

UNIVERSIDADE FEDERAL DO PARANÁ

PATRICK EDUARDO MARCHOR KARVAT

LOCALIZAÇÃO DE FALTAS EM REDES DE TRANSMISSÃO DE ENERGIA ELÉTRICA
UTILIZANDO AS REDES NEURAS RECORRENTES LSTM E GRU

CURITIBA PR

2022

PATRICK EDUARDO MARCHOR KARVAT

LOCALIZAÇÃO DE FALTAS EM REDES DE TRANSMISSÃO DE ENERGIA ELÉTRICA
UTILIZANDO AS REDES NEURAIIS RECORRENTES LSTM E GRU

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. Eduardo Cunha de Almeida.

Coorientador: Leandro Ensina.

CURITIBA PR

2022

*A todos que estiveram ao meu lado
durante minha jornada.*

AGRADECIMENTOS

Durante essa jornada muitas pessoas me ajudaram e foram fundamentais para que esse sonho se tornasse realidade. Primeiramente, agradeço aos meus orientadores, professor Eduardo Cunha de Almeida e ao Leandro Ensina por toda a ajuda e orientação fornecidas.

Agradeço aos meus pais, Rosalina Marchor Sant'Anna e Alcionir José Karvat, por me incentivarem e motivarem a nunca desistir. Às minhas irmãs, Dayane Grace Marchor Cupini e Jéssica Adrielly Marchor Karvat, por estarem sempre ao meu lado.

À minha namorada, Elisa Maria Chiarello, por estar ao meu lado a cada instante, me ajudando a ter forças para continuar em todas às vezes que pensei em desistir. Aos meus cunhados Elaine Cristina Chiarello e Juliano Marcelo de Oliveira, por estarem sempre dispostos a me ajudar, mesmo quando fui orgulhoso demais para pedir.

Por fim agradeço ao Bruce e a Mel pelo suporte emocional que foi fundamental nesse processo. E a todos que, direta ou indiretamente fizeram parte desse momento.

RESUMO

Esta pesquisa se propõe a desenvolver um modelo capaz de realizar a localização de faltas em uma linha de transmissão de um sistema elétrico de potência, a partir de dados de tensão e corrente. Para isso, foram utilizadas as redes neurais recorrentes *Long-Short Term Memory* e *Gated Recurrent Unit*, treinadas com dados obtidos através de simulações. Para o desenvolvimento deste modelo foi utilizada uma base de dados retirada do repositório público FADb, composto por simulações de uma rede de alta tensão baseada no sistema elétrico de nove barramentos do IEEE, geradas com a ferramenta ATPDraw. As redes neurais recorrentes utilizadas para este trabalho foram implementadas em Python por meio da biblioteca Keras. Além do Keras, outra ferramenta de apoio utilizada neste projeto foi o Jupyter, uma ferramenta de código aberto que permite editar e executar códigos em diversas linguagens a partir de um navegador web. O teste realizado utilizando o modelo *LSTM* treinado apresentou como resultado um erro médio absoluto de 0,11682 quilômetros, com desvio padrão de 0,51927 quilômetros, enquanto o modelo treinado da rede GRU atingiu um erro médio absoluto de 0,5 quilômetros, com desvio padrão de 1,187 quilômetros. Analisando os resultados obtidos através dos testes realizados, foi possível perceber que o modelo proposto é capaz de apontar a localização de faltas em linhas de transmissão de um sistema elétrico de potência com uma precisão considerável e que sua utilização prática pode agilizar o trabalho de encontrar falhas nesses sistemas.

Palavras-chave: Sistema Elétrico De Potência. Localização de Faltas. Rede Neural Recorrente.

ABSTRACT

This research sets out to develop a model capable of locating faults in a transmission line of an electric power system, based on voltage and current data. To do so, the Long-Short Term Memory and Gated Recurrent Unit recurrent neural networks were used, trained with data obtained through simulations. For the development of this model, the FADb database was used, composed of simulations of a high voltage network based on the IEEE nine-bus electrical system, generated with the ATPDraw tool. The recurrent neural networks used for this work were implemented in Python using the Keras library. In addition to Keras, another support tool used in this project was Jupyter, an open source tool that allows editing and execution of code in several programming languages straight from a web browser. The test performed using the trained LSTM model resulted in an average absolute error of 0.11682 kilometers, with a standard deviation of 0.51927 kilometers, while the trained GRU model reached an average absolute error of 0.5 kilometers, with a standard deviation of 1,187 kilometers. Analyzing the results obtained through the performed tests, it was possible to conclude that the proposed model is able to point out the location of faults in transmission lines of an electric power system with considerable precision and that its practical use can speed up the work of finding faults in such systems.

Keywords: Electric Power Systems. Fault location. Recurring Neural Network.

LISTA DE FIGURAS

2.1	Principais componentes de uma linha de transmissão	13
2.2	Adicionando conexões recorrentes a uma rede neural <i>feedforward</i>	15
2.3	Componentes internos de uma célula em uma Rede Neural Recorrente	16
2.4	Estrutura geral de uma célula LSTM..	17
2.5	Funcionamento de uma célula LSTM.	18
2.6	<i>Gated Recurrent Unit</i> e os portões que a compõe.	19
3.1	Exemplo de simulação.	20
3.2	Exemplo de simulação contendo dois ciclos pós falta de tensão e corrente respectivamente.	22
4.1	Gráfico das previsões do modelo LSTM	25
4.2	Gráfico das previsões do modelo GRU	25

LISTA DE TABELAS

3.1	Definição das camadas que compõe o modelo.	22
4.1	Comparativo entre os erros – em quilômetros – resultantes da aplicação da base de testes sobre os modelos desenvolvidos.	26
4.2	Comparativo entre os erros – em quilômetros – resultantes da aplicação da base de testes sobre os modelos desenvolvidos.	26

LISTA DE ACRÔNIMOS

IEEE	Institute of Electrical and Electronics Engineers
RNN	Redes Neurais Recorrentes
LSTM	Long-Short Term Memory
GRU	Gated Recurrent Unit
FADB	Fault Analysis Database
GPU	Graphics Processing Unit

LISTA DE SÍMBOLOS

Ω	Ohms, unidade de medida de resistência elétrica, representado pela letra grega ômega
GB	Gigabyte é uma unidade de medida de informação, segundo o Sistema Internacional de Unidades

SUMÁRIO

1	INTRODUÇÃO	12
1.1	PROPOSTA	12
1.2	ORGANIZAÇÃO DO TRABALHO	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	LINHAS DE TRANSMISSÃO	13
2.1.1	Faltas em Linhas de Transmissão	13
2.2	APRENDIZAGEM PROFUNDA	14
2.2.1	Rede Neural Recorrente	15
2.2.2	Considerações Finais	19
3	MÉTODO PROPOSTO	20
3.1	BASE DE DADOS	20
3.2	FERRAMENTAS	20
3.2.1	<i>Jupyter</i>	21
3.2.2	Keras	21
3.3	PROTOCOLO EXPERIMENTAL	21
3.3.1	Preparação dos dados	21
3.3.2	Modelos	22
3.4	CONSIDERAÇÕES FINAIS	23
4	RESULTADOS E DISCUSSÃO	24
5	CONCLUSÃO	28
	REFERÊNCIAS	29

1 INTRODUÇÃO

Neste capítulo estão dispostas as motivações e a proposta elaborada para o problema. Por fim, há uma exposição da organização dos demais capítulos que compõe este trabalho.

