprender representações mais robustas e menos dependentes de unidades específicas (SRIVASTAVA et al., 2014). O comportamento

empírico observado no treinamento de Arapiraca (*loss* de validação consistentemente abaixo da *loss* de treino, sem *overfitting*, conforme discutido no Capítulo 4) é compatível com a hipótese de que o *dropout* elevado, combinado às demais estratégias de regularização, atinge o efeito desejado.

Sobre o ajuste fino. Não foi conduzido neste trabalho nenhum procedimento de busca em grade (*grid search*) ou otimização sistemática dos hiperparâmetros estruturais. A justificativa é dupla. Primeiro, o objetivo do trabalho é replicar o pipeline de Albors Zumel, Tizzoni e Campedelli (2025) em um contexto urbano distinto, e introduzir variações arquiteturais comprometeria a comparabilidade dos resultados com o estudo-base. Segundo, dada a configuração de *hardware* disponível (uma única GPU NVIDIA RTX 4070 *Laptop* com 8 GB de VRAM), uma exploração ampla do espaço de hiperparâmetros é inviável dentro do escopo do TCC. A varredura de *hardware* e hiperparâmetros, incluindo *batch size*, profundidade, *kernel* e *dropout*, fica indicada como trabalho futuro (Capítulo 5).

3.5.2 Função de perda e otimização

Dada a severa esparsidade dos dados, foi utilizada a função de perda `BCEWithLogitsLoss` com `pos_weight` calculado como a razão entre o número de amostras negativas e positivas no conjunto de treinamento ($\approx 123{,}4$ para Maceió, com $\sim$0,80% de células positivas; $\approx 163{,}3$ para Arapiraca, com $\sim$0,61% de células positivas, ambos medidos sobre os sub-grids 16×16 de treino). Essa ponderação atribui maior importância aos casos positivos durante o cálculo do gradiente.

O otimizador utilizado foi o Adam com taxa de aprendizado de 10^{-5}, e um escalonador do tipo Cosine Annealing com $T_{\max} = 20$ épocas. O treinamento foi conduzido por 200 épocas com *batch size* de 252, utilizando GPU (NVIDIA GeForce RTX 4070 *Laptop* GPU) e semente aleatória fixa (42) para reprodutibilidade. O valor de *batch size* adotado é maior do que o reportado por Albors Zumel, Tizzoni e Campedelli (2025) (que usa 55); essa divergência foi adotada para melhor utilizar a memória disponível na GPU única e reduzir o tempo total de treinamento, sem alterar a dinâmica de convergência observada empiricamente.

3.5.3 Avaliação com tolerância espacial

Além das métricas tradicionais (precision, recall, F1), foi implementada uma avaliação com tolerância espacial, conforme proposto por Albors Zumel, Tizzoni e Campedelli (2025). Nessa variante, um falso positivo cuja célula vizinha (distância de Chebyshev ≤ 1) contenha um crime real é reclassificado como acerto espacial ("true positive espacial"). Isso resulta em métricas *precision_spatial*, *recall_spatial* e *f1_spatial*, que reconhecem previsões geograficamente próximas ao evento real.

A avaliação foi realizada sobre o conjunto de validação, varrendo limiares de classificação

de 0,50 a 1,00 (incrementos de 0,01). A máscara de validade municipal foi aplicada para que apenas as células dentro dos limites da cidade contribuíssem para as métricas.

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados obtidos com o modelo ConvLSTM treinado para a predição de crimes violentos contra o patrimônio nas cidades de Maceió e Arapiraca, conforme o pipeline descrito no Capítulo 3. Os experimentos foram conduzidos com semente aleatória fixada em $\texttt{seed} = 42$ para garantir reprodutibilidade.

4.1 CONFIGURAÇÃO EXPERIMENTAL

A Tabela 1 resume a configuração experimental para as duas cidades. O modelo ConvLSTM foi implementado em PyTorch e organizado em 3 camadas empilhadas com núcleo convolucional 3×3 e 28 canais ocultos por camada (correspondendo a $\texttt{seq\_length} - 1$), seguidos de uma camada *dropout* com $p = 0{,}8$ e de uma convolução final 3×3 que produz o mapa de probabilidades de ocorrência. O treinamento foi configurado para 200 épocas com otimizador Adam (taxa de aprendizado 1×10^{-5}), *batch size* de 252 e escalonamento `CosineAnnealingLR` com $T_{\max} = 20$. As entradas têm formato $(B, 28, 16, 16, 1)$ e correspondem a 14 dias de *lookback* em granularidade de 12h.

Tabela 1 – Configuração experimental por cidade ($\texttt{seed} = 42$).

Parâmetro	**Maceió**	**Arapiraca**
Grid ($N \times N$)	73×73	57×57
Células válidas	2.714 (50,9%)	1.896 (58,4%)
Ocorrências (após filtragem)	72.649	17.356
Amostras de treinamento (sub-grids)	32.556	27.590
Amostras de validação (sub-grids)	3.238	2.484
Esparsidade do conjunto de treino	$\sim$0,80%	$\sim$0,61%
`pos_weight`	$\approx 123{,}4$	$\approx 163{,}3$

Fonte: Elaborado pelo autor (2026).

