

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



APLICAÇÃO DE REDES NEURAS RECORRENTES E *SELF-ORGANIZING MAPS* EM
DADOS REAIS DE OPERADOR DE TELECOMUNICAÇÕES PARA PREDIÇÃO DE
TRÁFEGO DE INTERFACE DE BNG E ANÁLISE DE RELATÓRIO DE FALHA DE
REDE GPON

ORIENTANDO: WILLIAN SANTOS SILVA
ORIENTADOR: Dr. ANIEL SILVA DE MORAIS.

DEZEMBRO
2020

WILLIAN SANTOS SILVA

*APLICAÇÃO DE REDES NEURAIIS RECORRENTES E SELF-ORGANIZING MAPS EM
DADOS REAIS DE OPERADOR DE TELECOMUNICAÇÕES PARA PREDIÇÃO DE
TRÁFEGO DE INTERFACE DE BNG E ANÁLISE DE RELATÓRIO DE ANÁLISE DE
FALHA DE REDE GPON*

Dissertação submetida ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **MESTRE EM CIÊNCIAS**.

Área de concentração: Controle e automação

Orientador: Dr. Aniel Silva de Morais.

UBERLÂNDIA - MG

2020

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

S586a
2020 Silva, Willian Santos, 1989
Aplicação de redes neurais recorrentes e self-organizing maps em dados reais de operador de telecomunicações para predição de tráfego de interface de bng e análise de relatório de análise de falha de rede GPON [recurso eletrônico] / Willian Santos Silva. - 2020.

Orientador: Aniel Silva de Moraes.
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.

Modo de acesso: Internet.

Disponível em:

<http://dx.doi.org/10.14393/ufu.S586a> Inclui
bibliografia.

Inclui ilustrações.

1. Engenharia elétrica. I. Moraes, Aniel Silva de, 1979, (Orient.) II.
Universidade Federal de Uberlândia. Programa de Pós-Graduação em
Engenharia Elétrica. III. Título.

CDU: 621.3

Angela Aparecida Vicentini Tzi Tziboy – CRB-6/947



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
 Coordenação do Programa de Pós-Graduação em Engenharia Elétrica
 Av. João Naves de Ávila, 2121, Bloco 3N - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
 Telefone: (34) 3239-4707 - www.posgrad.feelt.ufu.br - copel@ufu.br



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Elétrica				
Defesa de:	Dissertação de Mestrado Acadêmico, 752, PPGEELT				
Data:	Dezenove de novembro de dois mil e vinte	Hora de início:	09:00	Hora de encerramento:	12:00
Matrícula do Discente:	11912EEL006				
Nome do Discente:	Willian Santos Silva				
Título do Trabalho:	Aplicação de redes neurais recorrentes e selforganizing maps em dados reais de operador de telecomunicações para predição de tráfego de interface de bng e análise de relatório de falha de rede GPON				
Área de concentração:	Sistemas de energia elétrica				
Linha de pesquisa:	Controle e automação				
Projeto de Pesquisa de vinculação:	Título: Técnicas de identificação de sistemas visando aplicabilidade em processos industriais Agência Financiadora: UNIVERSIDADE FEDERAL DE UBERLÂNDIA - (Programa Institucional de Apoio a Pesquisa (apoio financeiro)) Início 02/01/2013 Término __/__/__MESTRADO No. do Projeto na agência: __ Professor Coordenador: Marcio Jose da Cunha				

Reuniu-se por meio de videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta: Professores Doutores: Márcio José da Cunha - FEELT/UFU; Josué Silva de Moraes - FEELT/UFU; André Luís Dias - IFSP; Aniel Silva de Moraes - FEELT/UFU, orientador(a) do(a) candidato(a).

Iniciando os trabalhos o(a) presidente da mesa, Dr(a). Aniel Silva de Moraes, apresentou a Comissão Examinadora e o candidato(a), agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor(a) presidente concedeu a palavra, pela ordem sucessivamente, aos(às) examinadores(as), que passaram a arguir o(a) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o(a) candidato(a):

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Aniel Silva de Moraes, Professor(a) do Magistério Superior**, em 19/11/2020, às 11:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **ANDRE LUIS DIAS, Usuário Externo**, em 19/11/2020, às 11:46, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Marcio José da Cunha, Professor(a) do Magistério Superior**, em 19/11/2020, às 11:46, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Josué Silva de Moraes, Professor(a) do Magistério Superior**, em 19/11/2020, às 11:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2318387** e o código CRC **B92A2A22**.

“Não há nada de nobre em sermos superiores ao próximo. A verdadeira nobreza consiste em sermos superiores ao que éramos antes.”

Esopo.

AGRADECIMENTOS

Agradeço ao professor e orientador Dr. Aniel Silva de Moraes pelo incentivo, motivação, orientação e paciência nesta caminhada acadêmica.

A minha família, meus pais Fábio e Wilma, meus irmãos Gabriel e Rafaela, pelo apoio incondicional em toda a minha caminhada, e que mesmo nos tempos de dificuldades, estiveram ao meu lado.

A minha noiva Carolina, que desde que nos conhecemos tem sido inspiração para que eu sempre busque crescer como pessoa, que me apoiou nos momentos mais difíceis, me acompanhou em grande parte dos momentos de maior luta, meu eterno amor e gratidão.

Agradeço a Faculdade de Engenharia Elétrica, em seu núcleo de pós-graduação, aos professores e funcionários que tem trabalhado incansavelmente contribuindo para a formação de centenas de profissionais, gerando grande impacto econômico e social.

Aos colegas de laboratório LAMEP (Laboratório de Aterramentos, Máquinas e Processamento de Energia).

A Universidade Federal de Uberlândia, que tem sido ao longo dos anos formadora de profissionais de excelência, buscando a expansão das oportunidades para aqueles que, como eu, não conseguiriam ter acesso ao ensino superior e pós graduação gratuita de qualidade.

Dedico este trabalho às minhas avós, Celina e Arlinda, que sempre desejaram me ver chegando mais longe. Elas infelizmente não estão mais entre nós, mas viverão eternamente em meu coração.

RESUMO

Este trabalho consiste na aplicação de dois algoritmos de Inteligência Artificial em duas situações reais de uma operadora de telecomunicações. A primeira aplicação é o conceito de rede neural recorrente ou *recurrent neural network* (RNN). Um tipo especial de rede neural, que possui grande aplicabilidade em cenários com dados temporais e que foi aplicada para previsão de tráfego em uma interface de um *broadband remote access server* (também denominado de *broadband network gateway*). A segunda situação de inteligência artificial aplicada foi a avaliação da utilização do modelo de rede neural *self-organizing maps* (SOM). Este foi utilizado para classificar e detectar dados inconsistentes de um relatório de falhas em campo de uma rede de acesso banda larga GPON. A utilização de SOM para detecção de anomalias e classificação conseguiu fazer a redução da dimensionalidade dos dados. Foi possível extrair a lista desses dados distantes (ou outliers) e fazer uma observação. O estudo busca demonstrar que as aplicações de redes neurais podem ser utilizadas como ferramenta para automatização de análises de rede em telecomunicações, gerando benefícios como redução de custo e maior agilidade o que propicia melhor qualidade aos usuários, redes mais otimizadas e evolução dos serviços de telecomunicações.

Palavras-chave: Redes neurais recorrentes, mapas auto-organizáveis, LSTM, previsão de tráfego, análise de falhas.