Conforme a presença de dispositivos eletrônicos cresce nos centros urbanos, aumenta também a quantidade de componentes necessários, para que os sistemas elétricos de potência funcionem corretamente e consigam atender à toda demanda gerada. Com isso, esses sistemas ficam cada vez mais complexos e extensos, além de se tornarem mais suscetíveis a defeitos.

Um Sistema Elétrico de Potência pode ser definido como o conjunto de instalações e equipamentos que possibilitam a geração, transmissão e distribuição de energia elétrica, responsável por assegurar que a energia gerada nas usinas alcance os centros urbanos e industriais de maneira segura e confiável (Blume, 2016).

As linhas de transmissão representam um papel fundamental nestes sistemas, garantindo que a energia gerada seja transportada aos diferentes pontos que compõe o sistema, como as subestações próximas ao usuário final. As linhas se estendem por grandes distâncias e, devido as suas características físicas, estão constantemente expostas à ação de fenômenos ambientais, como o contato com a chuva, animais e a vegetação. Deste modo, mesmo com a presença de sistemas de proteção robustos e eficazes, é impossível garantir um sistema completamente à prova de faltas (Kindermann, 1997).

Uma falta pode ser definida como um fluxo incomum de grandezas elétricas que ocorre nos componentes do sistema de potência, provocando grandes alterações na tensão e corrente do sistema, podendo ser consideradas simétricas ou assimétricas, dependendo das fases afetadas (Bichels, 2018).

A ocorrência de faltas em linhas de transmissão pode trazer várias consequências negativas, podendo impactar o fornecimento de energia de cidades inteiras, além de comprometer o funcionamento de instalações críticas, como hospitais. Portanto, é essencial a presença de sistemas que forneçam métodos para identificar, localizar e corrigir essas falhas de maneira a gerar o menor impacto possível ao usuário final.

1.1 PROPOSTA

Esse estudo se propõe a desenvolver um modelo capaz de identificar – com base de dados de tensão e corrente – a localização de faltas em linhas de transmissão de um sistema elétrico de potência. Para isso, serão utilizadas as redes neurais recorrentes Long-Short Term Memory e Gated Recurrent Unit, treinadas com dados obtidos através de simulações.

1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em 5 capítulos.

Neste capítulo é definida a motivação e a proposta de solução. No Capítulo 2 são apresentados os conceitos e definições fundamentais para este estudo. O capítulo 3 descreve como foi feita a construção dos modelos propostos, apresentando as ferramentas utilizadas para gerar estes modelos e descreve como foram executados os experimentos realizados. No Capítulo 4 estão dispostos os resultados obtidos nos experimentos para cada modelo gerado. O Capítulo 5 apresenta as conclusões obtidas neste trabalho e sugestões de trabalhos futuros relacionados a este estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão introduzidos os conceitos e as definições necessárias para compreender e justificar o problema de pesquisa proposto.

2.1 LINHAS DE TRANSMISSÃO

Sistemas elétricos de potência são usualmente divididos em três categorias: geração, transmissão e distribuição. O processo de geração é responsável pela transformação da energia mecânica em energia elétrica, a transmissão garante que a energia originada no processo de geração chegue aos centros urbanos e, por fim, a distribuição é responsável por receber a energia em altas tensões e convertê-la para baixa tensão, para que o usuário final possa utilizá-la (Bichels, 2018).

Para este trabalho, o enfoque será nas linhas de transmissão, cuja função principal é o transporte de energia dos centros de produção até os centros de consumo, como também a interligação de centros de produção e sistemas independentes (Fuchs, 1977). Na Figura 2.1 estão apresentados os principais componentes de uma linha de transmissão. Neste cenário, destacam-se os condutores de fase A, B e C que transportam a eletricidade, com os demais componentes servindo primordialmente como apoio a esses condutores.

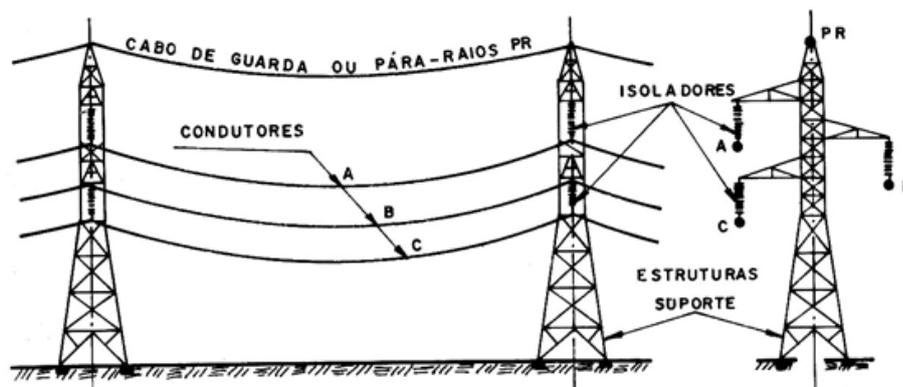


Figura 2.1: Principais componentes de uma linha de transmissão (Labegalini et al., 1977).

Devido à presença indispensável da energia elétrica nos centros urbanos, industriais e rurais, surge a necessidade do monitoramento e correção de falhas que ocorram nas linhas de transmissão, de modo a garantir a disponibilidade contínua e com qualidade de energia elétrica para os usuários. Para conseguir manter essas condições, diversos mecanismos de segurança, como relés de desligamento automático, são aplicados nas subestações para proteger o sistema elétrico contra falhas que podem ocorrer em qualquer ponto de sua extensão. Esses mecanismos de proteção são capazes de detectar situações desfavoráveis e anormais, para então isolar as áreas comprometidas do restante do sistema para evitar danos mais extensivos (Sato e Freitas, 2014).

2.1.1 Falhas em Linhas de Transmissão

Apesar de todos os cuidados tomados de modo a garantir o funcionamento contínuo do sistema elétrico, problemas nas linhas de transmissão são inevitáveis. Entre as adversidades encontradas estão problemas mecânicos, como os provocados pelo contato com vegetação,

problemas ocasionados pelo próprio equipamento, como a escolha inadequada de materiais em sua construção ou o desgaste natural destes materiais, e ambientais, como fortes ventos e descargas atmosféricas (Kindermann, 1997).

Uma falta pode ser definida como uma perturbação no estado normal de operação do sistema elétrico de potência, afetando as grandezas elétricas que o compõe (corrente, tensão, frequência e potências). Isso pode ocasionar restrição ou interrupção do funcionamento de um, ou mais equipamentos do sistema, criando assim uma demanda para que ações de correção sejam implementadas (Sato e Freitas, 2014).

As falhas podem ser de dois tipos: permanentes ou temporárias. O primeiro tipo requer algum meio de manutenção ou substituição dos componentes defeituosos do sistema de modo a retomar seu funcionamento regular, como a remoção de galhos que estejam em contato com a linha, ou a substituição de condutores, isoladores ou estruturas de suporte. Já o segundo tipo representa as faltas decorrentes de eventos que interrompem o funcionamento temporariamente, como sobretensão, umidade, chuva, contato temporário com a vegetação e raios.