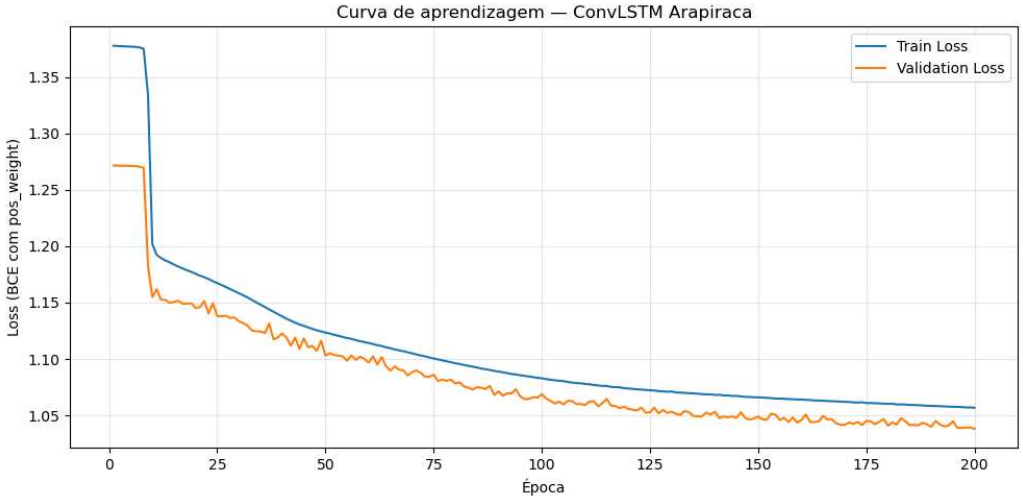
Os valores de `pos_weight` foram calculados, durante o treinamento, como a razão entre o número de células negativas e o número de células positivas no conjunto de treino de sub-grids 16×16 de cada cidade. A maior esparsidade observada em Arapiraca, em relação a Maceió, decorre tanto do menor volume absoluto de ocorrências disponíveis quanto da maior área proporcional do município ocupada por regiões de baixa densidade criminal.

4.2 CURVAS DE TREINAMENTO

4.2.1 Arapiraca

A Figura 4 apresenta a evolução das perdas de treino e validação (`BCEWithLogitsLoss` ponderada por `pos_weight`) ao longo das 200 épocas de treinamento da ConvLSTM em Arapiraca, com `seed` $= 42$. A perda de treino parte de 1,3779 na primeira época e atinge 1,0567 na época final; a perda de validação parte de 1,2717 e termina em 1,0381, valor que coincide com o mínimo observado em todo o treinamento.

Figura 4 – Curva de aprendizagem da ConvLSTM em Arapiraca: perda de treino e perda de validação por época (`seed` $= 42$, 200 épocas).



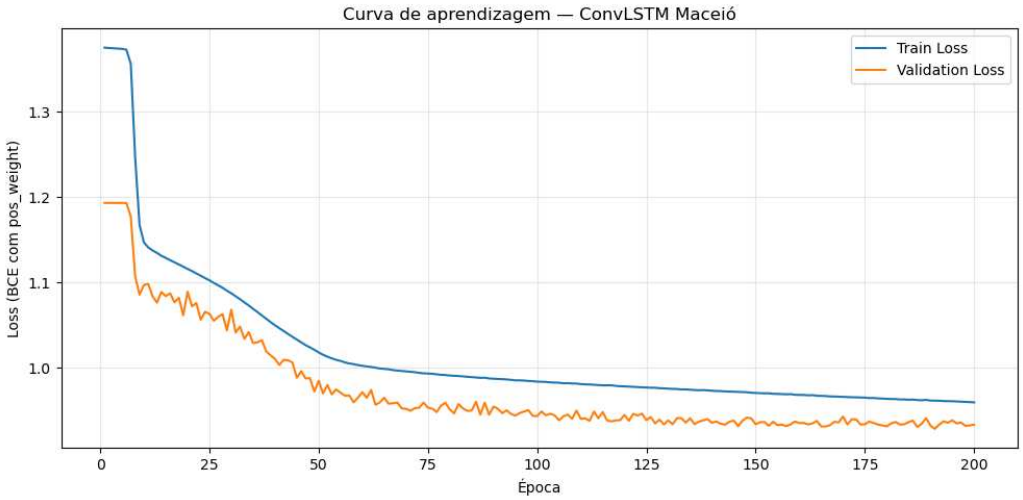
Fonte: Elaborado pelo autor (2026).

Três aspectos merecem destaque. Primeiro, a perda de validação não apresenta sinais claros de *overfitting*: ela permanece sistematicamente abaixo da perda de treino ao longo das 200 épocas, com diferença final inferior a 0,02. Esse comportamento é coerente com o uso de *dropout* elevado ($p = 0{,}8$) e com o processo de extração aleatória de sub-grids, que introduz variabilidade espacial em cada época. Segundo, a perda de validação atinge seu valor mínimo na última época, indicando que o treinamento foi interrompido pelo limite de épocas estipulado e que o modelo possivelmente ainda se beneficiaria de épocas adicionais. Terceiro, a redução das perdas é mais acentuada nas primeiras $\sim$50 épocas e desacelera gradualmente nas etapas seguintes, comportamento típico de cenários com classes severamente desbalanceadas e taxa de aprendizado baixa (10^{-5}).

4.2.2 Maceió

A Figura 5 apresenta a evolução das perdas de treino e validação ao longo das 200 épocas de treinamento da ConvLSTM em Maceió, com `seed` $= 42$. A perda de treino parte de 1,3753 na primeira época e atinge 0,9592 na época final; a perda de validação parte de 1,1932 e termina em 0,9331, com valor mínimo de 0,9281 atingido na época 191.

Figura 5 – Curva de aprendizagem da ConvLSTM em Maceió: perda de treino e perda de validação por época (`seed` = 42, 200 épocas).



Fonte: Elaborado pelo autor (2026).

A perda de validação permanece sistematicamente abaixo da perda de treino ao longo das 200 épocas, com diferença final inferior a 0,03, indicando ausência de *overfitting* apesar do número elevado de épocas. O mínimo da perda de validação é alcançado próximo ao fim do treinamento (época 191 de 200), padrão também observado em Arapiraca, sugerindo que ambas as cidades poderiam se beneficiar de épocas adicionais ou de um agendamento de *learning rate* ajustado, como discutido nos trabalhos futuros (Capítulo 5).

4.3 MÉTRICAS POR LIMIAR DE CLASSIFICAÇÃO

A saída do modelo, após a aplicação da função sigmoide, fornece uma probabilidade de ocorrência de crime por célula. A conversão dessa probabilidade em classe binária exige a escolha de um limiar de classificação τ: células com probabilidade $\geq \tau$ são classificadas como positivas. Foram avaliados 51 valores de limiar no intervalo $[0{,}50;\ 1{,}00]$, com passo de 0,01. Para cada limiar foram calculadas métricas clássicas de classificação binária (*recall*, *precision* e F1) e suas variantes com tolerância espacial, nas quais um falso positivo é reclassificado como acerto caso exista um crime real em uma célula vizinha (distância de Chebyshev ≤ 1), conforme descrito na Seção 3.5.3.