ABSTRACT

This work consists of the application of two Artificial Intelligence algorithms in two real situations of a telecommunications operator. The first application is the concept of recurrent neural network or recurrent neural network (RNN). A special type of neural network, which has great applicability in scenarios with temporal data and which was applied for traffic forecast in an interface of a broadband remote access server (also called broadband network gateway). The second situation of applied artificial intelligence was the evaluation of the use of the self-organizing maps (SOM) neural network model. This was used to classify and detect inconsistent data from a field failure report from a GPON broadband access network. The use of SOM for anomaly detection and classification was able to reduce the dimensionality of the data. It was possible to extract the list of these distant data (or outliers) and make an observation. The study seeks to demonstrate that neural network applications can be used as a tool for automating network analysis in telecommunications, generating benefits such as cost reduction and greater agility, which provides better quality to users, more optimized networks and evolution of telecommunications services.

Keywords: Recurrent neural networks, self-organizing maps, LSTM, traffic prediction, failure analysis.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - EVOLUÇÃO DOS ACESSOS DE BANDA LARGA FIXA	18
FIGURA 2 - TOTAIS DE ACESSOS BANDA LARGA POR TIPO DE TECNOLOGIA	19
FIGURA 3- ESTRUTURA DE REDE COM BNG DE INTERFACE ENTRE CORE E ANEL DE AGREGAÇÃO	25
FIGURA 4- INTERLIGAÇÃO DE REDES DE ACESSO COM REDE METROPOLITANA.....	26
FIGURA 5- ESTRUTURA CABO DE FIBRA ÓPTICA DE 1 FIBRA	27
FIGURA 6- DIAGRAMA DE UM PROJETO GPON	29
FIGURA 7- NEURÔNIO ARTIFICIAL	34
FIGURA 8- GRÁFICOS DE FUNÇÕES DE ATIVAÇÕES.....	34
FIGURA 9- REPRESENTAÇÃO DE CAMADAS LSTM.....	37
FIGURA 10- REPRESENTAÇÃO DAS ETAPAS DE FUNCIONAMENTO DE UM LSTM.....	38
FIGURA 11- REPRESENTAÇÃO DAS INTERAÇÕES DE UM LSTM.....	39
FIGURA 12- SEGUNDA ETAPA LSTM, INPUT GATE.....	40
FIGURA 13- ATUALIZAÇÃO DO ESTADO C_T, CELL UPDATE.....	41
FIGURA 14- ATUALIZAÇÃO DA ETAPA DE SAÍDA LSTM	41
FIGURA 15- REPRESENTAÇÃO DO MAPA DE KOHONEN E SEUS VETORES DE ENTRADA	43
FIGURA 16- ETAPAS DE EXECUÇÃO DO TESTE USANDO SOM	51
FIGURA 17- AMOSTRA DOS DADOS UTILIZADOS EM CSV	54
FIGURA 18- TRÁFEGO INCOMING E OUTGOING UTILIZADO (TODAS AS AMOSTRAS DE 1 MÊS)....	54
FIGURA 19- ETAPAS REALIZADAS PARA PELO CÓDIGO USANDO RNN E EM CADA EXECUÇÃO....	56
FIGURA 20- MAPA SOM COM GRID 10X10 REPRESENTANDO A VISÃO 2D DOS DADOS CONFORME SEMELHANÇA.....	59
FIGURA 21- MAPA SOM COM AUXÍLIO DE MARCADORES CONFORME AS DUAS SAÍDAS DO DATASET.....	59
FIGURA 22- LISTA DE OUTLIERS CONFORME DATASET INSERIDO, VISÃO SPYDER	60
FIGURA 23- VISÃO RELATÓRIO CSV	61
FIGURA 24- MAPA DO TESTE 1.2.3.1.....	63
FIGURA 25- MAPA DO TESTE 1.3.1.....	63
FIGURA 26- MAPA TESTE 1.5.2.5	64
FIGURA 27 - GRÁFICO REAL X PREDIÇÕES DE TRÁFEGO INCOMMING TESTE 2.1.....	67
FIGURA 28- CURVAS PERDA POR ÉPOCAS PARA TESTES DE VARIAÇÃO DE TIMESTEP.....	67
FIGURA 29- CURVAS DO TESTE 2.2 CONCATENADAS	69

FIGURA 30- TESTE LSTM 2.2 COM 4 CAMADAS	69
FIGURA 31- TESTE LSTM COM 5 CAMADAS	69
FIGURA 32- TESTE LSTM COM 8 CAMADAS	70
FIGURA 33- TESTE LSTM COM 12 CAMADAS.	70
FIGURA 34- CURVAS PERDAS POR ÉPOCA VARIANDO QUANTIDADE DE CAMADAS.....	70
FIGURA 35- CURVAS DE PREDIÇÃO VARIANDO QUANTIDADE DE NEURÔNIOS	71
FIGURA 36- VISÃO DE PERDA POR ÉPOCAS COM VARIAÇÃO DE NEURÔNIOS	72
FIGURA 37- CURVAS DE PREDIÇÃO VARIANDO QUANTIDADE DE ÉPOCAS.....	73
FIGURA 38- PERDA POR ÉPOCA NO TESTE 2.4.....	73
FIGURA 39 - CURVAS PARA TESE OTIMIZADO DE TRÁFEGO INCOMMING	74
FIGURA 40- PERDA POR ÉPOCAS TESTE 2.5	75
FIGURA 41- PREDIÇÃO DE TREFEGO TESTE 2.6 OUTCOMMING TRÁFFIC OTIMIZADO	75
FIGURA 42- PERDA POR ÉPOCA TESTE 2.6	76

LISTA DE TABELAS

TABELA 1- INDICADORES DE ACESSOS ANATEL FEVEREIRO 2020	17
TABELA 2- EVOLUÇÃO DOS ACESSOS BANDA LARGA POR FAIXA DE VELOCIDADE	18
TABELA 3- TRABALHOS TERMO INTELIGÊNCIA ARTIFICIAL DE 2016 A 2019 NA PLATAFORMA SCOPUS.....	21
TABELA 4- VOLUME E ÍNDICE DE RECLAMAÇÕES POR SERVIÇO (2015-2019).....	31
TABELA 5- GRUPOS DE TESTES EXECUTADOS	47
TABELA 6- TIPOS E QUANTIDADES DE DADOS REAIS UTILIZADOS POR CATEGORIA	49
TABELA 7- CONJUNTO DE TESTES APLICADO USANDO SOM.....	52
TABELA 8- TESTES COM LSTM EM PREDIÇÃO DE INTERFACE BRAS	57
TABELA 9- QUANTIDADE DE OUTLIERS IDENTIFICADOS DE ACORDO COM OS TESTES.....	62
TABELA 10- COORDENADAS OBSERVADAS NOS MAPAS QUE INDICARAM OUTLIERS (X E Y)	62
TABELA 11- DISTRIBUIÇÃO ESTATÍSTICA DE QUANTIDADE DE OUTLIERS POR TESTE	64
TABELA 12- DISTRIBUIÇÃO ESTATÍSTICA CONSIDERANDO QUANTIDADE DE ITENS CLASSIFICADOS POR TESTE E SUBTESTE	65
TABELA 13 - INDICADORES PARA TESTES VARIANDO TIMESTEPS	68
TABELA 14 - INDICADORES TESTES VARIANDO CAMADAS	71
TABELA 15- INDICADORES DOS TESTES VARIANDO QUANTIDADE DE NEURÔNIOS	72
TABELA 16- INDICADORES DOS TESTES VARIANDO QUANTIDADE DE ÉPOCAS	74

