

# Previsão do nível do rio Itajaí-Açú durante situações de enchente em Blumenau utilizando redes neurais recorrentes

Jonathan de Souza, Aurélio Faustino Hoppe – orientador

Curso de Especialização em Data Science  
Universidade Regional de Blumenau (FURB) – Blumenau/SC – Brasil

johnnysouza3@gmail.com, aureliof@furb.br

**Resumo.** Enchentes são eventos climáticos extremos, especialmente danosos em relevos acidentados como no Vale do Itajaí onde se localiza a cidade de Blumenau e o rio Itajaí-Açú. Comumente para se prever as enchentes são utilizados modelos manuais complexos, que exigem muitos dados de entrada para aferirem previsões com boa assertividade. Nesse contexto, este trabalho apresenta a construção de modelos de redes neurais recorrentes para prever o nível do rio Itajaí-Açú em Blumenau. Os dados de precipitação e nível do rio foram tratados para exclusão de valores repetidos, preenchimento dos períodos faltantes e agrupados em eventos de cheias quando o rio ultrapassa 4 metros. Foram criados modelos de redes neurais recorrentes do tipo Long Short-Term Memory e Gated Recurrent Unit, para previsões de 3, 6 e 9 horas futuras. As métricas avaliadas foram a taxa do erro quadrático médio como função de perda e a acurácia percentual utilizando uma margem de erro com 5% acima e 3% abaixo dos valores reais. A partir dos experimentos realizados, notou-se que previsões mais longas possuem menor precisão e que redes neurais do tipo Gated Recurrent Unit possuem melhor performance, com acurácias de aproximadamente 90%, 80% e 79% nas respectivas previsões para 3, 6 e 9 horas sobre o conjunto de dados de desenvolvimento.

**Palavras-chave:** Previsão de enchentes. Blumenau. Itajaí-Açú. Redes Neurais. Long Short Term Memory. Gated Recurrent Unit.

## 1. Introdução

As enchentes são calamidades causadas pelo excesso de chuva em determinada região com destruição de propriedades, perdas de vidas e diversos impactos socioeconômicos [1]. Tais eventos podem ser potencializados em localidades onde o relevo é acidentado, facilitando o rápido escoamento das águas pluviais em direção a terrenos mais baixos. Isso acontece de forma contundente na região da bacia hidrográfica do rio Itajaí-Açú, onde se localiza a cidade de Blumenau, que recebe as águas escoadas dos pontos mais altos do Vale do Itajaí [2].

Blumenau já foi atingida por diversas enchentes ao longo dos anos, sendo os casos mais notórios em 1983 e 1984, a tragédia de 2008 e mais recentemente a cheia de 2011 [1]. Devido à recorrência, a cidade possui uma atenção especial para o monitoramento e previsão do nível do rio Itajaí-Açú, provendo uma preparação que visa mitigar os impactos na vida das pessoas da região para saberem quando devem ser deslocar para evitar a cheia do rio.

A região em si, conta com diversas estações para captação de dados hidrológicos, principalmente através do Centro de Operações do Sistema de Alerta (CEOPS) de Blumenau [3]. Tais estações fornecem dados que ajudam na geração de previsões para modelos matemáticos calculados manualmente, que possuem grande complexidade e apresentam uma demora para geração das previsões do nível do rio Itajaí-Açú. Além disso, grande parte das estações hidrológicas brasileiras

apresentam apenas dados sobre precipitação, vazão e nível do rio. Havendo pouco volume de variáveis disponíveis, dificulta-se a adoção de modelos matemáticos mais complexos e precisos [3]. Outro fator importante se dá pela necessidade em mão de obra extremamente qualificada para criação, manutenção e utilização dos modelos manuais.

Visando abordar modelos de redes neurais para previsão de nível do rio, existem trabalhos similares como Agudelo-Otalora, L. M. et al [4], notando que os modelos hidrológicos utilizando métodos de regressão linear, sofrem em explicar eventos com características não lineares. Em outros casos são utilizados modelos baseados em leis físicas, mas que geram previsões instáveis e com muitos erros, pelas necessidades de muitas variáveis e informações a serem cruzadas, como nível do rio, chuva, dados geológicos, topográficos, tipos de solo, entre outros. Assim o trabalho comparou um modelo físico a um modelo de redes neurais artificiais (RNA), na bacia do rio Bogotá, concluindo que os modelos de redes neurais geraram melhores previsões do que os modelos físicos.

Silva et al. [5] focaram na predição mensal do nível do Rio Xingu, baseado nos dados históricos de precipitação da bacia, juntamente com a temperatura da superfície do mar. A partir disso, foram criados modelos de redes neurais MultiLayer Perceptron (MLP), utilizando como métricas de avaliação o Mean Square Error (MSE) e o coeficiente de determinação ( $R^2$ ). Foram atingidos valores de  $2,99 \times 10^{-4}$  para o MSE e 0,9991 para o  $R^2$ , sendo resultados satisfatórios que demonstraram a aplicabilidade de RNA em problemática não linear como a previsão de nível do rio.