A causa mais comum de falha temporária é a sobretensão, que ocorre quando o isolador é enfraquecido, ocasionando uma corrente de fuga que ioniza o ar, transformando a atmosfera em um condutor, gerando assim um arco elétrico que mantém o sistema em curto-circuito usando sua própria tensão. Para extinguir o arco a área afetada é isolada do resto do sistema e, sem a fonte de energia, o ar deixa de estar ionizado e o curto-circuito é interrompido. Então é efetuado o religamento e resume seu funcionamento normal (Kindermann, 1997).

As faltas por curto-circuito podem ser classificadas em (Kindermann, 1997):

- Monofásicos: Quando envolve somente uma das fases em contato com o aterramento, exemplos: AG¹, BG e CG;
- Bifásicos: Quando envolve duas fases, ou duas fases e o aterramento, exemplos: AC, ACG, AB, ABG, BC e BCG;
- Trifásicos: Quando envolve três fases, ou três fases e o aterramento, exemplos: ABC e ABCG.

As letras A, B e C representam cada uma das três fases da linha de transmissão, enquanto a letra G corresponde ao terra. Portanto, as iniciais AG indicam uma falta envolvendo a fase A e o terra, enquanto AB representa uma falta entre as fases A e B, sem a ação do aterramento. As falhas trifásicas são também conhecidas como falhas simétricas, por afetarem simultaneamente às três fases que compõem o sistema, enquanto as falhas monofásicas e bifásicas são denominadas falhas assimétricas (Yadav e Dash, 2014).

2.2 APRENDIZAGEM PROFUNDA

Aprendizagem profunda é um campo do aprendizado de máquina que estuda as redes neurais artificiais profundas, capazes de processar dados não estruturados, como textos e imagens, automatizando o processo de extração de características (Skansi, 2018).

Programas que utilizam aprendizagem profunda são compostos por uma hierarquia de algoritmos, que aplicam transformações não lineares sobre os dados de entrada e usam o que foi aprendido para criar um modelo estatístico de saída. Essas iterações se repetem até que seja obtida uma saída com um grau aceitável de precisão. O termo aprendizagem profunda se dá devido ao grande número de camadas de processamento pelos quais os dados são submetidos,

¹A, B e C são as fases do sistema, e G representa o aterramento.

sendo geralmente consideradas profundas quando possuindo 4 ou mais camadas (Burns e Brush, 2021).

No modelo tradicional de aprendizado de máquina, cabe ao especialista a tarefa de extrair características do problema, sendo que a eficácia depende diretamente de sua habilidade de definir os atributos relevantes. Na aprendizagem profunda, o próprio algoritmo é capaz de extrair e identificar quais características devem ser levadas em consideração (Burns e Brush, 2021).

2.2.1 Rede Neural Recorrente

As redes neurais são compostas por sequências de camadas interligadas, propagando as informações processadas em cada uma delas para as camadas no próximo nível. Essas redes são chamadas redes neurais *feedforward*. Esse modelo funciona muito bem para tratar dados uniformes, que podem ser representados em vetores de características. Entretanto, técnicas de aprendizado profundo não exigem que um vetor explícito de características seja fornecido ao algoritmo, o qual pode aprender a extrair os atributos mais significativos por si mesmo. Neste cenário, as Redes Neurais Recorrentes (RNN) possuem não apenas as conexões que propagam os dados de uma camada para outra, como também conexões capazes de realimentar uma camada com seus próprios resultados (Skansi, 2018).

Outro importante fator positivo de uma RNN é sua capacidade de atenuar o problema do *vanishing gradient*. Junto com os dados, cada camada possui um conjunto de pesos, que definem a maneira com a qual o algoritmo aprende os dados, e cada camada recebe os valores dos pesos da camada anterior e os atualiza. O problema do *vanishing gradient* se dá quando a atualização leva o peso a um valor tão pequeno que o algoritmo para de aprender, o que impossibilita a convergência desse algoritmo (Skansi, 2018).

Na Figura 2.2a está representada a arquitetura padrão de uma rede neural *feedforward*, em que I representa a camada em que a rede recebe os dados de entrada, H a camada escondida e O a camada em que a rede retorna o resultado encontrado. Na Figura 2.2b está apresentada a mesma rede, porém dessa vez ela é acrescida das conexões recorrentes, e a camada representada por H é dividida, demonstrando as n iterações pelas quais os dados passam nesta camada antes de chegar à camada de saída (Skansi, 2018).

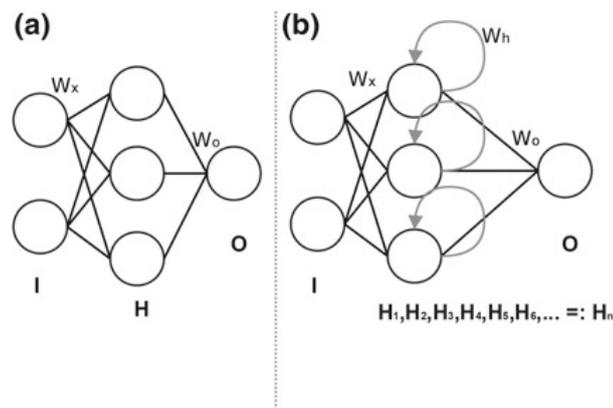


Figura 2.2: Adicionando conexões recorrentes a uma rede neural *feedforward* (Skansi, 2018).

Na Figura 2.3 é possível observar o funcionamento de uma célula RNN, evidenciando como é possível atingir, em uma única camada, um nível de processamento que demandaria uma rede neural *feedforward* com um número muito maior de camadas para atingir.

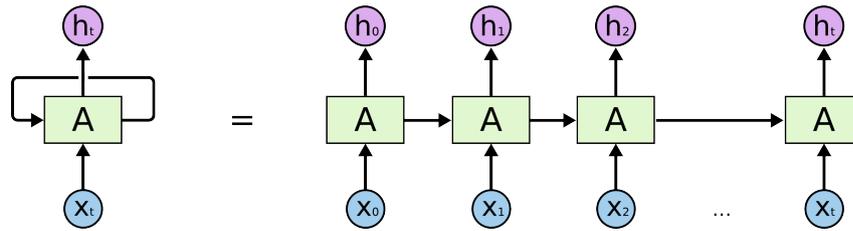


Figura 2.3: Componentes internos de uma célula em uma Rede Neural Recorrente (Olah et al., 2015).

Apesar de apresentarem atenuação para o problema do *vanishing gradient*, as redes neurais recorrentes ainda podem sofrer com ele, principalmente quando o aprendizado está sendo realizado sobre uma longa sequência, acarretando em RNNs com uma memória de curto prazo. Para resolver esse problema de memória de curto prazo, foram criadas as redes *Long Short-Term Memory* e *Gated Recurrent Unit* que possuem mecanismos internos chamados portões que regulam o fluxo de informação. Esses portões são capazes de aprender quais partes das informações são importantes e quais devem ser descartadas (Phi, 2018).

2.2.1.1 Long Short-Term Memory

A *Long Short-Term Memory* (LSTM) possui um fluxo similar a uma RNN convencional, processando dados ao propagar informações, sendo a principal diferença as operações efetuadas dentro da célula. Essas operações permitem que a LSTM opte por esquecer ou armazenar informações. O conceito principal de uma LSTM é a célula de estado, através da qual os dados relevantes são transportados por toda a estrutura da rede, funcionando como uma memória de longo prazo. Conforme essa célula de estado é transportada, informações são adicionadas ou removidas por meio dos portões. Os portões são estruturas capazes de aprender qual informação é relevante (Phi, 2018).