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ACM	Association Computer Machinery
ANATEL	Agência Nacional de Telecomunicações
APON	ATM Passive Optical Network
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Network
ATM	Asynchronous Transfer Mode
BMU	Best Match Unit
BNG	Broadband network Gateway
BPON	Broadband Passive Optical Network
BRAS	Broadband remote access server
CAPEX	Capital Expenditure
CEO	Caixa de Emenda Óptica
CPE	Customer Premise Equipment
CTO	Caixa de Terminação Óptica
DBA	Dynamic Bandwidth Allocation
DGO	Distribuidor Geral Óptico
DHCP	Dynamic Host Configuration Protocol
DIO	Distribuidor Interno Óptico
DSL	Digital Subscriber Line
EAQ	Entidade Aferidora da Qualidade
EPON	Ethernet Passive Optical Network
EPON	Ethernet Passive Optical Network
FTP	File Transfer Protocol
FTTb	Fiber to the build
FTTh	Fiber to the home
GPON	Gigabit Passive Optical Network
HFC	Hybrid Fiber Coax
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet Protocol

IPoE	Internet Protocol over Ethernet
ISDN	Integrated Service Digital Network
ISP	Internet Service Provider
ITU	International Telecommunication Union
LAN	Local Area Network
LGPD	Lei Geral de Proteção de Dados Pessoais
LSTM	Long Short-Term Memory
NOC	Network operations center
NSP	Network Service Provider
ODN	Optical Distribution Network
OLT	Optical Line Terminal
OMCI	ONU Management Control Interface
ONT	Optical Network Terminal
ONU	Optical Network Unit
OPEX	Operational Expenditure
OSI	Open System Interconnection
PD	Proporcional Derivativo
PI	Proporcional Integral
PID	Proporcional Integral Derivativo
PMD	Physical Media Dependent
PON	Passive Optical Network
POP	Point of Presence
POTS	Plain Old Phone Telephone Service
PPP	Point-to-Point Protocol
PPPoE	Point-to-Point Protocol over Ethernet
QoS	Quality of Service
RNA	Rede neural artificial
RNN	Recurrent neural network
SCM	Serviço de Comunicação Multimídia
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOM	Self-organizing maps
TCP	Transmission Control Protocol

UDP	User Datagram Protocol
DM	Wavelength-division Multiplex

SUMÁRIO

1 INTRODUÇÃO.....	17
1.1 Inteligência artificial e solução de problemas.....	20
1.2 Objetivo.....	22
1.1.1 Objetivos específicos.....	22
1.3 Organização.....	23
2 FUNDAMENTAÇÃO TEÓRICA.....	24
2.1 Redes e a Internet.....	24
2.1.2 BRAS/BNG.....	24
2.1.4 Redes de Acesso GPON.....	26
2.1.5 Qualidade e manutenção de Redes.....	30
2.2 Inteligência Artificial.....	32
2.2.1 Redes Neurais Artificiais.....	32
2.2.2 Redes Neurais Recorrentes e LSTM.....	36
2.2.3 Self-Organizing Maps.....	42
3 METODOLOGIA DA APLICAÇÃO.....	46
3.1 Levantamento dos dados reais para aplicação em algoritmo SOM.....	47
3.2 Ambiente de desenvolvimento SOM.....	49
3.3 Aplicação <i>Self-Organizing Maps</i>.....	50
3.4 Levantamento dos dados reais para aplicação em algoritmo LSTM.....	53
3.5 Ambiente de desenvolvimento RNN.....	55
3.6 Aplicação de LSTM e testes.....	55
4 RESULTADOS E DISCUSSÃO.....	58
4.1 Testes usando Som.....	58
4.2 Testes usando RNN.....	66
4.2.1 Teste RNN 2.1 variando timesteps.....	66
4.2.2 Teste RNN 2.2 variando camadas.....	69
4.2.3 Teste RNN 2.3 variando quantidade de neurônios.....	71
4.2.4 Teste RNN 2.4 variando quantidade de épocas.....	73
4.2.5 Teste 2.5 predição de tráfego incomming.....	74
4.2.6 Teste 2.6 predição de trafego outcomming.....	75

5 CONCLUSÃO.....	77
REFERÊNCIAS	80
ANEXO A – DADOS E CÓDIGO APLICADO PARA APLICAÇÃO SOM.....	88
ANEXO B – DADOS E CÓDIGO APLICAÇÃO RNN.....	91

1 INTRODUÇÃO

Com o desenvolvimento do computador pessoal surgiram grandes evoluções na indústria, ciências e educação. Também importantes mudanças nas comunicações de dados, surgimento das redes, sua evolução e crescimento possibilitando o uso de sua capacidade expandida (FOROUZAN, 2010).

Rede é um grupo de dispositivos de comunicação conectados entre si, como os computadores. Uma internet (i minúsculo) são duas ou mais redes que podem se comunicar entre si. A Internet (I maiúsculo) são centenas de milhares de redes interconectadas. Um sistema com milhões de usuários que passou a existir apenas em 1969 (FOROUZAN, 2010). As redes podem ser de diversos tamanhos, modelos e formas e a Internet também pode ser definida como uma rede de redes (TANENBAUM, 2011).

A Internet de hoje é provavelmente o maior sistema de engenharia já criado pela humanidade, com centenas de milhões de computadores conectados, enlaces de comunicação e computadores. Bilhões de usuários que se conectam por meio de laptops, tablets e smartphones; e com uma série de dispositivos como sensores, webcams, console para jogos, quadros de imagens, e até mesmo máquinas de lavar sendo conectadas (KUROSE; ROSS, 2013).

Com sua evolução, o atendimento aos consumidores evoluiu significativamente, possibilitando acessos de maior largura de banda a um menor custo. Juntamente com a microeletrônica, a computação e a evolução das aplicações, fizeram elevar o tráfego de dados de forma exponencial.

Observando o panorama Brasileiro na tabela 1, segundo a Agência Nacional de Telecomunicações (ANATEL), o Brasil chegou em fevereiro de 2020 a marca de 33,9 milhões de acessos à internet por banda larga fixa, o que significa que de cada 100 domicílios, aproximadamente 49 possuem acesso fixo à internet (ANATEL, 2019).

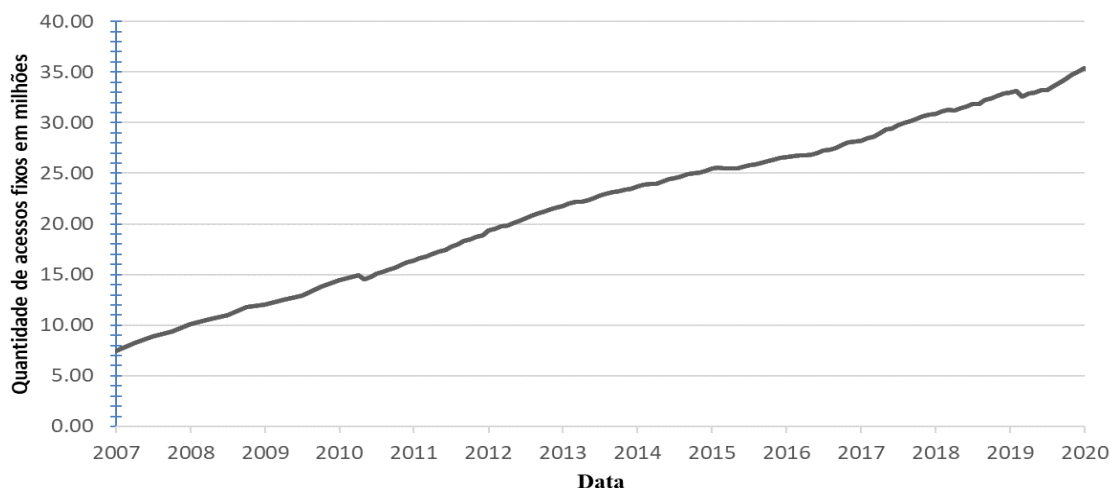
Tabela 1- Indicadores de acessos ANATEL fevereiro 2020

Tipo de acesso	Banda larga fixa	Telefonia Móvel	TV por Assinatura	Telefonia Fixa
Quantidade de acessos (Milhões)	33,9	225,1	15,2	31,5
Densidade (acesso/100 domicílios)	48,8	95,7	21,8	45,2

Fonte: ANATEL (2020).