A pesquisa de Soares e Teive [6], apresenta um modelo de RNA para previsão de curto prazo no Rio Itajaí-Açú na cidade de Rio do Sul. Foram utilizados os dados de duas estações acima do ponto de medição de saída como entradas para o modelo. Aplicaram-se três técnicas para avaliação dos resultados, o coeficiente de eficiência de Nash e Sutcliffe (CE), o Root Mean Squared Error (RMSE) e o Erro Absoluto Médio Percentual (EAMP). O melhor experimentos atingiu os respectivos valores de 0,9788, 0,0202 e 5,8024, com isso os autores concluíram que o modelo criado apresentou bons resultados perante o baixo volume de entradas disponíveis para treinamento.

Segundo Alberton et al. [3], as medidas estruturais como sistemas de alertas, ajudam na mitigação dos impactos de um evento de enchente. Nesse contexto, a longa série histórica disponível no município de Blumenau e região serve como bom objeto de estudo para avaliar a eficácia de modelos de redes neurais na previsão de tais eventos. Foi avaliada a aplicação de redes neurais do tipo Multilayer Perceptron e Long Short-Term Memory (LSTM), sendo que o modelo LSTM apresentou melhor eficiência em suas previsões.

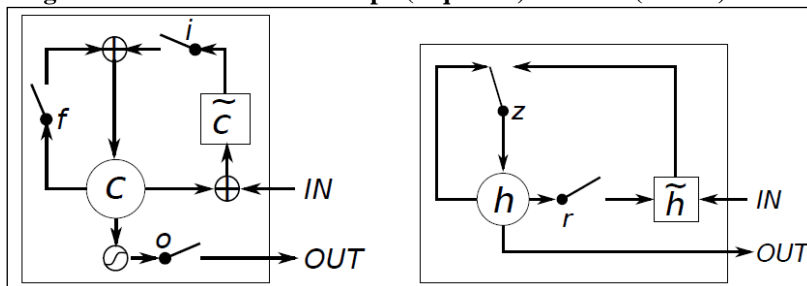
Visto os entraves para previsões com modelos manuais tradicionais e a vantagem dos modelos de redes neurais recorrente sobre os estilos mais tradicionais, este trabalho aborda a previsão do nível do rio Itajaí-Açú em Blumenau utilizando metodologias de RNR do tipo LSTM e Gated Recurrent Unit (GRU). Tais redes são propícias para trabalhos com dados em formato de séries temporais, como os dados históricos das estações hidrológicas na bacia do Itajaí. Diante deste cenário, tem-se como objetivo realizar a previsão de curto prazo para o nível do rio Itajaí-Açú em Blumenau, utilizando RNR durante situações em que o nível do rio ultrapasse 4 metros. Os objetivos específicos são: (a) verificar se RNR são assertivas nas precisões do nível do rio; (b) validar se o modelo gerado é adequado para auxiliar os modelos tradicionais de previsão do nível do rio. Para isso o estudo foi subdividido nos seguintes tópicos: revisão bibliográfica, desenvolvimento, resultados e conclusões.

## 2. Recurrent Neural Networks

Dada as características temporais dos eventos de enchentes, onde comportamentos passados de chuvas e a própria tendência de subida ou descida do nível do rio afetam o seu nível futuro, optou-se por utilizar a técnica de RNN. Em sequências de dados elas são eficazes tanto para classificação quanto para regressão de valores. Em ambos os casos, o treinamento dos dados de entrada cria uma representação das relações temporais através dos estados internos aprendidos [7].

As RNNs são úteis para analisar e extrair aprendizado de séries temporais, mas possuem limitações quanto a grandes sequências de dados. Nesses casos as informações mais antigas não conseguem serem propagadas pela rede por causa do frequente problema do desaparecimento de gradientes e do mais raro problema da explosão de gradientes [8]. Com isso, são abordadas as categorias de RNR do tipo LSTM e GRU, que visam aprimorar a passagem de informações passadas ao longo do tempo sem a ocorrência dos problemas inerentes ao repasse desses conhecimentos por longos períodos. A Figura 1 mostra as principais estruturas das células LSTM e GRU.

**Figura 1. Células de RNR do tipo (esquerda) LSTM e (direita) GRU.**



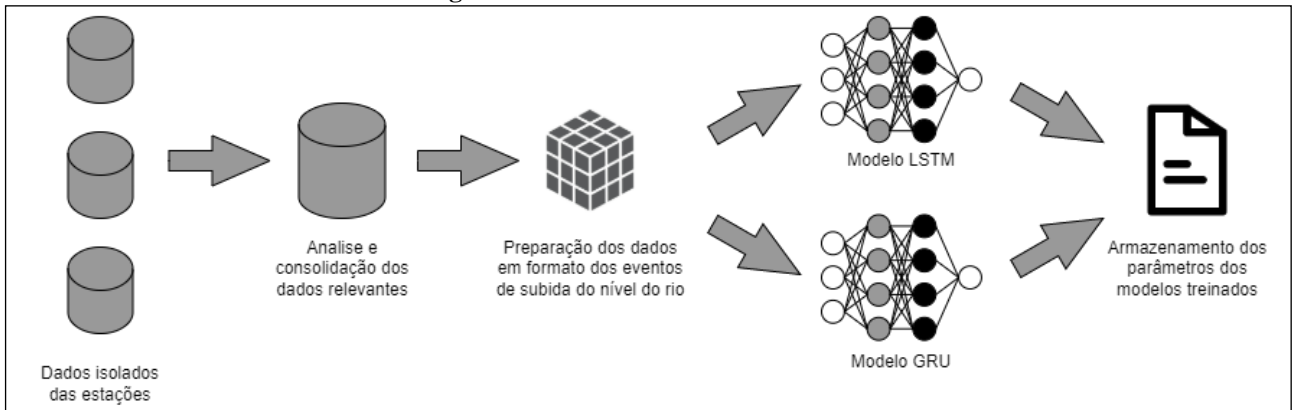
Conforme visto na Figura 1 (esquerda), uma unidade LSTM é composta por um portão de entrada  $i$ , um portão de esquecimento  $f$  e um portão de saída  $o$ . A célula interna de memória é denotada por  $c$ , enquanto a nova célula interna de nova memória é denotada por  $\tilde{c}$  [9]. Os portões de entrada, esquecimento e saída podem, respectivamente, escrever, redefinir ou ler os dados armazenados na célula de memória. Diferente das RNR tradicionais que sobrescrevem seu conteúdo a cada passagem de tempo, a LSTM decide através de seus portões se deve ou não manter partes da sua memória. Isso permite que uma característica inicialmente importante possa ser facilmente lembrada por um longo tempo e assim capturando potenciais dependência de longo termo [9].