Na Figura 2.4 é possível observar com mais detalhes uma célula LSTM e os portões que operam dentro dela. Os portões sigmoide são responsáveis por normalizar os dados do vetor no intervalo $[0, 1]$, importantes para garantir que o algoritmo esqueça informações, devido à multiplicação dos dados por valores próximos a zero. O portão de tangente é responsável por garantir que os dados que estão sendo propagados permaneçam no intervalo $[-1, 1]$, para impedir um crescimento descontrolado decorrido das seguidas transformações que a rede faz sobre ele (Phi, 2018).

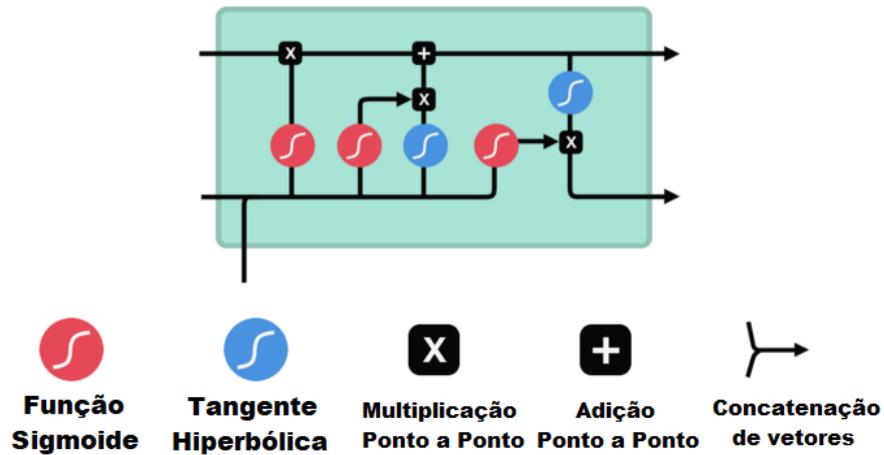


Figura 2.4: Estrutura geral de uma célula LSTM (Phi, 2018).

Na Figura 2.5a está representado como é calculado o vetor da célula de estado, por meio da soma e da multiplicação de seu valor anterior com o resultado dos portões da célula atual. Na Figura 2.5b está demonstrado o funcionamento do primeiro portão de esquecimento (sigmoide), que opera sobre os dados brutos da entrada somados ponto a ponto dos valores do vetor resultante do processamento da camada anterior, decidindo quais valores devem ser propagados, e quais devem ser esquecidos. A Figura 2.5c está demonstrado o portão de entrada, responsável por atualizar a célula de estado com novas informações que julgar relevantes para o problema. O portão de entrada opera sobre o mesmo vetor resultante utilizado pelo primeiro portão de esquecimento, porém utilizando um portão de esquecimento em conjunto com um portão tangencial. Por fim, na Figura 2.5d está representada a maneira como os dados da célula de estado são utilizados para gerar o resultado passado para a próxima camada (Skansi, 2018).

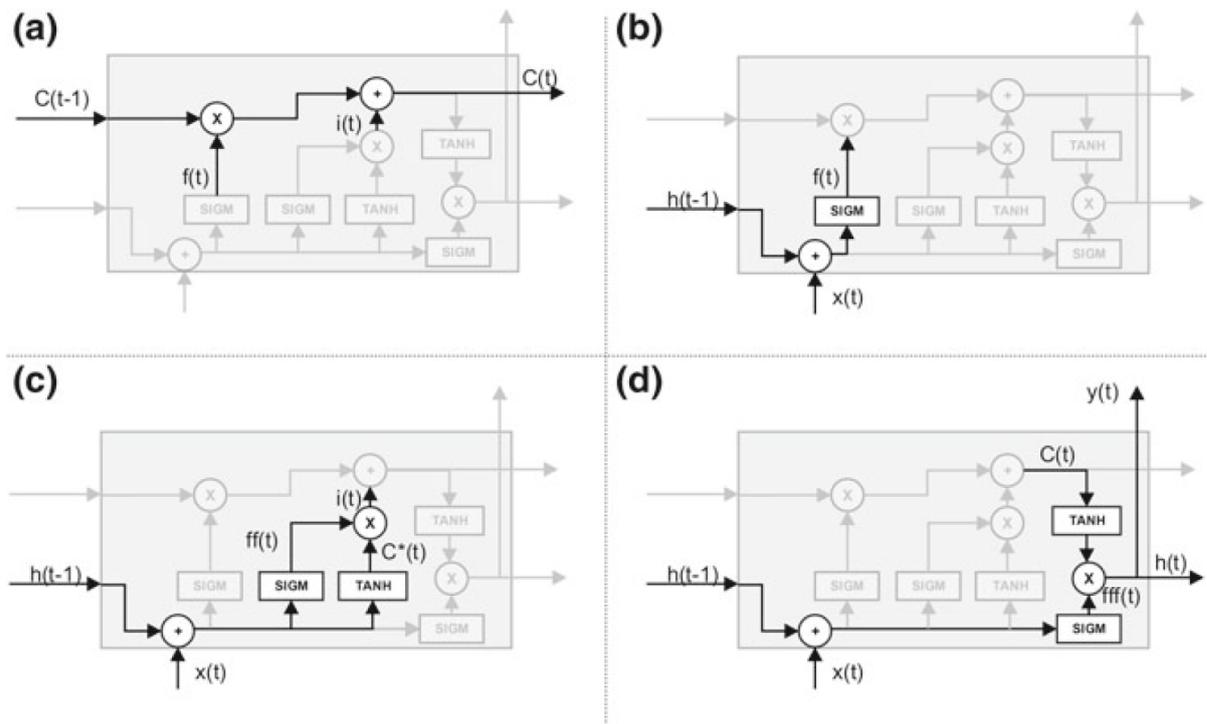


Figura 2.5: Célula de estado (a), portão de esquecimento (b), portão de entrada (c) e portão de saída (d) (Skansi, 2018).

2.2.1.2 Gated Recurrent Unit

A rede *Gated Recurrent Unit* (GRU) funciona de maneira similar a LSTM, sem o emprego de vetores de estado individuais, utilizando o próprio *hidden state* (vetor de dados que recebe da camada anterior) para transferir informações. GRUs possuem somente dois portões, o portão de redefinição e o portão de atualização (Phi, 2018).

Na Figura 2.6 estão destacados os portões que compõe uma célula GRU. O portão de atualização (*update gate*) funciona da mesma maneira que os portões de esquecimento e entrada da LSTM, decidindo quais dados são importantes, e devem ser mantidos, além daqueles que devem ser descartados. Enquanto o portão de redefinição (*reset gate*) determina o quanto das informações do estado anterior serão esquecidas (Phi, 2018).