O salto no número de acessos à banda larga fixa foi de 14,95 milhões em 2010 para 33,9 milhões de acessos (considerando maio de 2020) conforme figura 1. Um salto de 126% na quantidade de acessos, valor muito expressivo e que demonstra o seu crescimento em presença, importância social e comercial e como novo meio de propagar informações e comunicação.

Figura 1 - Evolução dos acessos de Banda Larga Fixa



Fonte: O autor baseado em relatório de acessos banda larga Anatel (2020)

Observando tanto os dados quanto o perfil de produtos (velocidades dos acessos banda larga), o relatório de evolução de velocidade do órgão demonstra um comportamento exponencial de aumento de demanda em maiores velocidades acima de 34 Mbps, sendo que em 2020 já surgem prestadores de serviço de internet oferecendo velocidades de até 1Gbps. (ANATEL, 2020).

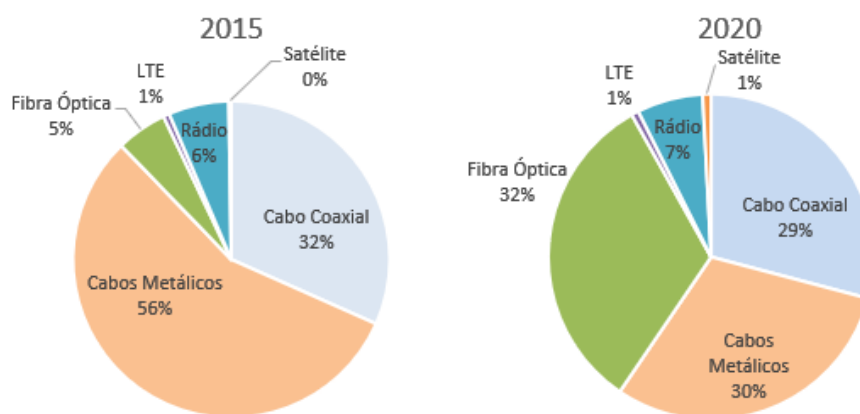
Tabela 2- Evolução dos acessos Banda Larga por faixa de velocidade

Faixa de velocidades	0Kbps a 512Kbps	2Mbps a 12Mbps	512kbps a 2Mbps	12Mbps a 34Mbps	> 34Mbps	Total Geral
Quantidade de acessos 2015 (Milhões)	1,31	10,70	8,19	3,13	0,82	24,16
Quantidade de acessos 2020 (Milhões)	0,39	7,77	3,39	7,28	14,12	32,95
Evolução em 5 anos	-70%	-27%	-59%	133%	1613%	36%

Fonte: ANATEL (2020).

Existe uma relação entre maior demanda de velocidade e evolução tecnológica se observado o número de acesso em fibra óptica comparado aos meios metálicos. O percentual a quantidade de acessos em fibra saltou de 1,25 milhões de acessos (2015) para 10,67 milhões em 2020. Um salto de 5% em 2015 para 32% em 2020 dos acessos conforme figura 2 (ANATEL, 2020).

Figura 2 - Totais de acessos banda larga por tipo de tecnologia



Fonte: O autor baseado em relatório de acessos banda larga Anatel (2020)

Com o aumento da demanda é inevitável o aumento da necessidade de profissionais capacitados, melhorias de equipamentos e sistemas para manter níveis de qualidade aceitáveis pelos clientes, cumprindo as normas de competitividade.

Fazendo uma analogia ao setor industrial, conforme descrito no trabalho de Oliveira (2020), a tarefa de responder a eventos anormais do processo em sua grande maioria é feita por humanos. Envolvendo a detecção de um evento anormal; diagnóstico de origens, causas, e tomada de decisão para o controle. Em parte, essa tarefa se tornou difícil para os operadores humanos, devido à grande quantidade de variáveis. Sob pressão, devido à complexidade de se operar, essas pessoas podem vir a tomar decisões e ações erradas (OLIVEIRA, 2020).

A ação de desenvolver ferramentas para automatização dos casos escolhidos agregam valor ao planejamento de engenharia de capacidade e operação. Os equipamentos e projetos de execução demandam que as empresas tenham recursos financeiros disponíveis (planejamento financeiro), demandam equipes para execução e que sejam adquiridos equipamentos eletrônicos. Grande parte dos insumos necessários são importados, sofrem alterações do câmbio e demandam prazos de importação. Assim a previsão e correta identificação de causas de falhas

se torna meio de planejar com antecedência e criar projetos e ações com custo e prazo adequado. Como consequência é possível manter competitividade, qualidade e benefícios a usuários e empresas.

1.1 Inteligência artificial e solução de problemas

A utilização ou desenvolvimento de softwares para automatização dessas atividades que exigem repetição ou avaliação de grande volume de informações contribuem para a redução de erros. Conseqüentemente, contribuem na melhoria da qualidade dos serviços além da redução do custo operacional.

Um sistema especialista procura modelar o comportamento de um especialista em uma área. Se baseiam em regras, sendo capazes de simular as mesmas ações que um especialista faria para chegar a uma conclusão (COPPIN, 2013).

Dentre as técnicas inspiradas no ser humano temos a Inteligência Artificial. Seu funcionamento possibilita o aprendizado de padrões (em situações muitas vezes complexas), a partir de elementos simples que simulam o comportamento de neurônios (MEDEIROS, 2018).

Dentre as diversas teorias desenvolvidas temos as redes neurais artificiais. São modelos computacionais que se baseiam no sistema nervoso, realizando o processamento de informações em unidades denominadas neurônios artificiais (SILVA; SPATTI; FLAUZINO, 2010). Esses algoritmos podem reconhecer padrões escondidos e correlacionar dados brutos, agrupá-los, classificá-los, e com o tempo, aprender e melhorar os resultados.

Nas estruturas de redes artificiais auto-organizáveis destaca-se o trabalho do finlandês Teuvo Kohonen (1984). Essas redes foram inspiradas no córtex cerebral, que é ativa por frente a determinados estímulos (SILVA; SPATTI; FLAUZINO, 2010).

A pesquisa em IA ganhou grande relevância com o passar dos anos, principalmente com a evolução da capacidade computacional, um panorama sobre o IEEE *Xplore Top Searches* mostra alguns assuntos que estão diretamente ligados às áreas de inteligência artificial como processamento de imagens com mais de 374 mil buscas, *Data Mining* com mais de 118 mil pesquisas, e o termo de pesquisa *Deep Learning* com de 46,6 mil consultas em agosto de 2020 (IEEE Xplore, 2020).