Já as RNR do tipo GRU foram propostas inicialmente em 2014 para fazer cada unidade recorrente capturar de forma adaptativa dependências de diferentes escalas temporais [10]. De forma semelhante a unidade LSTM, a GRU possui portões que moldam o fluxo dentro da unidade, mas sem a existência de células internas de memória [9]. Conforme visto na Figura 1 (direita), uma unidade GRU possui um portão de reset  $r$  e um portão de atualização  $z$ , além da ativação  $h$  e da ativação candidata  $\tilde{h}$ . O portão de reset possui a função de decidir quantos dados do passado devem ser esquecidos, enquanto o portão de atualização limita os dados a serem repassados para a ativação. A ativação candidata considera o estado passado e quanto deve esquecê-lo, conforme o portão de reset, já a ativação soma os dados e transmite de volta a rede o que for mais relevante.

### 3. Método proposto

A Figura 2 apresenta o fluxo utilizado para realizar a predição do nível do rio Itajá-Açú. Pode-se observar que após a obtenção da base de dados, iniciou-se a análise das diversas estações. Os dados foram manipulados para identificar os eventos relevantes de cheias, sendo utilizados nos modelos de redes neurais recorrentes LSTM e GRU. No desenvolvimento do trabalho, utilizou-se a linguagem de programação Python e o framework de redes neurais Pytorch.

**Figura 2. Fluxo desenvolvimento.**



Na montagem da base foram utilizados dados de estações do CEOPS que de 15 em 15 minutos registram o horário da medição, nível do rio e precipitação do período. As cidades abrangidas por essas estações são: Alfredo Wagner, Apiúna, Benedito Novo, Blumenau, Botuverá, Brusque, Gaspar, Ibirama, Indaial, Ituporanga, Pouso Redondo, Rio do Oeste, Rio dos Cedros, Rio do Sul, Taió, Timbó e Vidal Ramos.

Destaca-se que foram descartados os dados das estações de Botuverá, Brusque e Vidal Ramos que pertencem a sub-bacia do rio Itajaí-Mirim, pois desaguam no Rio Itajaí-Açú no município de Itajaí no qual suas influências na vazão acabam sendo pouco relevantes. Por fim, também foram descartados os dados de precipitação na cidade de Gaspar, por não influenciarem diretamente no aumento do nível. No entanto, pode influenciar na vazão do rio em Blumenau por ser a próxima cidade na sequência do curso do rio.

Ressalta-se que muitas estações não possuíam o mesmo período de registros. Por exemplo, os dados de Blumenau, que era o alvo a ser considerado, iniciavam em 2009 e terminavam em 2020. Porém, muitas outras estações tinham suas observações encerradas em 2016, 2017 e 2018. Como muitas colunas continham informações faltantes, optou-se por considerar os dados de todas as estações até o término do último evento relevante de cheia em Blumenau, em 12/07/2017. Dessa forma, os dados de Rio do Sul que normalmente são utilizados como parâmetro para as águas que chegam em Blumenau, puderam ser mantidos sem prejudicar os modelos. Já os dados das estações de Benedito Novo, Gaspar, Rio dos Cedros, Alfredo Wagner e Pouso Redondo foram descartados, pois estavam comprometidos ou não representavam a série inteira. Na Figura 3 pode-se ver a localização das estações utilizadas para desenvolvimento dos modelos de redes neurais. A partir da legenda, pode-se entender quais dados que foram utilizados de cada estação, sendo o alvo da previsão o nível da estação de Blumenau.

Mapa da Bacia do Rio Uruguai, mostrando a rede hidrográfica e os pontos de coleta de água e precipitação. A legenda indica:

- Previsão (círculo rosa)
- Chuva/Nível do rio (círculo verde)
- Chuva (círculo azul)
- Não utilizada (círculo laranja)

A Equação 1 apresenta o cálculo da interpolação linear, onde  $y$  é o valor a ser interpolado,  $y_1$  e  $y_2$  são respectivamente seus valores anterior e posterior. Como valor correspondente de  $y$  tem-se o valor  $x$ , que assim como *este* possui  $x_1$  e  $x_2$ , sendo seus valores correspondentes anterior e posterior. Colocando em contexto,  $x$  é a data da medição e  $y$  o nível do rio que precisa ser predito.