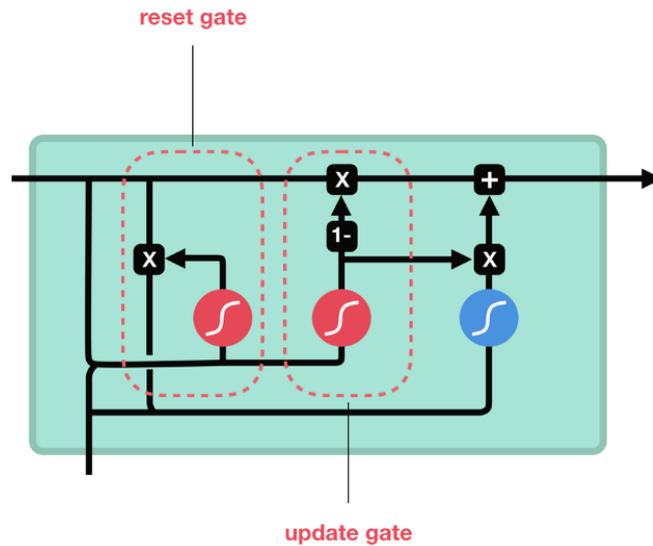


Figura 2.6: *Gated Recurrent Unit* e os portões que a compõe (Phi, 2018).

2.2.2 Considerações Finais

Existem distintas maneiras de implementar a localização de uma falta em um sistema de transmissão de energia elétrica. Neste cenário, os métodos baseados em aprendizado de máquina têm ganhado maior atenção nos últimos anos, principalmente devido a sua adaptabilidade e bom desempenho.

Devido a natureza dos dados abordados por este trabalho, consistindo em longas sequências de dados que, ao serem analisadas individualmente, podem apresentar poucos indícios a respeito da localização de faltas, propõe-se a utilização das redes neurais recorrentes LSTM e GRU, por serem capazes de analisar sequências de tamanho indefinido sem perder dados relevantes. No próximo capítulo será descrito o método proposto para este trabalho.

3 MÉTODO PROPOSTO

Neste capítulo está descrita a base de dados com simulações de falta utilizada neste trabalho, bem como as principais ferramentas empregadas e o método proposto para a localização de faltas em linhas de transmissão.

3.1 BASE DE DADOS

Os dados utilizados neste trabalho são provenientes do repositório público Fault Analysis Database (FADb)¹, composto por simulações de uma rede de alta tensão baseada no sistema elétrico de nove barramentos do IEEE, modelada a partir da ferramenta ATPDraw. A rede utiliza como base os parâmetros da mais longa linha de transmissão da Companhia Paranaense de Energia (Copel), possuindo 414 km de extensão, 500 kV de tensão e frequência de 60 Hz.

A base é constituída por 168.000 simulações a uma taxa de amostragem de 10 kHz, resultando – com base na frequência de operação – em 167 amostras por ciclo. Estão presentes simulações de faltas desde o quilômetro 4,14 (1% da linha) até o quilômetro 414 (100% da extensão da linha), com resistências que variam de 0,01 Ω à 200 Ω em intervalos de 10 Ω .

A base está dividida em dez conjuntos de 16.800 simulações, representando as categorias de falta descritos na Seção 2.1.1.

Na base, cada simulação compreende os dados de corrente e tensão de ambos os terminais (extremidades) da linha de transmissão.

Na Figura 3.1 pode ser observado um trecho de uma simulação do tipo ABG, em que as colunas destacadas representam os dados de tensão e corrente para um único terminal. Em vermelho estão destacados os dados de tensão para as fases A, B e C respectivamente, enquanto as colunas destacadas em verde representam os dados de corrente para as mesmas fases.

v:X0014A	v:X0014B	v:X0014C	v:X0013A	v:X0013B	v:X0013C	c:X0013A:BUS8A	c:X0013B:BUS8B	c:X0013C:BUS8C	c:BUS9A:X0014A	c:BUS9B:X0014B	c:BUS9C:X0014C
368730,72	17398,168	-386128,88	382069,7	-7397,7866	-374671,9	-254,98827	316,58102	-61,59275	86,56792	-353,36502	266,79712
359688,88	33809,97	-393498,84	373807,2	9072,551	-382879,75	-263,0354	312,1483	-49,112877	100,00134	-357,03653	257,0352
350135,9	50173,797	-400309,72	365013,6	25530,016	-390543,6	-270,70898	307,27222	-36,563248	113,29296	-360,20114	246,90817
340085,44	66466,42	-406551,84	355701,38	41951,242	-397652,62	-277,99826	301,96	-23,961725	126,42387	-362,85425	236,43039
329551,66	82664,7	-412216,38	345883,78	58312,914	-404196,7	-284,8931	296,2193	-11,326216	139,3754	-364,99216	225,61676
318549,6	98745,65	-417295,25	335574,72	74591,805	-410166,5	-291,38367	290,05832	1,3253492	152,1291	-366,6117	214,48259
307094,84	114686,41	-421781,25	324788,78	90764,83	-415553,62	-297,4608	283,48578	13,975023	164,66689	-367,71057	203,04369
295203,66	130464,35	-425668	313541,3	106809,016	-420350,3	-303,11584	276,51096	26,604868	176,9709	-368,28717	191,31627
282892,94	146057,05	-428950	301848,25	122701,59	-424549,84	-308,34064	269,14365	39,196995	189,02368	-368,3407	179,31703
270180,2	161442,38	-431622,56	289726,2	138419,98	-428146,2	-313,1276	261,394	51,73357	200,80807	-367,871	167,06294
257083,42	176598,47	-433681,88	277192,34	153941,89	-431134,22	-317,46976	253,27293	64,196815	212,30736	-366,8788	154,57146
243621,28	191503,78	-435125,06	264264,53	169245,22	-433509,75	-321,3608	244,79176	76,56905	223,50517	-365,36542	141,86026
229812,86	206137,17	-435590,03	250961,1	184308,22	-435269,3	-324,7951	235,96239	88,83271	234,38565	-363,33304	128,9474
215677,77	220477,83	-436155,6	237300,95	199109,5	-436410,47	-327,76767	226,7973	100,97037	244,93329	-360,7845	115,851204

Figura 3.1: Exemplo de simulação. Em verde estão destacados os dados de corrente e em vermelho os dados de tensão para um único barramento.

3.2 FERRAMENTAS

Nesta seção estão descritas as ferramentas de apoio empregadas para o desenvolvimento deste trabalho.

¹<https://ldrv.ms/u/s!ArMEeMx4MYDNimHVxiDx3b4CI3iL?e=8GfXg7>

3.2.1 Jupyter

Jupyter é uma ferramenta de código aberto, que permite editar e executar códigos em diversas linguagens de programação a partir de um navegador *web*. Os cadernos do Jupyter podem ser armazenados localmente, ou em repositórios *online*, nos quais é possível executar código e acessar dados remotamente (Randles et al., 2017). Neste projeto, os notebooks foram utilizados com a linguagem Python, acessando os dados citados na seção 3.1 armazenados no Google Drive.

3.2.2 Keras

As redes neurais recorrentes utilizadas para este trabalho foram implementadas em *Python* 3.7.13 por meio da biblioteca *Keras* na versão 2.8.0.

Keras é uma biblioteca de alto nível em Python que permite ao desenvolvedor o foco nos conceitos principais de aprendizagem profunda, deixando para a biblioteca os detalhes de baixo nível da arquitetura, como tensores e equações matemáticas. Por meio desta biblioteca são definidos apenas os detalhes fundamentais para o problema a ser trabalhado, como a quantidade de camadas necessárias, neurônios por camada e funções de ativação (Manaswi, 2018).