Por ordem de pesquisa no mês de agosto de 2020, as buscas mais populares no IEEE *Xplore*. têm em segundo lugar processamento de imagens, em sétimo lugar *deep learning*. Além

disso, vários dos demais temas como 5G, *Internet of Things*, *Antenna*, são assuntos diretamente ligados a evolução das redes e Internet. Na base de pesquisas Google Scholar o termo "*recurrent neural network*" chegou a 44 mil resultados de 2016 a 2020. O termo "*artificial neural networks*" possui mais de 214 mil resultados em agosto de 2020. Avaliando pela plataforma Scopus o termo inteligência artificial atingiu a seguinte abrangência conforme a tabela 3.

Tabela 3- Trabalhos termo inteligência artificial de 2016 a 2019 na plataforma Scopus

<i>Source title</i>	<i>Highest percentile</i>	2016-19 <i>Citations</i>	2016-19 <i>Documents</i>	% <i>Cited</i>	<i>Publisher</i>
IEEE Transactions on Pattern Analysis and Machine Intelligence	99.0% Applied Mathematics	28267	804	93	IEEE Now Publishers Inc.
Foundations and Trends in Machine Learning	99.0% Human-Computer Interaction	313	12	83	
IEEE Transactions on Cybernetics	99.0% Control and Systems Engineering	25372	1245	95	IEEE
IEEE Transactions on Systems, Man, and Cybernetics: Systems	99.0% Control and Systems Engineering	15852	866	97	IEEE
IEEE Transactions on Neural Networks and Learning Systems	98.0% Computer Science Applications	23443	1331	89	IEEE
IEEE Transactions on Fuzzy Systems	99.0% Applied Mathematics	12475	768	90	IEEE
International Journal of Computer Vision	96.0% Artificial Intelligence	5205	341	87	Springer Nature

Fonte: SCOPUS (2020).

A aplicação de conceitos e trabalhos correlatos usando *self-organizing maps* como método de criação de grupos com características semelhantes, ou também chamado *clustering* são trabalhos como os de (SHARMA; SHARMA; DWIVEDI, 2018); (PILLAI *et al.*, 2019) e (AFFONSO, 2011), mineração e avaliação de falhas apresentado em (HASHMI; DARBANDI; IMRAN, 2017). Uso de *self-organizing maps* como detecção de outliers apresentado por (YANG *et al.*, 2019) e detecção de anomalias como apresentado por (ZAPATER, 2014).

Outra aplicação dentro de inteligência artificial são as redes neurais recorrentes, do inglês *Recurrent Neural Networks* (RNN) e como trabalhos correlatos (RUSEK *et al.*, 2020); (HUANG; CHUNG; TANG, 2018); (TELES VIEIRA; LEMOS; LEE, 2003); (TIAN *et al.*, 2020); o uso em predição de tráfego conforme (ZHAO *et al.*, 2018).

O código e metodologia foi inspirado nas referências de curso de aprendizado de máquina aplicado (EREMENKO; PONTEVES, 2018).

1.2 Objetivo

A necessidade de técnicas de automatização juntamente com mais diversas oportunidades de aplicações baseadas em IA, propicia que os problemas hoje tratados por especialistas em serviços de telecomunicações (e nas mais diversas áreas) possam através deste trabalho ser estudados.

A sociedade passa por um crescimento da demanda por serviços de Internet de forma exponencial. Cada dia se torna mais urgente a aplicação de modelos que auxiliem órgãos públicos e privados na obtenção rápida e precisa de informações que possam auxiliar a melhorar os serviços. Fato que inspira este trabalho e a busca por melhores modelos em trabalhos futuros.

Como objetivo geral o trabalho busca utilizar IA para automatizar situações de uma operadora de telecomunicações, hoje realizadas por especialistas. Para tal, são utilizados os conceitos de SOM e RNN na análise dos dados, automatizando parte do processo. Com isto reduz-se a chance do erro humano e tempo de análise.

Com uma avaliação mais assertiva é possível um planejamento com redução do investimento e reinvestimento em equipamentos, melhorar a velocidade de entrega e ampliações, (em grande parte atreladas a importação de equipamentos com prazos longos e sujeitos a variação cambial). Um planejamento de ampliações de rede adequado também resulta em maior qualidade e competitividade dos serviços prestados.

1.1.1 Objetivos específicos

Como objetivos específicos, a primeira situação é criar grupo por similaridade (cluster) em um mapa de Kohonen uma base real de um relatório de análise de falha massiva GPON, encontrar os dados discrepantes, investigar possíveis erros de análise pelos especialistas de manutenção.

O segundo objetivo específico é automatizar a predição de comportamento do tráfego de dados em interfaces reais de broadband network access utilizando a rede neural recorrente LSTM.

1.3 Organização

Este trabalho está estruturado da seguinte forma:

O CAPÍTULO 1 aborda a contextualização do tema e das áreas abordadas além de introduzir alguns conceitos e posteriormente justificar o problema e objetivos da pesquisa.

CAPÍTULO 2 apresenta a fundamentação teórica, abordando conceitos da aplicação em telecomunicações e fundamentação das ferramentas de inteligência artificial utilizadas.

CAPÍTULO 3 descreve o cenário, as premissas utilizadas e metodologia de aplicação e desenvolvimento.

CAPÍTULO 4 apresenta os resultados obtidos da aplicação, bem como a descrição de suas características.

CAPÍTULO 5 reúne a conclusão dos fatos observados de acordo com a metodologia e premissas adotadas na aplicação, observações e proposta de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Redes e a Internet

O termo dados se refere a informações que sejam acordadas entre as partes que criam e as usam. Já comunicação de dados são as trocas de dados entre dois dispositivos por intermédio de algum tipo de meio de transmissão, como um cabo condutor formado por fios. Estes dispositivos que trocam dados são parte de um sistema constituído de hardware (equipamento físico) e software. Os dados devem ser recebidos no destino correto, com garantia de entrega, tempo finito e predeterminado (FOROUZAN, 2010).

Os sistemas finais são conectados por enlaces (links) e comutadores de pacotes, que são equipamentos que encaminham os pacotes (conjunto de dados) de um enlace de entrada para um enlace de saída.

Estar conectado a outros computadores locais não garante o acesso à internet, é necessário que essas redes locais (LANs – *Local Area Network*) estejam conectadas às redes globais. Assim os sistemas finais acessam a internet por meio de ISPs (*Internet Service Providers*) ou os Provedores de Serviço de Internet, que são as empresas de telecomunicações que os usuários contratam para obter o fornecimento de acesso à internet. O usuário contrata todo um serviço de infraestrutura com enlaces, links, roteadores, *switches* e caminhos para poder acessar outros sistemas finais pelo mundo que também estejam com uma conexão estabelecida.

Os ISPs também fornecem acesso a provedores de conteúdo e devem se interconectar através de ISPs maiores nacionais ou internacionais com roteadores de alta velocidade e enlaces de fibra ótica, o que é chamado interconexão. Apesar de estarem conectados, cada ISP tem sua gerência de forma independente, utilizando um protocolo IP (*Internet Protocol*) e seguindo regras de endereçamento (KUROSE; ROSS, 2013).