O próximo passo consistiu em encontrar os eventos onde o nível do rio atingiu situação de atenção, passando de 4 metros. Caso o rio baixasse de 4 metros e volta-se a ultrapassar essa medida em menos de 30 horas, considerava-se o mesmo evento. Depois do mapeamento dos eventos, foram excluídos os eventos com lacunas de registros para Blumenau, mesmo após a interpolação linear de até 5 horas. Após a separação em eventos, restaram 19.018 registros para as variáveis de precipitação e nível do rio, mas nem todas estavam completamente preenchidas. Dessa forma, estabeleceu-se um limiar que para uma variável ser mantida nos eventos ela precisaria ter ao menos 13 mil registros reais, sendo removidas as variáveis com menos registros.

Para os eventos que sobraram, caso alguma outra estação ainda tivesse intervalo de dados, eles foram preenchidos pela mediana da estação durante aquele evento de cheia. Mesmo assim, verificou-se que ainda existiam dados vazios, por causa de estações que ficaram sem nenhum registro durante todo o evento de enchente. Nesses casos, os valores faltantes foram preenchidos com média daquela estação durante todos os eventos. Na Equação 2, encontra-se a fórmula da mediana utilizada para preenchimento desses casos, onde  $n$  é a quantidade de elementos contidos no conjunto de elementos e  $M$  é a posição da mediana, que deve ser arredondada para cima em caso de número ímpar e mantida em caso de número par de elementos.

$$M = \frac{n + 1}{2} \quad (2)$$

No total, foram estabelecidos 32 eventos de enchente a serem utilizados na criação dos modelos, sendo divididos em 26 para treino, 4 para desenvolvimento e 2 para validação, representando uma proporção aproximada de 75% para treino e 12,5% para desenvolvimento e 12,5% validação. Utilizou-se um percentual relativamente baixo dos eventos para treino por causa do baixo número total de eventos.

O processo de carregamento dos dados para treinamento do modelo consistiu na utilização de uma janela temporal, que dependendo da situação podia ser de 24, 27 ou 30 horas anteriores ao registro analisado. Isso significa que cada iteração do modelo da rede neural olha para os dados atuais e para essa janela passada e com isso buscava estabelecer as relações dos dados de cada estação para explicar os dados de 3, 6 e 9 horas no futuro do nível do rio em Blumenau.

A estrutura geral do modelo se dava pelos dados de entrada baseados na quantidade de variáveis utilizadas das estações, que ficou em 17. Depois havia a(s) camada(s) de RNR do tipo LSTM ou GRU, seguida por uma taxa estatística de *dropout* para ignorar determinados resultados da saída antes de repassá-los para uma(s) camada(s) totalmente conectada de neurônios tradicionais que por fim valoravam a predição final do nível futuro do rio em Blumenau, conforme a quantidade de horas futuras definida para o modelo. Cada uma dessas camadas tinha uma quantidade de neurônios que era informada na inicialização do modelo.

Optou-se por criar 6 modelos, sendo 3 LSTM e 3 GRU com previsões para 3, 6 e 9 horas futuras. Foram treinados modelos distintos, pois as estações possuem pesos diferentes na previsão conforme a quantidade de horas futuras a serem preditas. Isso acontece pois as chuvas e nível do rio de estações mais próximas influenciam mais na previsão de 3 horas e por terem menos relevância do que estações mais afastadas para uma previsão de 9 horas.

No treinamento, foi controlado a quantidade de épocas a serem executadas e adotada a técnica de pequenos lotes para reduzir o consumo de memória durante a execução. Em cada iteração foi executado o treinamento do modelo, aplicando a função de perda MSE e, posteriormente, um método de retro propagação para atualização dos pesos internos dos neurônios da rede, além da avaliação da perda sobre os dados de desenvolvimento.

Para avaliação em conjunto com a saída das perdas do modelo, criou-se uma função para calcular a acurácia do modelo, onde os dados de entrada eram passados para cálculo no modelo e avaliado se a saída estava dentro de uma margem de erro de 5% para mais ou 3% para menos para considerar como uma predição correta. Foi considerada uma margem inferior menor por motivos de maior impacto social caso o modelo realize previsões abaixo do que realmente possa ocorrer.

Foram executadas rodadas de treinamento para cada um dos modelos separadamente, a fim de avaliar como os hiper parâmetros de taxa de aprendizado, iterações, neurônios ocultos, dropout, amsgrad e higher betas melhor se adaptavam a fim de produzir uma menor perda e maior acurácia. Os modelos foram construídos utilizando o otimizador de retro propagação Adam e sempre utilizando a mesma quantidade de neurônios ocultos nas camadas recorrentes e camada totalmente conectada. Outra característica marcante ao longo dos testes foi o uso de minilotes de tamanho 32, que levaram a maior velocidade no treinamento dos modelos, sem impactar negativamente os resultados obtidos.