3.3 PROTOCOLO EXPERIMENTAL

O protocolo experimental foi definido como a etapa do projeto em que são realizadas manipulações nos dados de modo a prepará-los para o treinamento do modelo, bem como a definição dos modelos utilizados.

3.3.1 Preparação dos dados

A tarefa de localização de falta do método proposto emprega os dados de dois ciclos pós-falta (de corrente e tensão) para um único terminal da linha de transmissão onde ocorreu uma falta.

Os dados foram padronizados utilizando a técnica Z-Score, que consiste em encontrar a razão entre a diferença de cada valor para a média e o desvio padrão, conforme pode ser observado na equação 3.1, em que X_i representa o valor a ser padronizado, \bar{X} representa a média e S o desvio padrão, resultando em um conjunto de dados com zero de média e um de desvio padrão (Weinberg e Abramowitz, 2008).

$$Z_i = \frac{X_i - \bar{X}}{S} \quad (3.1)$$

Na Figura 3.2 é possível observar um exemplo de simulação para uma falta do tipo AG. Na seção destacada é possível observar os impactos gerados por uma falta nos dados de corrente e tensão, respectivamente, referentes a dois ciclos após o instante de falta.

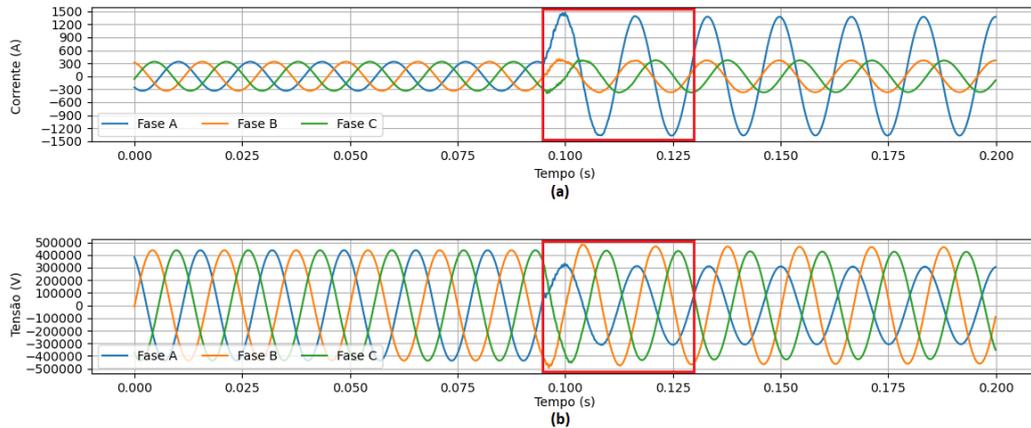


Figura 3.2: Exemplo de simulação contendo dois ciclos pós falta de tensão e corrente respectivamente.

3.3.2 Modelos

A definição da arquitetura dos modelos adotada nesse trabalho foi realizada com base em experimentos empíricos. Inicialmente foram criadas arquiteturas com um número reduzido de neurônios e camadas, que foram gradualmente incrementadas visando otimizar as métricas utilizadas para o treinamento. Os melhores resultados foram obtidos com a implementação da arquitetura disposta na Tabela 3.1, composta por três camadas, cada uma com 1.000 neurônios, além da camada de saída, que possui um neurônio, cuja saída é representada por um valor entre zero e um, que ao ser multiplicado pela extensão da linha indica a localização, em quilômetros, do local de ocorrência da falta.

Os modelos foram compilados utilizando o otimizador Adam (Kingma e Ba, 2015) e a métrica utilizada foi *Mean Squared Error*, para o treinamento foi utilizado um *batch_size* de 1.680. A rede *LSTM* foi treinada por 5.569 épocas, atingindo um erro médio quadrático de $6,2620 \cdot 10^{-7}$ nos dados de validação, enquanto a *GRU* precisou ser treinada por 1.800 épocas, atingindo um erro médio quadrático de $4,6111 \cdot 10^{-06}$. A entrada dos modelos é uma matriz de 6 colunas, cujo conteúdo são os valores para as fases A, B e C de corrente e tensão respectivamente, conforme é possível observar na coluna *input_shape* da tabela 3.1. O treinamento foi realizado em um ambiente Linux Ubuntu 18.04.5 com 25 GB de memória RAM, utilizando uma GPU NVIDIA Tesla P100 com 16 GB de memória VRAM.

Tabela 3.1: Definição das camadas que compõe o modelo.

	units	return_sequences	input_shape
LSTM	1000	True	(6, 334)
LSTM	1000	True	—
LSTM	1000	—	—
Dense	1	—	—

3.3.2.1 Separação dos dados

Antes de serem padronizados, os dados descritos na Seção 3.3.1 foram particionados em três conjuntos. O primeiro, compreendendo 60% dos dados (equivalente a 100.800 amostras), constitui a base de treinamento. Dos 40% restantes, 20% foram utilizados como base de validação (totalizando 33.600 amostras), com a finalidade de avaliar o progresso do treinamento do modelo e estabelecer as arquiteturas mais adequadas dos algoritmos para a localização de falha, e os 20%

restantes foram utilizados após o treinamento, tendo a função de representar o comportamento do modelo com uma base de dados com a qual nunca teve contato.

3.4 CONSIDERAÇÕES FINAIS

Neste capítulo foi definido o método proposto para a localização de faltas em linhas de transmissão. Especificou-se a origem e construção da base de treinamento e como ela foi tratada para obter os dados utilizados nos experimentos, além de serem introduzidas as ferramentas através das quais foram realizados os experimentos, cujos resultados estão expostos no Capítulo 4.

4 RESULTADOS E DISCUSSÃO

Nesse capítulo estão descritos os resultados obtidos nos experimentos realizados utilizando as ferramentas apresentadas no Capítulo 3.

Neste trabalho foi proposto um método para a localização de faltas em redes de transmissão de energia elétrica utilizando as redes neurais recorrentes *LSTM* e *GRU*, treinadas utilizando a base de dados FADb. A escolha das redes neurais recorrentes se dá devido à capacidade de tais redes de extrair características de dados que tem seu comportamento afetado com a passagem de tempo, como é o caso dos sinais de tensão e corrente presentes na base extraídos de linha de transmissão de Sistemas Elétricos de Potência.

Os dados foram separados em três conjuntos sem intersecção – treino, teste e validação – e padronizados utilizando Z-Score, cuja finalidade é garantir que os dados utilizados estejam em uma mesma ordem de grandeza, evitando assim possíveis enviesamentos que possam ser causados pela presença de valores muito altos. A padronização dos dados foi realizada após a separação para evitar *Data Leakage*, ou seja, para evitar que características dos dados de teste influenciem os valores de treinamento (Brownlee, 2020).

As arquiteturas utilizadas nos modelos gerados por este trabalho foram alcançadas por experimentos empíricos. Inicialmente foram utilizados modelos com um número reduzido de camadas e neurônios que foram incrementados gradualmente, buscando minimizar o erro obtido durante o treinamento.