4.3.1 Arapiraca

4.3.1.1 Métricas tradicionais

A Tabela 2 apresenta as métricas tradicionais para Arapiraca em limiares representativos. O comportamento observado é característico de problemas com classes severamente desbalanceadas: à medida que o limiar aumenta, o *recall* cai rapidamente, enquanto a *precision* permanece muito

baixa e não ultrapassa $\sim$2,3% em nenhum dos 51 pontos avaliados.

Tabela 2 – Métricas tradicionais (sem tolerância espacial) por limiar de classificação para Arapiraca (`seed` $= 42$, $N = 57$).

Limiar	Recall	Precision	F1
0,50	0,6415	0,0170	0,0316
0,55	0,5740	0,0174	0,0326
0,60	0,4873	0,0190	0,0344
0,65	0,3908	0,0181	0,0334
0,70	0,2997	0,0201	0,0354
0,74	**0,2274**	**0,0226**	**0,0379**
0,75	0,2000	0,0225	0,0366
0,80	0,1081	0,0199	0,0303
0,85	0,0350	0,0111	0,0149
0,90	0,0061	0,0039	0,0041

Fonte: Elaborado pelo autor (2026).

O melhor F1 tradicional foi observado em $\tau = 0{,}74$, com valor de 0,0379 (*recall* $= 0{,}2274$ e *precision* $= 0{,}0226$). A *precision* permanece sistematicamente abaixo de 2,5% em toda a varredura, indicando que a esmagadora maioria dos alertas gerados pelo modelo não corresponde a ocorrências reais na célula exata. Esse padrão é consistente com os achados de Albors Zumel, Tizzoni e Campedelli (2025) em quatro cidades norte-americanas e reforça a tese de que a métrica célula a célula é excessivamente rigorosa para problemas de previsão criminal em *grids* de alta resolução.

4.3.1.2 Métricas com tolerância espacial

A Tabela 3 apresenta as métricas com tolerância espacial (vizinhança de Chebyshev ≤ 1) para os mesmos limiares. A mudança de perspectiva, passar a considerar acerto qualquer previsão que caia em uma das 8 células vizinhas de um crime real, produz melhora substancial em todas as métricas.

O melhor F1 espacial foi obtido em $\tau = 0{,}62$, atingindo 0,2063 (*recall* espacial $= 0{,}5860$ e *precision* espacial $= 0{,}1380$). Em $\tau = 0{,}50$, o modelo captura 73,2% dos crimes reais na vizinhança imediata, ao custo de uma *precision* espacial de 12,0%. Comparando as Tabelas 2 e 3, observa-se que a tolerância espacial multiplica o F1 por um fator de aproximadamente 5,4$\times$ no limiar ótimo, evidenciando que o modelo aprende a localizar corretamente vizinhanças de risco, ainda que erre a célula exata.

4.3.2 Maceió

4.3.2.1 Métricas tradicionais

A Tabela 4 apresenta as métricas tradicionais para limiares representativos. O comportamento observado é característico de problemas com classes severamente desbalanceadas: à medida que

Tabela 3 – Métricas com tolerância espacial (vizinhança de Chebyshev ≤ 1) por limiar de classificação para Arapiraca (`seed` $= 42$, $N = 57$).

Limiar	Recall espacial	Precision espacial	F1 espacial
0,50	0,7319	0,1197	0,1934
0,55	0,6835	0,1284	0,2015
0,60	0,6155	0,1336	0,2050
0,62	**0,5860**	**0,1380**	**0,2063**
0,65	0,5327	0,1364	0,2023
0,70	0,4434	0,1475	0,2038
0,75	0,3350	0,1485	0,1904
0,80	0,2099	0,1306	0,1481
0,85	0,0957	0,0851	0,0835
0,90	0,0163	0,0212	0,0176

Fonte: Elaborado pelo autor (2026).

o limiar aumenta, o *recall* cai rapidamente enquanto a *precision* permanece muito baixa, não ultrapassando $\sim$3,0%.

Tabela 4 – Métricas tradicionais (sem tolerância espacial) por limiar de classificação para Maceió (`seed` $= 42$, $N = 73$).

Limiar	Recall	Precision	F1
0,50	0,6502	0,0202	0,0378
0,55	0,5747	0,0214	0,0397
0,60	0,4785	0,0219	0,0407
0,65	0,3821	0,0237	0,0430
0,70	0,2886	0,0266	0,0456
0,75	**0,1981**	**0,0297**	**0,0476**
0,80	0,1064	0,0300	0,0417
0,85	0,0318	0,0185	0,0207
0,90	0,0026	0,0036	0,0029

Fonte: Elaborado pelo autor (2026).

O melhor F1 tradicional foi observado em $\tau = 0{,}75$, com valor de 0,0476 (*recall* $= 0{,}1981$ e *precision* $= 0{,}0297$). A *precision* permanece inferior a 3,0% em todos os limiares, indicando que a grande maioria dos alertas gerados pelo modelo não corresponde a ocorrências reais na célula exata. Esse padrão é consistente com os achados de Albors Zumel, Tizzoni e Campedelli (2025) em quatro cidades norte-americanas e reforça a tese de que a métrica célula a célula é excessivamente rigorosa para problemas de previsão criminal em *grids* de alta resolução.

4.3.2.2 Métricas com tolerância espacial

A Tabela 5 apresenta as métricas com tolerância espacial (vizinhança de Chebyshev ≤ 1) para os mesmos limiares. A mudança de perspectiva, passar a considerar acerto qualquer previsão que caia em uma das 8 células vizinhas de um crime real, produz melhora em todas as métricas.

O melhor F1 espacial foi obtido em $\tau = 0{,}68$, atingindo 0,2405 (*recall* espacial $= 0{,}5414$

Tabela 5 – Métricas com tolerância espacial (vizinhança de Chebyshev ≤ 1) por limiar de classificação para Maceió (`seed` $= 42$, $N = 73$).