O primeiro modelo a ser testado foi o LSTM para previsão futura de 3 horas. Foram feitos diversos testes sobre a quantidade de neurônios na camada oculta, quantidade de camadas recorrente, tamanho da janela de dados para analisar para cada predição, quantidade de iterações para treinamento, taxa de aprendizado e melhor taxa de *dropout*. A Tabela 1 apresenta os resultados obtidos a partir deste modelo.

**Tabela 1. Comparativo com 50 iterações para quantidade de neurônios ocultos.**

Iterações	Neurônios ocultos	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
50	32	58,8%	49,83%	0,07605	0,09184
50	64	61,13%	56,44%	0,05212	0,08817
50	128	70,62%	65,76%	0,03270	0,04846
50	256	77,81%	90,93%	0,02642	0,01651
50	512	53,66%	40,66%	0,03035	0,07068

Através da Tabela 1 é possível observar que os primeiros testes com 50 iterações para diversas quantidades de neurônios na camada oculta, apresentaram acurácia para treino e desenvolvimento, perda para treino e validação. Nessa fase não se salvou a taxa de aprendizado e *droupout*. Com os resultados, percebeu-se que 256 neurônios ocultos era a melhor opção. Na sequência dos testes foi estabelecida a melhor quantidade de iterações de treinamento, conforme pode ser visto na Tabela 2 que compara os resultados com relação a quantidade de iterações de treinamento, sendo que o processo com 200 iterações apresentou melhor resultado, tanto em relação a acurácia quanto a perda.

**Tabela 2. Comparativo para quantidade de iterações.**

Iterações	Neurônios ocultos	Dropout	Taxa aprendizado	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
100	256	10%	1e-05	70,89%	72,84%	0,02994	0,04206
200	256	10%	1e-05	89,22%	83,08%	0,01917	0,0324

Observou-se que mais iterações de treinamento melhoram o resultado. Nesse contexto, optou-se por aumentar o *dropout* para adicionar mais aleatoriedade e deixar generalista com relação aos dados de treino e validação. Outros testes realizados abordaram: aumentar da quantidade de neurônios e adicionar uma segunda camada LSTM com 400 iterações, conforme pode se observar na Tabela 3.

**Tabela 3. Comparativo final para o modelo LSTM para previsão de 3 horas.**

Tamanho sequência	Iterações	Camadas LSTM	Neurônios ocultos	Dropout	Taxa aprendizado	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
97	200	1	256	25%	1e-05	90,39%	84,74%	0,04148	0,03216
97	400	1	256	25%	1e-05	96,46%	87,97%	0,03555	0,02678
97	100	2	256	-	1e-05	75,25%	73,82%	0,02546	0,04022
97	200	1	512	25%	1e-05	84,99%	87,14%	0,02709	0,0195
97	100	2	512	-	1e-05	75,01%	84,28%	0,02416	0,0321
97	100	1	1024	-	1e-05	81,6%	80,44%	0,04001	0,02895
97	200	1	1024	25%	1e-05	69,23%	82,87%	0,0196	0,01687
97	100	1	2048	-	1e-05	66,55%	49,83%	0,0285	0,17194

O melhor resultado encontrado em termos de acurácia continuou sendo com 256 neurônios, no qual a utilização *dropout* em 25% ajudou a reduzir o sobre ajuste nos dados de treino. Conclui-se que aumentar mais a quantidade de neurônios e camadas LSTM não se mostrou uma alternativa muito viável e pelo alto tempo de treino requerido, optou-se por abandonar essa linha de estudo. A Tabela 4 demonstra os resultados para o treinamento do modelo LSTM de previsão para 6 horas.

**Tabela 4. Comparativo para o modelo LSTM para previsão de 6 horas.**

Tamanho sequência	Iterações	Neurônios ocultos	Dropout	Taxa aprendizado	Amsgrad	Highter Betas	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
97	200	256	10%	1e-05	falso	não	78,64%	63,43%	0,0247	0,0597
97	200	256	25%	1e-05	falso	não	85,47%	45,65%	0,04636	0,11896
97	200	256	25%	5e-05	falso	não	78,71%	63,24%	0,04493	0,05865
97	200	256	10%	1e-04	falso	não	81,16%	57,95%	0,02225	0,06575
97	200	256	20%	1e-04	falso	não	81,92%	68,01%	0,03262	0,06086
97	200	256	25%	1e-04	falso	não	74,28%	72,38%	0,0264	0,02566
97	200	256	25%	1e-04	verdadeiro	não	83,82%	71,8%	0,04318	0,02388
97	200	256	25%	1e-04	verdadeiro	sim	62,51%	66,01%	0,08851	0,05122
97	200	256	25%	1e-03	falso	não	61,01%	54,57%	0,04068	0,07186
97	200	512	25%	1e-04	falso	não	58,5%	65,61%	0,07016	0,05092
97	200	512	25%	1e-05	falso	não	80,29%	64,29%	0,03346	0,06818
109	200	256	25%	1e-04	verdadeiro	não	87,73%	76,25%	0,04373	0,02003

Iniciou-se os testes com hiper parâmetros semelhantes aos utilizados para o modelo de 3 horas, mas os resultados não foram satisfatórios. Como alternativa descobriu-se que aumentar o valor da taxa de aprendizado para  $1e-04$  trouxe melhores resultados. Também se descobriu que a manipulação de parâmetros do otimizador Adam podem ser um caminho alternativo, sendo que a utilização do amsgrad que fica desativado por padrão, trouxe melhores resultados quando ativo para o modelo com sequência temporal de 109 observações. O outro parâmetro que foi alterado foi a tupla de betas, sendo respectivamente os coeficientes utilizados para calcular as médias do gradiente e seu quadrado, para 0,99 e 0.9999, mas não houve melhora do resultado.