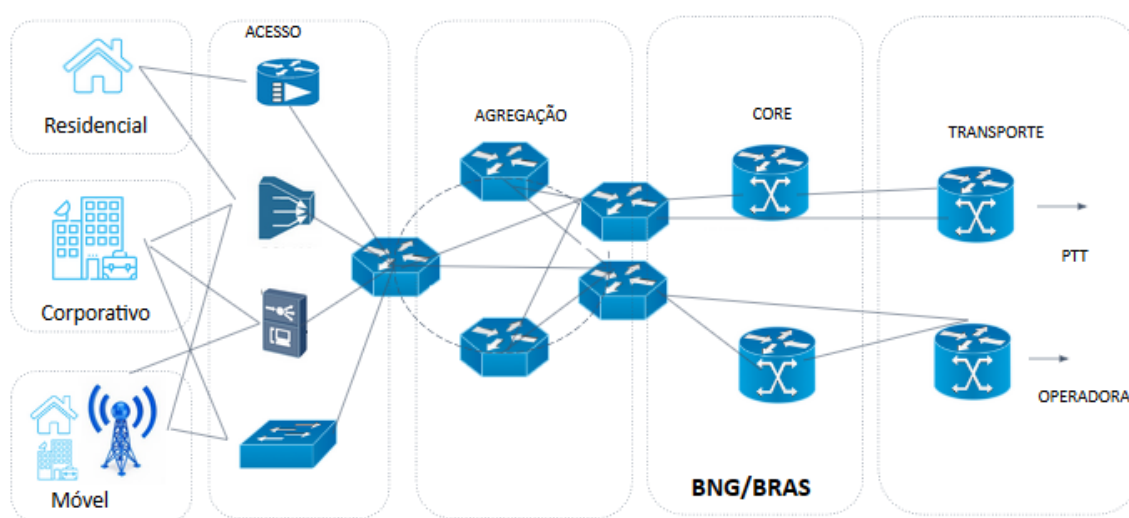
2.1.2 BRAS/BNG

O servidor de acesso remoto de banda larga (BRAS) é um tipo de *gateway*, ou seja, equipamento que faz interface em ponto da rede e que usa protocolos diferentes. Este equipamento é usado para aplicação de banda larga. É usado principalmente para construir a Rede de Área Metropolitana (em inglês *metropolitan area network* MAN) e uma interliga a

área de acesso banda larga e a rede de *backbone* (rede de transporte de dados ligando pontos distantes). Também fornece métodos de acesso e funções de gerenciamento (WANG, J. *et al.*, 2008).

O BRAS (também conhecido como B-RAS ou BBRAS) roteia o tráfego para nós de acesso, tais como terminais de linha óptica (OLTs) na rede de uma operadora. Esses equipamentos estão localizados na borda do núcleo da rede, como exemplificado na figura 3.

Figura 3- Estrutura de rede com BNG de interface entre Core e anel de agregação



Fonte: O autor baseado em (LINS, 2018).

O objetivo da arquitetura BNG é permitir que o roteador BNG interaja com dispositivos periféricos dos clientes e servidores (como DHCP - *Dynamic Host Configuration Protocol*), a fim de fornecer conectividade de banda larga e gerenciar sessões.

Em um cenário evolutivo, a capacidade do BRAS é centralizada em uma rede de malha óptica regional. São uma parte essencial das topologias de banda larga para controlar o acesso dos assinantes à rede. Quando uma conexão for estabelecida entre o BNG e o *Customer Premise Equipment* (CPE) que é o equipamento final. O assinante pode acessar os serviços fornecidos pelo *Network Service Provider* (NSP) ou *Internet Service Provider* (ISP) (LINS, 2018).

Quando uma sessão está ativa, o BNG agrega o tráfego de várias sessões de assinantes de uma rede de acesso e o roteia para a rede do provedor de serviços (ISP). Ele é implantado no primeiro ponto de agregação na rede, como o roteador de borda.

Outra atividade que realiza é gerenciar o acesso e as funções do assinante como: autenticação, autorização e contabilidade de sessões de assinante, atribuição de endereço; segurança, gerenciamento de políticas, qualidade de serviço (QoS) (PAING, 2016).

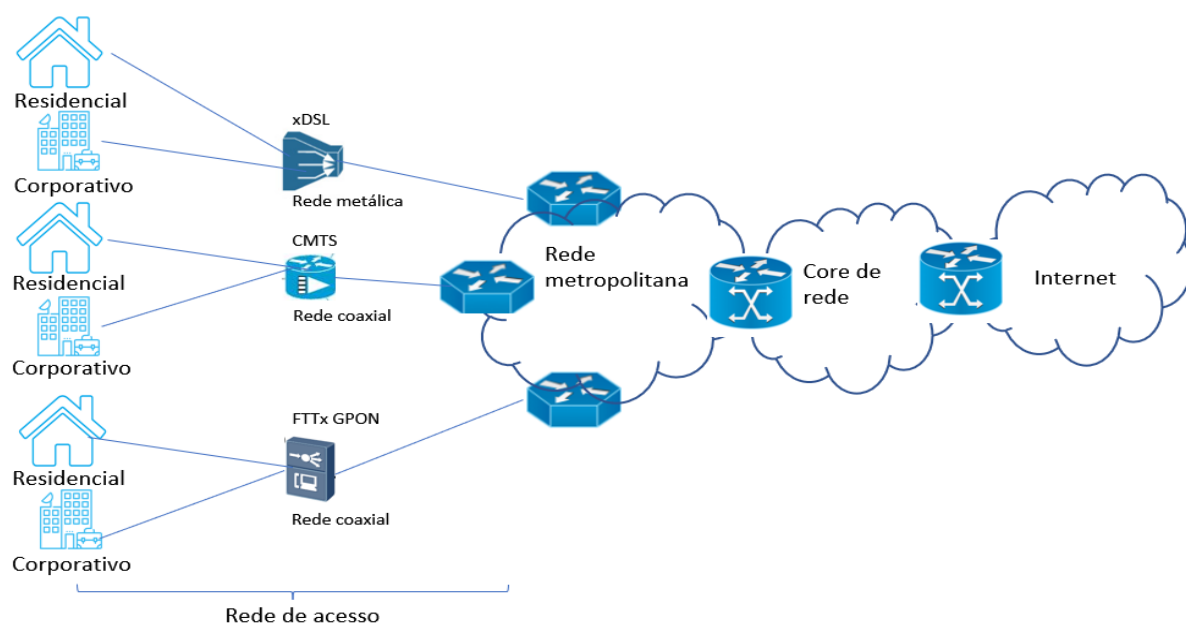
Outra característica é estabelecer sessões de assinante usando protocolos IPoE ou PPPoE além de fazer a agregação de circuitos de um ou mais dispositivos de acesso. Interage com os servidores DHCP e assim fornece endereços IP aos clientes. Roteia o tráfego IP em uma rede de *backbone* do ISP para a Internet (PAING, 2016).

Trabalhos que aplicam inteligência artificial em redes de transporte são apresentados em (WANG, J. *et al.*, 2008); no trabalho de (ZHANG *et al.*, 2019) e (LIU *et al.*, 2019).

2.1.4 Redes de Acesso GPON

A rede de acesso é a parte da estrutura de rede responsável por interligar os usuários aos nós de processamento conforme ilustrado na figura 4, fazendo assim uma conexão de rede entre operadora e cliente. Sua construção é feita de forma que possa conectar pontos de convergência e atendimentos de forma individual. Na figura 4 é representado alguns tipos de rede de acesso, como as redes metálicas, redes de cabos coaxiais e as redes de acesso em fibra óticas.