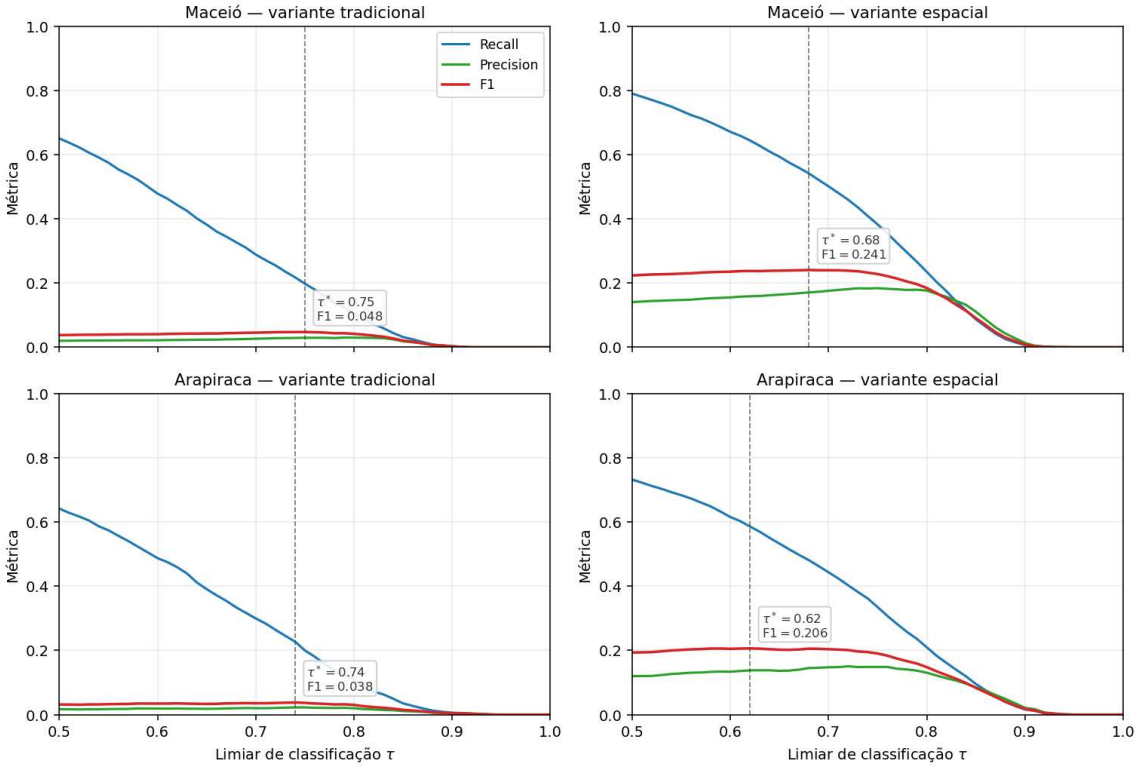
Limiar	Recall espacial	Precision espacial	F1 espacial
0,50	0,7899	0,1405	0,2235
0,55	0,7364	0,1472	0,2293
0,60	0,6710	0,1552	0,2353
0,65	0,5935	0,1640	0,2386
0,68	**0,5414**	**0,1710**	**0,2405**
0,70	0,5013	0,1758	0,2398
0,75	0,3828	0,1840	0,2281
0,80	0,2350	0,1766	0,1847
0,85	0,0871	0,1109	0,0909
0,90	0,0068	0,0136	0,0087

Fonte: Elaborado pelo autor (2026).

e *precision* espacial $= 0{,}1710$). Em $\tau = 0{,}50$, o modelo captura 79,0% dos crimes reais na vizinhança imediata, ao custo de uma *precision* espacial de 14,1%. Comparando as Tabelas 4 e 5, observa-se que a tolerância espacial multiplica o F1 por um fator de aproximadamente $5{,}1\times$ no limiar ótimo, evidenciando que o modelo aprende a localizar corretamente vizinhanças de risco, ainda que erre a célula exata.

A Figura 6 consolida o comportamento das métricas em função do limiar de classificação para as duas cidades e as duas variantes (tradicional e espacial). Em cada subplot, as três curvas correspondem a *recall*, *precision* e F1, e a linha vertical tracejada indica o limiar τ^* que maximiza o F1 daquela variante.

Figura 6 – Curvas de *recall*, *precision* e F1 em função do limiar de classificação τ para Maceió e Arapiraca, nas variantes tradicional (esquerda) e com tolerância espacial (direita). A linha vertical tracejada em cada subplot marca o τ^* que maximiza o F1 daquela variante; o valor de F1 nesse ponto é indicado em destaque.



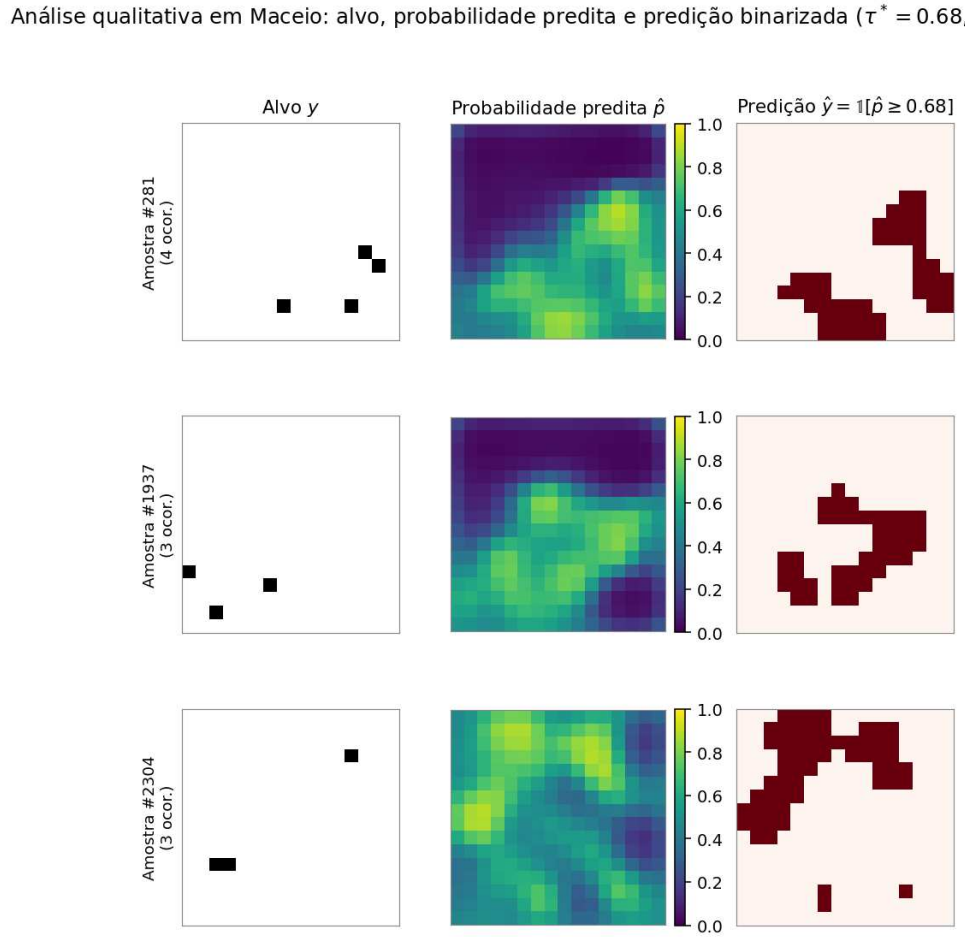
Fonte: Elaborado pelo autor (2026).