Conforme proposto por Reddi, Kale e Kuma [11], o parâmetro amsgrad do otimizador Adam busca resolver o problema da não convergência para solução ótima. Sua proposta é utilizar uma memória de longo termo dos gradientes passados, para solucionar essa falha de convergência. A Tabela 5 expõem os resultados para o modelo LSTM de previsão para 9 horas.

**Tabela 5. Comparativo para o modelo LSTM para previsão de 9 horas.**

Tamanho sequência	Iterações	Neurônios ocultos	Dropout	Taxa aprendizado	Amsgrad	Highter Betas	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
97	200	128	25%	$1e-04$	falso	não	70,48%	52,63%	0,07828	0,0991
97	200	128	25%	$1e-04$	verdadeiro	não	82,21%	58,2%	0,08013	0,10003
97	200	128	25%	$1e-04$	false	sim	74,82%	50,6%	0,10056	0,11961
97	200	128	25%	$1e-04$	verdadeiro	sim	82,46%	54,94%	0,0797	0,10042
97	200	256	25%	$1e-04$	verdadeiro	sim	84,71%	60,6%	0,0549	0,09467
97	200	512	25%	$1e-04$	verdadeiro	sim	77,68%	48,88%	0,03894	0,09361
121	200	256	25%	$1e-04$	verdadeiro	sim	81,65%	73,66%	0,05414	0,02619

Inicialmente foram utilizadas camadas com 128 neurônios e sequência de tamanho 97, que apresentam maior velocidade de treinamento para verificação da melhor configuração da arquitetura da rede neural. Optou-se por seguir com os testes para manipulação dos parâmetros do otimizador de retro propagação Adam e o melhor resultado alcançado nessa etapa foi para uso de sequência de tamanho 121, com 256 neurônios e taxa de aprendizado  $1e-04$ .

A partir do treinamento dos modelos LSTM, pode-se observar alguns comportamentos, (i) o tamanho da sequência ajuda conforme se aumenta a janela de tempo a ser prevista, (ii) a manipulação dos parâmetros da função de retro propagação também influencia em previsões mais longas. A taxa de 25% de *dropout* se mostrou eficaz para reduzir o sobre ajuste sobre os dados de treinamento, tornando os modelos mais generalistas.

Nos testes com modelo de RNR do tipo GRU, tem-se os resultados da previsão para 3 horas apresentados na Tabela 6. Pode-se observar ótimos resultados logo nos primeiros testes. Assim como no modelo LSTM de 3 horas, utilizou-se 97 para o tamanho da sequência e 256 neurônios ocultos. O uso do parâmetro amsgrad junto com o *dropout*, demonstrou ser um grande aliado para reduzir o sobre ajuste sobre os dados de treino. Além de alcançar bons resultados rapidamente, os modelos criados com GRU tiveram pouca variação entre cenários de treinamento executados, diferente dos modelos LSTM de 3 horas, que variava em seus testes.

**Tabela 6. Comparativo para o modelo GRU para previsão de 3 horas.**

Tamanho sequência	Iterações	Neurônios ocultos	Dropout	Taxa aprendizado	Amsgrad	Highter Betas	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
97	200	256	25%	$1e-05$	falso	não	92,59%	88,36%	0,04172	0,00714
97	200	256	25%	$1e-05$	verdadeiro	não	89,27%	90,96%	0,05625	0,00682

A Tabela 7 apresenta o resultado do modelo GRU para 6 horas. Assim como no modelo LSTM de 6 horas, uma sequência de 109 registros performou melhor que 97. Mas, diferente do modelo LSTM, utilizar o parâmetro amsgrad não teve a mesma eficiência. A utilização da taxa de aprendizado em  $1e-05$  trouxe resultados melhores do que  $1e-04$ .



**Tabela 7. Comparativo para o modelo GRU para previsão de 6 horas.**

Tamanho sequência	Iterações	Neurônios ocultos	Dropout	Taxa aprendizado	Amsgrad	Highter Betas	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
97	200	256	25%	1e-04	falso	não	72,49%	71,22%	0,04049	0,02574
97	200	256	25%	1e-04	verdadeiro	não	83,37%	71,66%	0,04434	0,02445
109	200	256	25%	1e-04	falso	não	81,54%	69,36%	0,04027	0,01949
109	200	256	25%	1e-05	verdadeiro	não	74,39%	80,75%	0,08288	0,01728
109	200	256	25%	1e-05	falso	não	81,5%	79,85%	0,0619	0,01457

Na Tabela 8 são apresentados os resultados do treinamento do modelo GRU para 9 horas. Novamente a taxa de aprendizado mais alta 1e-04, teve piores resultados e o melhor resultado foi com 1e-05, apesar de bons resultados também serem atingidos para taxa ainda mais baixa de 5e-06, principalmente com 512 neurônios. Percebeu-se que o uso do parâmetro amsgrad e maiores valores de betas do otimizador Adam, melhorou consideravelmente os resultados de desenvolvimento, mas acabaram comprometendo os resultados de treino. Novamente, os resultados das redes recorrentes GRU tiveram melhor estabilidade nos resultados do que as redes LSTM.