Figura 4- Interligação de redes de acesso com rede metropolitana



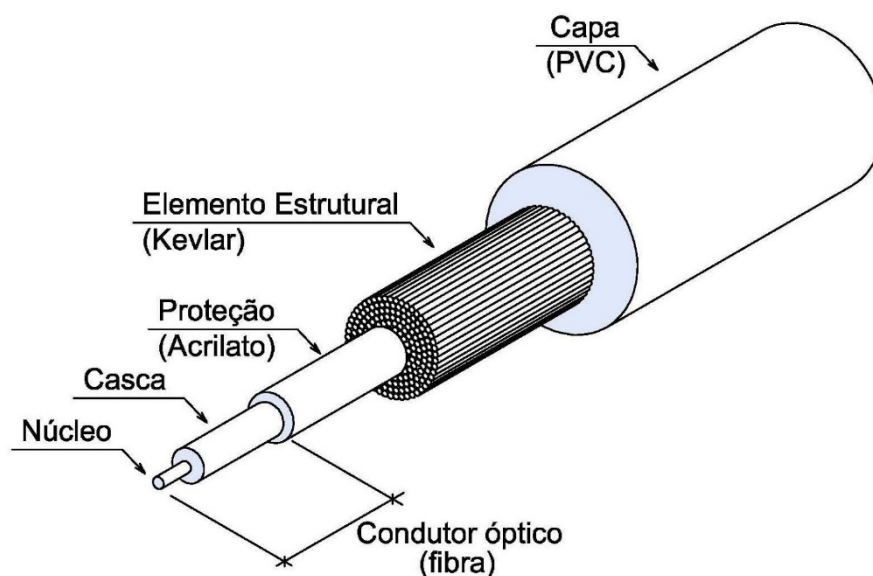
Fonte: O autor baseado em (JACINTO, 2013)

A tecnologia que popularizou nos primórdios do acesso residencial foi o modem discado, dispositivo que faz a modulação e demodulação das informações binárias através de uma linha telefônica analógica. Esta utiliza fios metálicos de acesso da rede de telefone residencial. A tecnologia conhecida como banda larga, trouxe mais velocidade à transmissão e tirou a necessidade de dedicar a linha telefônica durante a transmissão.

Existiam, basicamente, dois tipos de tecnologia de acesso residencial por banda larga: linha digital de assinante (*digital subscriber line* - DSL) e cabo híbrido coaxial (*hybrid fiber-coaxial cable* - HFC) (FARIAS, [S.d.]).

Com a popularização do uso de fibras ópticas ocorreu a evolução do seu uso não apenas em links de alta capacidade (*backbones*, redes metropolitanas) mas chegou até a casa dos clientes. Basicamente a fibra óptica é composta de um núcleo (plástico ou sílica) onde o sinal óptico é transmitido pelo princípio físico da reflexão total, uma casca e o material protetor conforme apresentado na figura 5.

Figura 5- Estrutura cabo de fibra óptica de 1 fibra



Fonte: (MONTUORO, 2013)

O termo FTTH (*Fiber to the Home*) é derivado do genérico FTTx, ou *Fiber to The Anywhere*, que significa fibra até qualquer lugar. O x representa o destino final da fibra óptica. No caso do FTTH, o cabo de fibra óptica vai até a casa do cliente residencial.

A arquitetura FTTH utiliza a tecnologia PON (*Passive Optical Network*), onde elementos que formam a rede externa são todos passivos (não precisam de energia elétrica para funcionar). Os únicos elementos ativos de uma rede PON ou FTTH são os equipamentos *Optical Line Terminal* (OLT) e *Optical Network Terminal* (ONT), que ficam central de equipamentos (operadora do serviço) e na casa do assinante.

As redes PON ao longo do tempo tiveram diversas evoluções. O APON (Rede Óptica Passiva sobre modo de transferência assíncrona) foi a primeira tecnologia óptica passiva a aparecer. Baseada no protocolo *Asynchronous Transfer Mode* (ATM) e aceita pelo ITU (*International Telecommunication Union*) como norma ITU-T G.983.

O BPON (Rede Óptica passiva banda larga) é a evolução da APON. A tecnologia é assimétrica, ou seja, taxa de transmissão de download diferente do upload sendo 622 Mbps para *downstream* e 155 Mbps para *upstream* com distância máxima de cobertura de 20 km.

O EPON (Rede Óptica passiva sobre ethernet) surgiu por causa das limitações que a APON apresentava, como, baixa taxa de transmissão, falta de capacidade de transmissão de vídeo, complexidade e custos.

A principal diferença entre EPON e APON está no tamanho do comprimento que os pacotes são transmitidos. A tecnologia é simétrica com taxas de transmissão de 1,25 Gbps para *downstream* e *upstream* com distância máxima de cobertura de 20 km (ITU.T, 2003).

G-PON (Rede Óptica Passiva Gigabit) surgiu para superar a EPON e a BPON com o intuito principal de transmitir comprimentos de pacotes variáveis a taxa de *gigabit* por segundo. A tecnologia é assimétrica com taxas de transmissão de 2,5 Gbps para *downstream* e 1,25 Gbps para *upstream* com distância máxima de cobertura de 20 km. É uma tecnologia padronizada pelo ITU-T com as Recomendações ITU-T G.984 (ITU.T, 2012).

Uma das arquiteturas de rede G-PON mais populares é a arquitetura de rede FTTH. Esse tipo de arquitetura possui as seguintes categorias: serviços de banda larga simétricos e assimétricos, POTS (*Plain Old Phone Telephone Service*) e ISDN (*Integrated Service Digital Network*) (SELMANOVIC; SKALJO, 2010).

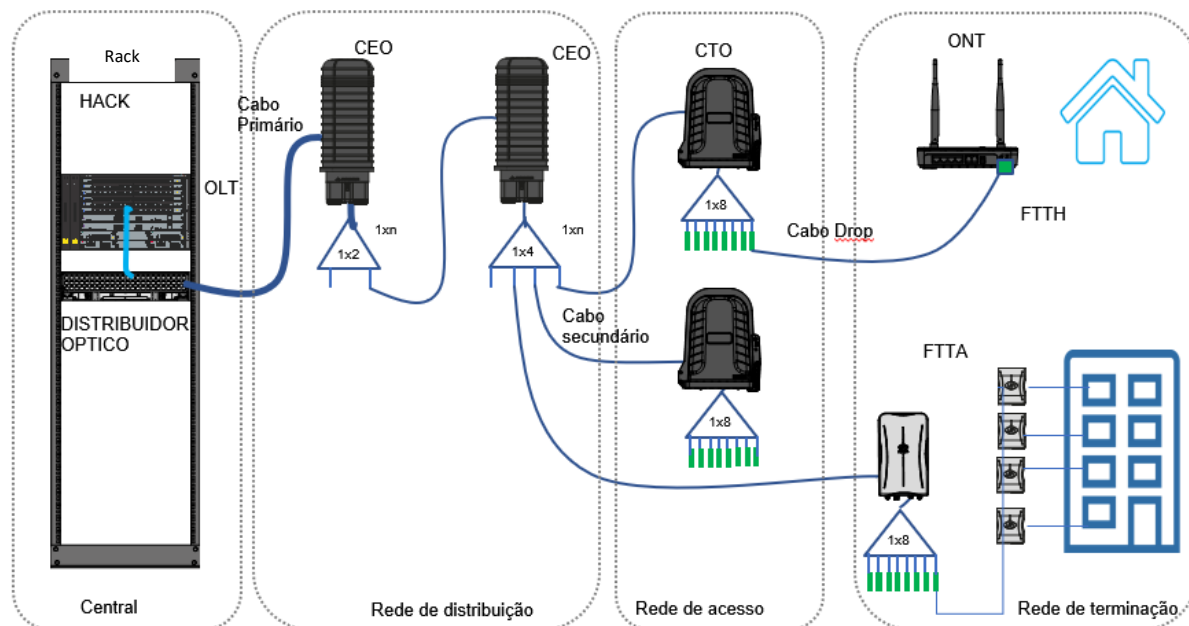
O sistema G-PON basicamente trabalha com dois componentes de transmissão ativos, são eles a OLT e ONT também chamado ONU (*Optical Network Unit*). Já os equipamentos passivos são fibras que compõem os cabos (aéreos suspensos em postes, subterrâneos lançados em dutos ou cortes em vias), *splitters* que são divisores de sinal óptico, caixas de terminação óptica (CTO) e caixas de distribuição.