A leitura da figura permite duas observações qualitativas relevantes. Primeiro, o *recall* é monotonamente decrescente com τ em todas as quatro configurações, ao passo que a *precision* apresenta um ligeiro pico em valores intermediários antes de cair: esse equilíbrio entre as duas métricas é precisamente o que o F1 captura, e o τ^* ótimo ocorre nas regiões onde *precision* ainda cresce e *recall* ainda não colapsou. Segundo, a curva do F1 é visivelmente mais alta nos subplots da variante espacial em relação aos da tradicional, ilustrando graficamente o ganho de aproximadamente $5\times$ proporcionado pela tolerância de Chebyshev ≤ 1, em ambas as cidades.

4.4 ANÁLISE QUALITATIVA

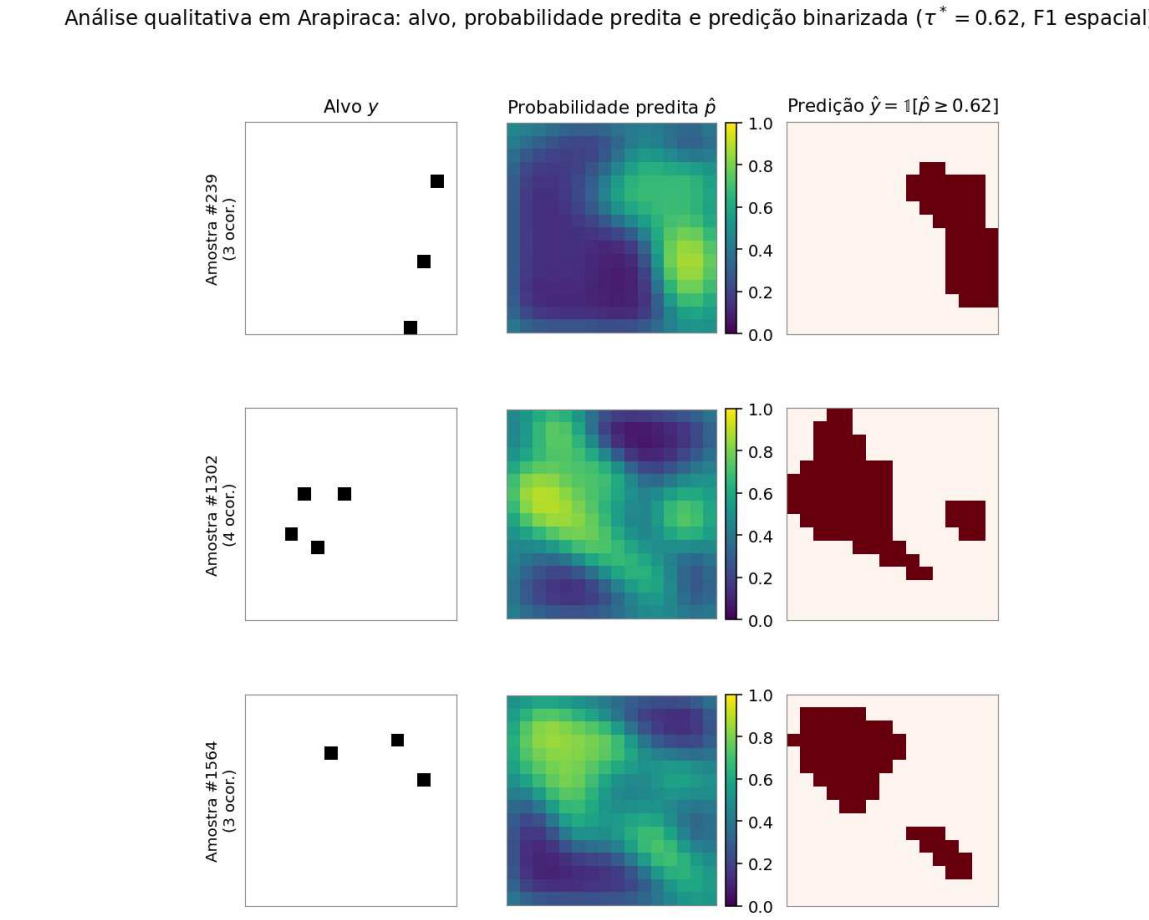
As Figuras 7 e 8 apresentam, respectivamente para Maceió e Arapiraca, três amostras do conjunto de validação ilustradas em três painéis lado a lado: a coluna da esquerda mostra o alvo real y (mapa binário das próximas 12h), a coluna do meio mostra a probabilidade predita $\hat{p}$ pela rede após a aplicação da sigmoide, e a coluna da direita mostra a predição binarizada $\hat{y} = \mathbf{1}[\hat{p} \geq \tau^*]$ no limiar ótimo de F1 espacial ($\tau^* = 0{,}68$ para Maceió e $\tau^* = 0{,}62$ para Arapiraca). As amostras foram sorteadas aleatoriamente, com semente fixa, dentre aquelas que apresentam pelo menos três ocorrências no alvo, garantindo visibilidade do sinal positivo.