**Tabela 8. Comparativo para o modelo GRU para previsão de 9 horas.**

Tamanho sequência	Iterações	Neurônios ocultos	Dropout	Taxa aprendizado	Amsgrad	Highter Betas	Acurácia treino	Acurácia dev.	Perda treino	Perda dev.
109	200	256	25%	1e-04	falso	não	62,24%	58,43%	0,05063	0,0508
109	200	256	25%	1e-05	falso	não	68,57%	65,39%	0,08991	0,02885
109	200	256	25%	1e-05	falso	sim	73,32%	67,79%	0,0834	0,02233
109	200	256	25%	1e-05	verdadeiro	sim	70,62%	77,0%	0,09773	0,01898
121	200	256	25%	1e-05	verdadeiro	sim	71,17%	80,19%	0,09444	0,01595
121	400	256	25%	1e-05	verdadeiro	sim	78,04%	78,63%	0,07111	0,01918
121	400	256	25%	5e-06	verdadeiro	sim	72,01%	74,13%	0,09064	0,01811
121	200	512	25%	5e-06	verdadeiro	sim	68,1%	84,85%	0,08448	0,0174
121	400	512	25%	5e-06	verdadeiro	sim	75,65%	74,36%	0,05929	0,01949

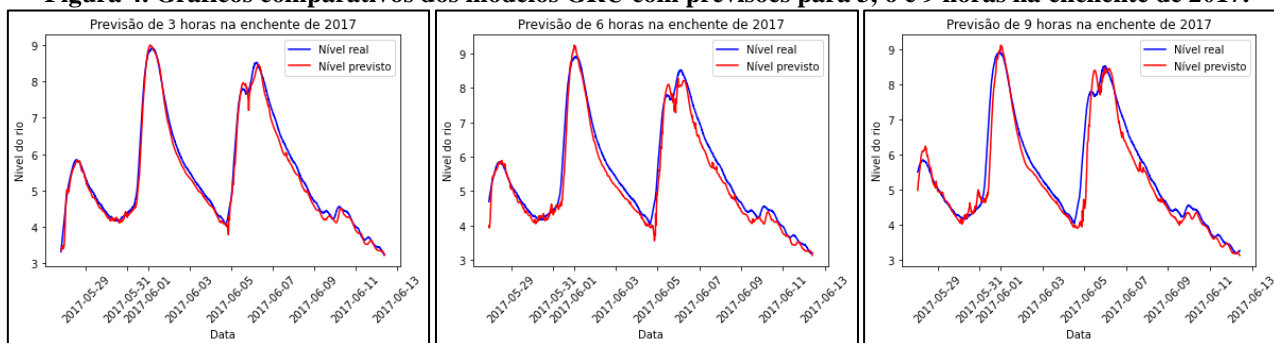
Na Tabela 9, pode-se observar os resultados comparativos entre os melhores modelos, avaliando a acurácia no conjunto de dados de teste, além das acurácias de treino e validação. Nela também pode-se observar que os resultados de testes nos modelos GRU são melhores que os modelos LSTM.

**Tabela 9. Comparativo entre os melhores modelos escolhidos.**

Modelo	Acurácia treino	Acurácia validação.	Acurácia teste
LSTM para 3 horas	96,46%	87,97%	62,42%
LSTM para 6 horas	87,73%	76,25%	24,08%
LSTM para 9 horas	81,65%	73,66%	28,45%
GRU para 3 horas	89,27%	90,96%	72,97%
GRU para 6 horas	81,5%	79,85%	47,75%
GRU para 9 horas	78,04%	78,63%	53,53%

A Figura 4 apresenta os valores preditos em relação aos reais em um evento do conjunto de dados de teste, referente a um evento de enchente ocorrido em 2017 em Blumenau. Pode-se observar, que o modelo proposto acaba sendo mais assertivo em previsões mais curtas de 3 horas, mas conseguiu capturar a tendência de alta ou baixa para o nível do rio em 6 e 9 horas.

**Figura 4. Gráficos comparativos dos modelos GRU com previsões para 3, 6 e 9 horas na enchente de 2017.**



## 5. Conclusões

Enchentes podem ser desastres muito danosos para a sociedade, porém podem ter seus impactos mitigados com avisos prévios da subida do rio para que apenas as áreas potencialmente atingidas sejam evacuadas. Dessa forma, podem ser evitadas percas de vidas humanas, sem necessidade de movimentação em massa de regiões que nem seriam atingidas. Assim a proposta deste trabalho era prever em curto prazo a evolução do nível do rio Itajaí-Açu em Blumenau durante situações de enchentes com o uso de redes neurais recorrentes. Mesmo com um conjunto de dados limitado, inconsistente e com grandes lacunas, foi possível estabelecer boas previsões. Para superar esses obstáculos, foram utilizadas técnicas de interpolação linear para preenchimento de pequenas lacunas e mediana para grandes lacunas no conjunto de dados. Houve uma boa assertividade nos dados de treino e desenvolvimento, mas percebeu-se através do conjunto de dados de teste que os modelos tanto do tipo LSTM quanto GRU ainda podem ser melhorados para melhor generalização das previsões realizadas. Foram atingidos percentuais de acurácia de aproximadamente 90% para previsões do nível do rio em 3 horas, além de cerca de 70% a 80% para as previsões de 6 e 9 horas. Tais valores para os eventos de desenvolvimento, as acurácias dos eventos de teste tiveram piora nos resultados para cerca de 60% a 70% nas previsões de 3 horas e de cerca de 24% a 53% para previsões de 6 a 9 horas futuras. Dados que reforçam as possíveis melhorias para generalização dos modelos para outros tipos de eventos de enchente, tanto para as arquiteturas utilizando redes neurais recorrentes do tipo LSTM quanto GRU. Entretanto, os modelos já conseguem atender uma demanda latente para servir como ferramentas auxiliares, aliados aos modelos matemáticos tradicionais. Apesar de apresentar em alguns momentos erro acima do tolerado, os modelos construídos se mostraram capazes de captar com sucesso a tendência de subida ou descida no rio com os dados disponíveis.