O parâmetro *batch_size* também precisou ser incrementado gradualmente, devido a sua influência na precisão com a qual o modelo realiza as suas previsões (Radiuk, 2017). Nos testes iniciais, os dados de tensão e corrente eram armazenados em um único vetor com 2.004 valores (334 valores para cada fase de corrente e tensão, dados referentes a dois ciclos após a incidência de falta), porém com o aumento do parâmetro *batch_size*, foi necessário realizar uma alteração no formato dos dados, de modo a alcançar uma melhor utilização da memória RAM da GPU. Portanto, foi empregado o formato dos dados apresentado na Tabela 3.1, que consiste em uma matriz com 6 colunas, cada uma representando os valores para uma das fases de tensão e corrente respectivamente, similar à disposição dos dados exibidos na Figura 3.1.

Inicialmente os modelos foram treinados por 10.000 épocas, de modo a encontrar a quantidade necessária para evitar *overfitting* – ponto onde o modelo se torna especialista em prever dados da base de treinamento, tornando-se ineficaz na tarefa de prever dados desconhecidos (IBM, 2001).

Utilizando os modelos treinados para prever os dados de teste, foi obtida, para a rede LSTM, a distribuição de dados disponível na Figura 4.1, atingindo um erro médio absoluto de 0,11682 quilômetros (equivalente à 0,02822% da extensão da linha), com desvio padrão de 0,51927 quilômetros (0,12543% da extensão da linha). As simulações utilizadas para o modelo compreendem o intervalo [4, 14, 414] quilômetros, em parcelas de 4,14 quilômetros, portanto, as previsões do modelo que excediam esses valores foram alteradas, de modo a manter as previsões no intervalo considerado.

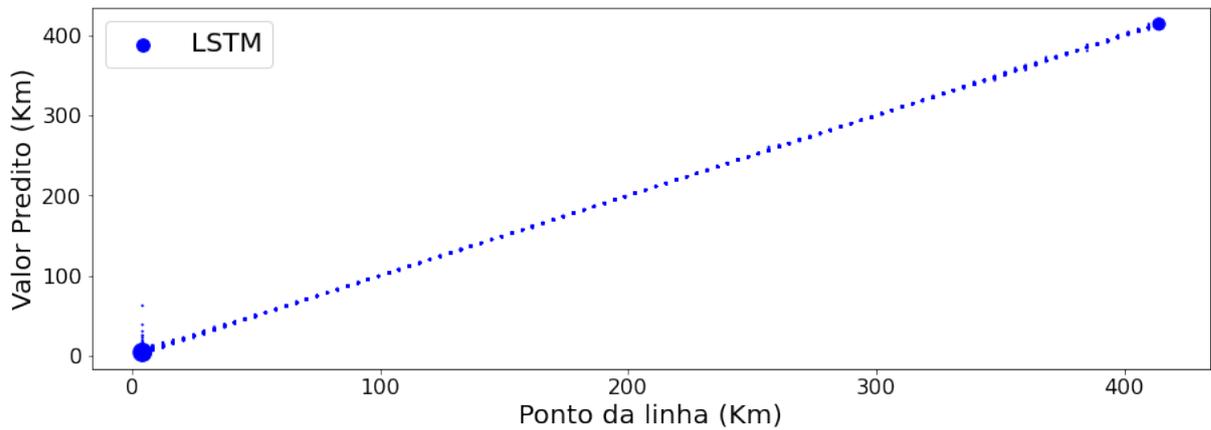


Figura 4.1: Gráfico das previsões do modelo da rede LSTM. O eixo x corresponde aos valores verdadeiros em quilômetros, e o eixo y corresponde aos valores retornados pelo modelo.

O modelo treinado para a rede GRU foi utilizado para prever a base de teste, alcançando os resultados observáveis na Figura 4.2, atingindo um erro médio absoluto de 0,5 quilômetros (0,12077% da extensão da linha), com desvio padrão de 1,187 quilômetros (0,28671%). Assim como na seção anterior, foram alteradas as previsões com valores fora do intervalo considerado para este trabalho.

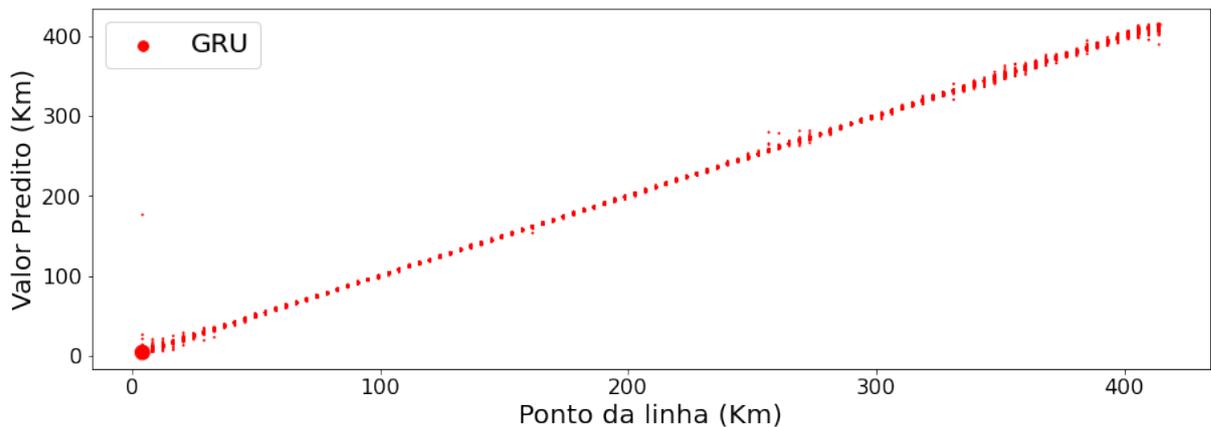


Figura 4.2: Gráfico das previsões do modelo da rede GRU. O eixo x corresponde aos valores verdadeiros em quilômetros, e o eixo y corresponde aos valores retornados pelo modelo.

Na tabela 4.1 estão dispostos os resultados parciais obtidos para cada modelo. Para esta análise, foram consideradas partições de dez por cento da extensão do barramento. Pode-se observar que a incidência maior de erros está presente nas seções próximas às extremidades da linha de transmissão. Esse comportamento foi observado na literatura (Ray e Mishra, 2016), porém ainda não foram encontrados resultados conclusivos a respeito de suas causas.

Tabela 4.1: Comparativo entre os erros – em quilômetros – resultantes da aplicação da base de testes sobre os modelos desenvolvidos.

Quilômetros		Erro médio		Erro máximo	
Início	fim	LSTM	GRU	LSTM	GRU
4,14	41,4	0,27885 ± 1,56277	0,65098 ± 3,16592	58,32434	172,27525
41,4	82,8	0,08128 ± 0,11632	0,35513 ± 0,33339	2,19071	3,95642
82,8	124,2	0,06447 ± 0,05771	0,30949 ± 0,27200	0,77581	3,35649
124,2	165,6	0,06887 ± 0,07118	0,34368 ± 0,32450	2,25345	7,92911
165,6	207,0	0,06478 ± 0,05501	0,34771 ± 0,30650	0,47009	3,31541
207,0	248,4	0,06825 ± 0,06294	0,34384 ± 0,33307	1,13448	4,45863
248,4	289,8	0,08777 ± 0,13133	0,43738 ± 0,74441	4,03100	22,93373
289,8	331,2	0,10842 ± 0,12348	0,48179 ± 0,54479	1,50352	10,69869
331,2	372,6	0,17350 ± 0,26758	0,71946 ± 0,88032	4,11628	10,96010
372,6	414,0	0,15515 ± 0,30287	1,00771 ± 1,29963	5,95679	24,72454

Visando mitigar o problema das predições nas extremidades, foram removidas do conjunto de predições os *outliers* – elementos que apresentam um valor destoante dos demais, de modo a levantar suspeitas a respeito de sua validade como parte do conjunto. Para tal foram calculados os valores dos quartis – valor abaixo do qual está determinada porção dos dados – e calculada a distância interquartil. Com base nessa distância, foi estipulado um limiar equivalente à 150% de seu valor e, todas as predições menores que o primeiro quartil (valor abaixo do qual estão dispostos 25% dos dados), ou maiores que o terceiro (limite abaixo do qual estão dispostos 75% dos dados) que extrapolavam este limiar foram desconsideradas dos resultados (Salles, 2018).