Figura 7 – Análise qualitativa em Maceió. Cada linha corresponde a uma amostra distinta do conjunto de validação; as três colunas mostram, respectivamente, o alvo real, a probabilidade predita pela ConvLSTM e a predição binarizada em $\tau^* = 0{,}68$. O número de ocorrências reais no alvo é indicado à esquerda de cada linha.



Fonte: Elaborado pelo autor (2026).

Figura 8 – Análise qualitativa em Arapiraca. Mesmo padrão de leitura da Figura 7, com a binarização realizada em $\tau^* = 0{,}62$.



Fonte: Elaborado pelo autor (2026).

A leitura conjunta das duas figuras evidencia três comportamentos consistentes com as métricas reportadas anteriormente. Primeiro, os mapas de probabilidade predita não exibem ativações arbitrárias: em ambas as cidades, $\hat{p}$ tende a se concentrar em sub-regiões compatíveis com onde estão as ocorrências reais, ainda que raramente coincida com a célula exata. Segundo, a binarização sob o limiar ótimo de F1 espacial é geralmente mais ampla do que o conjunto de células efetivamente positivas, refletindo o trade-off necessário para maximizar o equilíbrio entre *precision* e *recall* sob desbalanceamento severo. Terceiro, em vários casos a predição binarizada inclui células imediatamente vizinhas a uma ocorrência real, o que ilustra concretamente por que a tolerância de Chebyshev ≤ 1 produz ganho substancial de F1 em relação à métrica célula a célula.

4.5 DISCUSSÃO

A análise conjunta dos resultados das duas cidades permite as seguintes observações:

- **Desbalanceamento dominante.** A esparsidade extrema dos dados é o fator estrutural

mais relevante em ambas as cidades. Em Arapiraca, das 580.608 posições célula-tempo do conjunto de validação extraído dos sub-grids 16×16, apenas 2.986 ($\approx 0{,}51\%$) correspondem a ocorrências reais; em Maceió, das 774.144 posições, 4.563 ($\approx 0{,}59\%$) são positivas. Como consequência direta, a *precision* tradicional permanece abaixo de 2,5% em Arapiraca e abaixo de 3,0% em Maceió em todos os limiares avaliados. Esse padrão corrobora o diagnóstico de Albors Zumel, Tizzoni e Campedelli (2025) de que a tarefa célula a célula é fundamentalmente difícil sob classes tão desbalanceadas.

- **Utilidade prática na vizinhança.** A tolerância espacial produz melhora substancial em ambas as cidades. Em Arapiraca, o F1 salta de 0,0379 (tradicional) para 0,2063 (espacial), fator de aproximadamente 5,4×; em Maceió, o F1 sobe de 0,0476 para 0,2405, fator semelhante de $\sim$5,1×. Como cada célula do *grid* tem $\sim$0,2 km^2 de área, uma vizinhança de Chebyshev ≤ 1 corresponde a uma janela de aproximadamente 3×3 células ($\sim$1,8 km^2), granularidade coerente com o porte de uma operação de patrulhamento preventivo.

- **Convergência do treinamento.** Tanto em Arapiraca quanto em Maceió, a perda de validação permanece sistematicamente abaixo da perda de treino ao longo das 200 épocas, sem evidências de *overfitting* (gap final inferior a 0,03 em Maceió e a 0,02 em Arapiraca). O mínimo da perda de validação é atingido próximo ao fim do treinamento (época 191 de 200 em Maceió; época 200 de 200 em Arapiraca). Esse comportamento, somado ao uso de *dropout* elevado e à amostragem aleatória de sub-grids, sugere que o modelo poderia se beneficiar de épocas adicionais ou de uma taxa de aprendizado ligeiramente maior, pontos que podem ser explorados em trabalhos futuros.

- **Escolha do limiar.** Para Maceió, os limiares ótimos são $\tau = 0{,}75$ (F1 tradicional) e $\tau = 0{,}68$ (F1 espacial). Para Arapiraca, esses valores são $\tau = 0{,}74$ e $\tau = 0{,}62$, respectivamente. Os limiares ótimos relativamente altos para a variante tradicional em ambas as cidades indicam que o modelo precisa adotar um critério bastante conservador para tentar elevar a *precision*, ao custo de *recall* reduzido. Na variante espacial, Maceió equilibra-se em um limiar um pouco mais alto que Arapiraca, sugerindo predições marginalmente mais calibradas em virtude do maior volume de dados disponível para treinamento.

- **Comparação entre cidades.** Maceió apresenta desempenho superior a Arapiraca em todas as métricas espaciais: F1 espacial ótimo de 0,2405 contra 0,2063 (+17%), *recall* espacial de 54,1% contra 58,6% no respectivo limiar ótimo, e *precision* espacial de 17,1% contra 13,8%. Em $\tau = 0{,}50$, o desempenho de Maceió ainda é mais expressivo: 79,0% dos crimes são capturados na vizinhança imediata, contra 73,2% em Arapiraca. Esse padrão é coerente com a diferença de volume de dados (Maceió dispõe de 72.649 ocorrências contra 17.356 em Arapiraca, fornecendo ao modelo uma base de treinamento mais rica) e com a menor esparsidade do conjunto de treino de Maceió (`pos_weight` $\approx 123{,}4$ contra $\approx 163{,}3$ em Arapiraca).

- **Limitações.** Os valores reportados referem-se a uma única execução com `seed` $= 42$, sem médias entre sementes (no estudo de referência são utilizadas as sementes 0, 42, 123 e 999). Também não foram executados *baselines* (persistência, média móvel, regressão logística, *Random Forest*, LSTM padrão) para quantificar o ganho efetivo da ConvLSTM, nem foram incorporados canais adicionais de entrada (mobilidade, sociodemografia), limitações já elencadas entre os trabalhos futuros.