As RNR do tipo GRU tiveram maior efetividade do que as redes do tipo LSTM, principalmente num horizonte de previsão mais curto. Com uma base de treinamento maior e mais fiel à realidade, os modelos demonstram potencial para serem utilizados em situações reais de calamidade. Dessa forma, tem-se como extensão para este trabalho, além da utilização dos dados históricos da rede do CEOPS, montar um conjunto de dados aliados as estações registradas junto a Agência Nacional de Águas (ANA). Tendo uma maior quantidade de entradas e mais eventos de enchente para analisar, seria possível extrair melhor os padrões de subida e descida do rio para previsão pelos modelos.

Um outro grande desafio junto à quantidade de parâmetros de entrada e quantidade de eventos de cheias encontrados, está na metodologia adotada para preenchimento dos dados faltantes. Nesse trabalho, adotou-se uma abordagem linear, mas seria interessante avaliar um método não linear para preenchimento desses dados. Uma alternativa seria o cruzamento ou unificação dos dados de estações próximas, para suas informações se complementem.

## Referências

- [1] CARDOSO, M. **Mídia, desastres socioambientais e suas implicações para o desenvolvimento regional**: estudo de caso da enchente de 2011 no Vale do Itajaí a partir da cobertura realizada pelo Jornal de Santa Catarina no Twitter. 2014. 157 f. Dissertação (Mestrado em Desenvolvimento Regional) - Universidade Regional de Blumenau, Blumenau, 2014. Disponível em: [http://www.bc.furb.br/docs/DS/2014/357547\\_1\\_1.pdf](http://www.bc.furb.br/docs/DS/2014/357547_1_1.pdf). Acesso em: 11 out. 2021.
- [2] MARCUZZO, F. F. N. et al. Caracterização hidromorfológica da Bacia do Vale do Itajaí. In: SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS, 19. Maceió. **Anais....** Porto Alegre: ABRhidro, 2011. p. 1-20. Disponível em: [https://abrh.s3.sa-east-1.amazonaws.com/Sumarios/81/30b561de233dbb86de21e7bb16499a6a\\_773af1f15179fb62992a9c4d9ed18d57.pdf](https://abrh.s3.sa-east-1.amazonaws.com/Sumarios/81/30b561de233dbb86de21e7bb16499a6a_773af1f15179fb62992a9c4d9ed18d57.pdf). Acesso em: 11 out. 2021.
- [3] ALBERTON, G. B. et al. Aplicação de redes neurais artificiais para previsão de nível do rio Itajaí-Açú em Blumenau. **Revista Ibero-Americana de Ciências Ambientais**, v. 12, n. 4, p. 686-696, abr. 2021.
- [4] AGUDELO-OTALORA, L. M. et al. Comparison of physical models and artificial intelligence for prediction of flood levels. **Tecnol. cienc. agua**, Jiutepec, v. 9, n. 4, p. 209-235, Ago. 2018.
- [5] SILVA, A.; CASTRO, A.; VIEIRA, A. Modelo de previsão hidrológica utilizando redes neurais artificiais: um estudo de caso na bacia do Rio Xingu – Altamira-PA. **Revista Brasileira de Computação Aplicada**, v. 10, n. 3, p. 55-62, 23 out. 2018.
- [6] SOARES, D. G; TEIVE, R. C. G. Previsão de cheias do Rio Itajaí-Açú utilizando redes neurais artificiais. In: COMPUTER ON THE BEACH, 2015, Florianópolis. **Anais...** Florianópolis: UNIVALI, 2015. p. 308-317.
- [7] HÜSKEN, M.; STAGGE, P. Recurrent neural networks for time series classification. **Neurocomputing**, v. 50, p. 223-235, Jan. 2003.
- [8] BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, v. 5, n. 2, p. 157-166, Mar. 1994.
- [9] CHUNG, J. et al. **Empirical evaluation of gated recurrent neural networks on sequence modeling**. [S.l.]: arXiv:1412.3555, 2014.
- [10] CHO, K. et al. **On the properties of neural machine translation: encoder-decoder approaches**. [S.l.]: arXiv:1409.1259, 2014.
- [11] REDDI, S. J.; KALE, S; KUMAR, S. **On the convergence of adam and beyond**. [S.l.]: arXiv preprint arXiv:1904.09237, 2019.