Após a remoção dos *outliers* restaram 31.675 registros na base de testes da rede *LSTM* (94,27% do tamanho original) e 32.162 (95,72%) para a rede *GRU*. Na tabela 4.2 estão dispostos os dados obtidos após a limpeza da base.

Tabela 4.2: Comparativo entre os erros – em quilômetros – resultantes da aplicação da base de testes sobre os modelos desenvolvidos.

Quilômetros		Erro médio		Erro máximo	
Início	fim	LSTM	GRU	LSTM	GRU
4,14	41,4	0,07995 ± 0,07266	0,39829 ± 0,36183	0,40148	2,80647
41,4	82,8	0,06149 ± 0,04697	0,31778 ± 0,24324	0,23826	1,36249
82,8	124,2	0,05927 ± 0,04518	0,28787 ± 0,21538	0,22757	1,10254
124,2	165,6	0,06114 ± 0,04670	0,31458 ± 0,24385	0,23786	2,00675
165,6	207,0	0,05914 ± 0,04516	0,31923 ± 0,23507	0,22451	1,19116
207,0	248,4	0,06145 ± 0,04784	0,30575 ± 0,23457	0,23551	1,39250
248,4	289,8	0,07171 ± 0,05553	0,34889 ± 0,27077	0,27623	1,33190
289,8	331,2	0,08627 ± 0,06800	0,39404 ± 0,31114	0,35621	1,52760
331,2	372,6	0,11324 ± 0,09309	0,53960 ± 0,43621	0,46504	2,23142
372,6	414,0	0,10293 ± 0,10201	0,80552 ± 0,80595	0,83621	5,54864

Considerando os resultados apresentados, pode-se observar que o modelo gerado com a rede neural recorrente *LSTM* apresenta uma melhor solução para o problema proposto neste trabalho.

Devido à incidência de algumas predições equivocadas – especialmente nas extremidades da linha – observa-se a possibilidade aplicar metodologias similares em arquiteturas divergentes dos modelos propostos, utilizando alguma métrica estatística, como média ponderada ou mediana, de modo a mitigar os problemas encontrados.

5 CONCLUSÃO

Neste trabalho foi apresentada uma proposta de solução para o problema de localização de faltas em redes de transmissão de energia elétrica, utilizando redes neurais recorrentes treinadas com dados de dois ciclos pós falta de tensão e corrente.

Conforme pode ser observado nos resultados dos experimentos, os treinamentos realizados sobre a base de dados utilizada atingem bons resultados em sua função de localizar faltas, destacando-se o modelo LSTM, por alcançar uma melhor precisão em suas predições. Contudo, foram detectados alguns casos que comprometem a assertividade do algoritmo, em especial, quando são analisados os resultados obtidos nas seções próximas às extremidades das linhas de transmissão.

Considerando os resultados obtidos a partir dos experimentos deste trabalho, sugerem-se como trabalhos futuros:

1. Verificar a capacidade de generalização do modelo gerado, utilizando dados de outras redes de transmissão, ou dados reais da rede sobre a qual as simulações foram modeladas.
2. Desenvolver novos modelos, utilizando diferentes métricas e arquiteturas, e utilizar a mediana ou média entre as predições deste conjunto de modelos, podendo retornar não só o valor da predição, como também a margem de erro com base nos resultados obtidos, de modo a fornecer uma informação mais completa a respeito da precisão da predição.

REFERÊNCIAS

- Bichels, A. (2018). *Sistemas Eletricos de Potência - Métodos de Análise e Solução*. EDUTFPR.
- Blume, S. (2016). *Electric Power System Basics for the Nonelectrical Professional*. IEEE Press Series on Power and Energy Systems. Wiley.
- Brownlee, J. (2020). *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*, páginas 25 – 36. Machine Learning Mastery.
- Burns, E. e Brush, K. (2021). What is deep learning and how does it work? <https://searchenterpriseai.techtarget.com/definition/deep-learning-deep-neural-network>. Acessado em 09/12/2021.
- Fuchs, R. D. (1977). *Transmissão de Energia Eletrica Linhas Aereas*. Livros Tecnicos e Cientificos.
- IBM, C. E. (2001). What is overfitting? <https://www.ibm.com/cloud/learn/overfitting>. Acessado em 04/17/2022.
- Kindermann, G. (1997). *Curto-Circuito*. SAGRA LUZZATTO.
- Kingma, D. P. e Ba, J. L. (2015). Adam: A method for stochastic optimization. Em *International Conference on Learning Representations*, San Diego - United States of America.
- Labegalini, P. R., Labegalini, J. A., Fuchs, R. D. e Almeida, M. T. d. (1977). *Transmissão de Energia Eletrica Linhas Aereas*. Livros Tecnicos e Cientificos.
- Manaswi, N. K. (2018). *Deep Learning with Applications Using Python : Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras*, páginas 44–56. Apress.
- Olah, C., Gers, F., Cummins, F., Fernandez, S., Bayer, J., Wierstra, D., Togelius, J., Gomez, F., Gagliolo, M. e Graves, A. (2015). Understanding lstm networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acessado em 13/12/2021.
- Phi, M. (2018). Illustrated guide to lstm's and gru's: A step by step explanation. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>. Acessado em 10/12/2021.
- Radiuk, P. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Information Technology and Management Science*, 20:20–24.
- Randles, B. M., Pasquetto, I. V., Golshan, M. S. e Borgman, C. L. (2017). Using the jupyter notebook as a tool for open science: An empirical study. Em *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*.
- Ray, P. e Mishra, D. P. (2016). Support vector machine based fault classification and location of a long transmission line. *Engineering Science and Technology, an International Journal*, 19(3):1368–1380.

- Salles, R. (2018). Outlier: o ponto fora da curva. <https://medium.com/ensina-ai/outlier-o-ponto-fora-da-curva-1f28f3d9c23>. Acessado em 24/04/2022.
- Sato, F. e Freitas, W. (2014). *Análise de Curto- circuito e Princípios de Proteção em Sistemas de Energia Elétrica: Fundamentos e Prática*. GEN LTC.
- Skansi, S. (2018). *Introduction to Deep Learning From Logical Calculus to Artificial Intelligence*. Springer.
- Weinberg, S. e Abramowitz, S. (2008). *Statistics Using SPSS: An Integrative Approach*, página 98. Cambridge University Press.
- Yadav, A. e Dash, Y. (2014). An overview of transmission line protection by artificial neural network: Fault detection, fault classification, fault location, and fault direction discrimination. *Advances in Artificial Neural Systems*, 2014.