5 CONCLUSÃO

Este trabalho propôs a aplicação de uma arquitetura ConvLSTM para a predição de crimes violentos contra o patrimônio nas cidades de Maceió e Arapiraca, utilizando dados georreferenciados de ocorrências entre 2012 e 2022 cedidos pela Polícia Militar de Alagoas. O pipeline desenvolvido abrangeu desde o pré-processamento dos dados brutos (limpeza, filtragem por tipo de crime, filtragem espacial por polígono municipal obtido via OpenStreetMap) até a construção de representações espaço-temporais em *grids* regulares de área-alvo $\sim$0,2 km^2, a definição de janelas deslizantes de 14 dias em granularidade de 12h, a extração de sub-grids 16×16, o treinamento da ConvLSTM com tratamento explícito do desbalanceamento via `BCEWithLogitsLoss` ponderada, e a avaliação por varredura de limiar de classificação com métricas tradicionais e com tolerância espacial.

Os resultados obtidos para as duas cidades indicam que o modelo ConvLSTM é capaz de identificar regiões com maior probabilidade de ocorrência criminal, embora com limitações significativas na precisão célula a célula. Para Maceió ($N = 73$, 72.649 ocorrências), o melhor F1 tradicional foi de 0,0476 ($\tau = 0{,}75$) e o melhor F1 espacial de 0,2405 ($\tau = 0{,}68$), com *recall* espacial de 54,1% no respectivo limiar ótimo. Para Arapiraca ($N = 57$, 17.356 ocorrências), o melhor F1 tradicional foi de 0,0379 ($\tau = 0{,}74$) e o melhor F1 espacial de 0,2063 ($\tau = 0{,}62$), com *recall* espacial de 58,6%. Em ambas as cidades, a perda de validação permaneceu sistematicamente abaixo da perda de treino ao longo das 200 épocas, sem evidências de *overfitting*, e o mínimo da perda de validação foi atingido próximo ao fim do treinamento (época 191 em Maceió e época 200 em Arapiraca), sugerindo que o modelo poderia se beneficiar de mais épocas.

A comparação entre as cidades revela que o desempenho do modelo é consistentemente superior em Maceió, o que pode ser atribuído ao maior volume de ocorrências disponíveis para treinamento (72.649 contra 17.356) e à menor esparsidade do conjunto de treino ($\sim$0,80% contra $\sim$0,61% de células positivas nos sub-grids). Em ambos os casos, a tolerância espacial (vizinhança de Chebyshev ≤ 1) produziu melhora expressiva, multiplicando o F1 por fatores de aproximadamente $5{,}1\times$ em Maceió e $5{,}4\times$ em Arapiraca, evidenciando que o modelo aprende a localizar corretamente vizinhanças de risco mesmo quando erra a célula exata.

É importante destacar algumas limitações do presente estudo. A *precision* tradicional permaneceu sistematicamente baixa em todos os limiares (inferior a 3,0% em Maceió e a 2,5% em Arapiraca), o que significa que a grande maioria dos alertas não corresponde a ocorrências reais na célula exata. Esse padrão é consistente com os achados de Albors Zumel, Tizzoni e Campedelli (2025), que reporta limitações análogas em cidades norte-americanas e justificou a introdução, naquele estudo e replicada aqui, da avaliação com tolerância espacial. Além disso, o pipeline utiliza apenas o canal de ocorrência criminal, sem incorporar dados sociodemográficos ou de mobilidade humana, e adota uma única semente aleatória (42), em contraste com as

quatro sementes utilizadas no estudo de referência. Por fim, não foram executados modelos *baseline* (persistência do último mapa, média móvel, regressão logística, *random forest*, LSTM padrão) que permitiriam quantificar o ganho efetivo da ConvLSTM em relação a abordagens mais simples.

5.1 TRABALHOS FUTUROS

Com base nas limitações identificadas e nas possibilidades abertas pelo pipeline desenvolvido, os seguintes trabalhos futuros podem ser realizados:

- Adotar um *split* cronológico $80/10/10$ (treino, validação e teste), com a seleção do limiar de classificação ótimo realizada exclusivamente sobre a partição de validação e o reporte das métricas finais feito sobre a partição de teste. Essa separação eliminaria o viés otimista presente na configuração atual, em que validação e reporte coincidem, e está alinhada às boas práticas metodológicas de aprendizado supervisionado, ainda que afaste o pipeline da configuração $90/10$ adotada por Albors Zumel, Tizzoni e Campedelli (2025);
- Repetir o treinamento das duas cidades com múltiplas sementes aleatórias (por exemplo, $\{0, 42, 123, 999\}$, conforme o estudo de referência), reportando médias e desvios-padrão das métricas em vez de valores pontuais;
- Implementar e comparar modelos *baseline* (persistência do último mapa, média móvel, regressão logística, *Random Forest* e LSTM padrão) para quantificar o ganho efetivo da ConvLSTM em relação a alternativas mais simples;
- Estender o treinamento para mais épocas ou avaliar variações da taxa de aprendizado: as curvas de validação de Maceió e de Arapiraca atingem seus mínimos próximo ao fim do treinamento (épocas 191 e 200, respectivamente), indicando que o modelo possivelmente ainda se beneficiaria de mais iterações ou de um agendamento de *learning rate* ajustado;
- Incorporar canais adicionais de entrada (dados sociodemográficos do IBGE e dados de mobilidade humana) para avaliar se informações contextuais melhoram o desempenho preditivo, em linha com a configuração de melhores resultados reportada por Albors Zumel, Tizzoni e Campedelli (2025);
- Experimentar diferentes granularidades temporais (por exemplo, 8h ou 24h) e tamanhos de *lookback* (entre 2 e 28 dias), comparando o impacto sobre a previsão de crimes violentos contra o patrimônio em Alagoas;
- Avaliar o impacto de diferentes tamanhos de *grid* e de sub-grid na capacidade preditiva, eventualmente substituindo a amostragem aleatória de sub-grids por treinamento sobre o *grid* completo, caso o orçamento computacional permita;

- Investigar técnicas de aumento de dados e estratégias alternativas de balanceamento (*focal loss*, *oversampling* de amostras positivas) para mitigar o impacto da esparsidade extrema, sobretudo em Arapiraca;
- Estudar o critério de aceitação de sub-grids: este trabalho relaxou o limite mínimo de ocorrências por sub-grid de ≥ 2 (paper de referência) para ≥ 1, e seria interessante quantificar o impacto desse parâmetro sobre as métricas finais;
- Explorar a generalização do modelo por meio de transferência de aprendizado entre cidades (treinamento em Maceió e avaliação em Arapiraca, e vice-versa), de modo a verificar se padrões aprendidos em uma cidade se transferem para a outra.

REFERÊNCIAS

Albors Zumel, A.; TIZZONI, M.; CAMPEDELLI, G. M. Deep learning for crime forecasting: The role of mobility at fine-grained spatiotemporal scales. **Journal of Quantitative Criminology**, set. 2025. ISSN 0748-4518, 1573-7799. Disponível em: <https://link.springer.com/10.1007/s10940-025-09629-3>.

ALVES, L. G.; RIBEIRO, H. V.; RODRIGUES, F. A. Crime prediction through urban metrics and statistical learning. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 505, p. 435–443, 2018.

BEATO, C.; SILVA, B. F. A. d.; TAVARES, R. Crime e estratégias de policiamento em espaços urbanos. **Dados**, SciELO Brasil, v. 51, n. 3, p. 687–717, 2008.

Camacho Doyle, M.; GERELL, M.; ANDERSHED, H. Perceived unsafety and fear of crime: The role of violent and property crime, neighborhood characteristics, and prior perceived unsafety and fear of crime. **Deviant behavior**, v. 43, n. 11, p. 1347–1365, 2022. ISSN 0163-9625. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/01639625.2021.1982657>.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: Synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321–357, 2002.

DEZA, M. M.; DEZA, E. **Encyclopedia of Distances**. Berlin Heidelberg: Springer, 2009.

FE, H.; SANFELICE, V. How bad is crime for business? evidence from consumer behavior. **Journal of urban economics**, Elsevier, v. 129, p. 103448, 2022.

FURTADO, M. I. V. **Redes neurais artificiais: uma abordagem para sala de aula**. [S.l.]: Atena Editora, 2019. v. 19. ISBN 978-85-7247-326-2.

Fórum Brasileiro de Segurança Pública. **19º Anuário Brasileiro de Segurança Pública**. São Paulo: Fórum Brasileiro de Segurança Pública, 2025. 434 p. Anual. Ano 19 (2025). Acesso em: 19 maio 2026. ISSN 1983-7364. Disponível em: <https://publicacoes.forumseguranca.org.br/handle/123456789/279>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GUPTA, A. Introduction to deep learning: Part 1. **Chemical Engineering Progress**, AIChE New York, NY, USA, v. 114, n. 6, p. 22–29, 2018.

HASSAN, W.; CABRAL, M. M.; RAMOS, T. R.; FILHO, A. C.; NONATO, L. G. Modeling and predicting crimes in the city of são paulo using graph neural networks. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2024. p. 372–386.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. **International conference on machine learning**. [S.l.], 2015. p. 448–456.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

LECUN, Y.; BENGIO, Y. Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, 1998.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LIN, T.-Y.; GOYAL, P.; GIRSHICK, R.; HE, K.; DOLLÁR, P. Focal loss for dense object detection. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 2980–2988.

LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. A critical review of recurrent neural networks for sequence learning. **arXiv preprint arXiv:1506.00019**, 2015.

LOSHCHILOV, I.; HUTTER, F. Sgdr: Stochastic gradient descent with warm restarts. **arXiv preprint arXiv:1608.03983**, 2016.

MAXWELL, A. E.; WARNER, T. A. Thematic classification accuracy assessment with inherently uncertain boundaries: An argument for center-weighted accuracy assessment metrics. **Remote Sensing**, MDPI, v. 12, n. 12, p. 1905, 2020.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814.

PAPACHRISTOS, A. V. The promises and perils of crime prediction. **Nature Human Behaviour**, v. 6, n. 8, p. 1038–1039, 2022. ISSN 2397-3374. Disponível em: <https://www.nature.com/articles/s41562-022-01373-z>.

PyTorch contributors. **BCEWithLogitsLoss**. 2025. Documentação oficial do PyTorch.

ROTARU, V.; HUANG, Y.; LI, T.; EVANS, J.; CHATTOPADHYAY, I. Event-level prediction of urban crime reveals a signature of enforcement bias in us cities. **Nature human behaviour**, Nature Publishing Group UK London, v. 6, n. 8, p. 1056–1068, 2022.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. **PLoS ONE**, v. 10, n. 3, p. e0118432, 2015.

scikit-learn developers. **Tuning the decision threshold for class prediction**. 2025. Documentação oficial do scikit-learn.

SHI, X.; CHEN, Z.; WANG, H.; YEUNG, D.-Y.; WONG, W.-K.; WOO, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. **Advances in neural information processing systems**, v. 28, 2015.

SHI, X.; GAO, Z.; LAUSEN, L.; WANG, H.; YEUNG, D.-Y.; WONG, W.-k.; WOO, W.-c. Deep learning for precipitation nowcasting: A benchmark and a new model. **Advances in neural information processing systems**, v. 30, 2017.

SOARES, M.; SABOYA, R. T. d. Fatores espaciais da ocorrência criminal: modelo estruturador para a análise de evidências empíricas. **Urbe. Revista Brasileira de Gestão Urbana**, SciELO Brasil, v. 11, p. e20170236, 2019.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, v. 45, n. 4, p. 427–437, 2009.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SRIVASTAVA, N.; MANSIMOV, E.; SALAKHUDINOV, R. Unsupervised learning of video representations using lstms. In: PMLR. **International conference on machine learning**. [S.l.], 2015. p. 843–852.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. **Advances in neural information processing systems**, v. 27, 2014.

WANG, B.; YIN, P.; BERTOZZI, A. L.; BRANTINGHAM, P. J.; OSHER, S. J.; XIN, J. Deep learning for real-time crime forecasting and its ternarization. **Chinese Annals of Mathematics, Series B**, Springer, v. 40, n. 6, p. 949–966, 2019.

WEISBURD, D. The law of crime concentration and the criminology of place. **Criminology**, v. 53, n. 8, p. 133–157, 2015. ISSN 2397-3374. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1745-9125.12070>.