As funções mais importantes que a OLT desempenha são agendamento de tráfego, controle de *buffer* e alocação de largura de banda.

A rede de distribuição óptica (ODN) é uma rede com divisores adicionais. Como o G-PON é um tipo de rede PON, ele usa uma arquitetura de distribuição em árvore. Essa arquitetura tem a seguinte estrutura: uma fibra óptica está conectada a uma conexão óptica conhecida como porta G-PON na OLT e todos os usuários estão conectados a partir dela. A ODN é definida pelas classes do G-PON; essa classe é A, B, B +, C e C + que são referências aos padrões de lasers (fonte óptica) e características do sinal.

A ONU (Unidade de Rede Óptica) é o elemento do extremo da rede G-PON (instalado no ponto de uso dos clientes como casas ou escritórios) sua classificação depende da classe de laser, segundo a norma na classe B + são ramificados através da ODN até 64 clientes em uma porta PON (porta da OLT). Em cada extremidade do G-PON pode haver mais de uma porta (UNI - *User Network Interface*) nesse caso a ONU é chamada ONT (SELMANOVIC; SKALJO, 2010).

Figura 6- Diagrama de um projeto GPON



Fonte: O autor baseado em (eFurukawa, 2019)

A figura 6 representa um diagrama de arquitetura FTTx GPON. Na central é instalado numa estrutura chamada *rack* as fontes de alimentação, a OLT e demais equipamentos de conexão. Da porta PON é conectado o cordão óptico ao distribuidor interno óptico (DIO) ou

distribuidor geral óptico (DGO). A partir do DGO as fibras saem da central agrupadas em cabos chamados cabos primários, compostos por centenas de fibras. Esse cabo primário é então ramificado através de divisores ópticos (*splitter*) localizados em determinados pontos da rede de distribuição. Os divisores ópticos ficam alojados em caixas herméticas que também servem como pontos de emenda, também conhecidas como caixas de emenda óptica (CEO).

Ali é feito o primeiro nível de *splitters*, ramificando o sinal em um primeiro nível, e da mesma forma é feito no segundo nível através da rede secundária até a caixa de terminação óptica (CTO) podendo chegar até 128 portas de conexão (ITU.T, 2012)

A CTO é o ponto onde é conectado o cabo *drop* (que interliga a propriedade do assinante até o último nível de rede ou CTO). Esta conexão pode ser feita utilizando máquinas de fusão de fibra, conectores (*fast connectors*) ou cabos pré-conectorizados de fábrica.

Quanto a normatização de padrões técnicos é seguida as normas internacionais do ITU. A família de recomendações ITU-T G.984 é dividida em: G.984.1 características gerais G.984.2 especificações da camada do PMD (*Physical Media Dependent*). G.984.3 especificações da camada de convergência da transmissão. G.984.4 especificações da interface de controle e gerenciamento da ONU (OMCI) G.984.5 otimizações de banda. G.984.6 extensões de alcance. G.984.7 longo alcance.

Estas normas ditam as principais características e especificações do G-PON. As taxas de transmissão definidas são: *Downstream*: 2488,32 Mbps operando em 1490 nm; *Upstream*: 1244,16 Mbps operando em 1310 nm.

2.1.5 Qualidade e manutenção de Redes

No Brasil, a qualidade do serviço de acesso à Internet é medida pela Agência Nacional de Telecomunicações (ANATEL) através de indicadores tais como: velocidade, latência, perda de pacotes, capacidade de resolução dos problemas do cliente e tempo no atendimento. De acordo com a resolução número 574 da ANATEL, as prestadoras de Serviço de Comunicação Multimídia (SCM) são obrigadas a contratar uma Entidade Aferidora da Qualidade (EAQ) para que possam aferidos seus indicadores de qualidade de rede e atendimento. Após essa coleta, os dados são fornecidos à ANATEL para que se possa controlar o nível de qualidade das empresas na prestação de serviço de Internet no país (SILVA, 2020).

Em 2019, foram registradas 2,98 milhões de reclamações contra as prestadoras de telecomunicações. Em relação a 2018, isso significa crescimento da ordem de 1,28%, ou 37,5 mil reclamações. Esse é um dos volumes mais baixos dos últimos anos e representa queda de aproximadamente 27% em relação ao ano de 2015.

O maior aumento em volume de reclamações foi registrado no serviço de banda larga fixa, com quase 80 mil reclamações a mais do que em 2018, ou aumento de 15,7%. Em média, no ano de 2019, para esse serviço, foram registradas 1,50 reclamação por mês para cada mil acessos em serviço, enquanto, em 2018, esse índice foi de 1,37 (SILVA, 2020). Esses dados fazem parte dos relatórios de medição e acompanhamento da Agência Nacional de Telecomunicações (ANATEL, 2019) apresentados na tabela 4.

Tabela 4- Volume e Índice de Reclamações por serviço (2015-2019)

Serviço		2015	2016	2017	2018	2019	2019-2018
Banda larga fixa	Volume	634.514	580.847	525.102	501.979	580.68	15,68%
	Índice	2,39	2,06	1,81	1,37	1,50	0,13 p

Fonte: (ANATEL, 2019)

A redução do índice pode estar diretamente relacionada a maior concorrência e pela evolução dos acessos banda larga em tecnologias ópticas, que são imunes a interferências eletromagnéticas conforme (PINHEIRO, 2016) e (KEISER, 2013).

Mesmo utilizando meios ópticos de transmissão é preciso que as operadoras de telecomunicações mantenham boas práticas de manutenção em suas redes.

Dentre as boas práticas podemos citar o uso de tecnologias que sejam homologadas pela ANATEL, que as implantações de rede sejam feitas por equipes treinadas, preparadas e que se tenha uma boa gestão de capacidade e incidentes (rompimentos, falhas massivas, atenuações na rede de acesso passiva).

A capacidade determina em grande medida a disponibilidade do serviço e, portanto, impacta na qualidade de serviço. Com o crescimento do uso, esta necessidade pode ser agravada se os usuários não forem capazes de estabelecer conexões na velocidade que eles exigem, afetando assim a experiência de uso.

Ambas as situações refletem na receita financeira de uma operadora. A medida que clientes mais satisfeitos atrairão mais usuários, clientes com percepção ruim atuam como

detratores em relação a outros clientes quanto a uma determinada empresa e seus serviços (ARAGON-ZAVALA, 2017).

Por este motivo as operadoras mantêm equipes de prontidão em estruturas chamadas *Network Operations Center* (NOC), equipes de manutenção de campo e engenharia de clientes. Em todos estes cenários é importante o apoio de ferramentas de medição de tráfego, índices de reclamação, alarmes de gerência de equipamentos e utilizar os recursos disponíveis para investigação e atuar de forma preditiva seja em equipamentos, configurações de rede ou mesmo corrigir falhas de campo.

2.2 Inteligência Artificial

2.2.1 Redes Neurais Artificiais

A inteligência artificial (IA) é sem dúvidas uma das mais emblemáticas conquistas alcançadas pela humanidade. As pesquisas são bastante diversificadas, diferenciando conforme a abordagem dos aspectos da inteligência (MEDEIROS, 2018).

Segundo Haykin (2017) o objetivo da IA é o desenvolvimento de paradigmas ou algoritmos que requeiram máquinas para realizar tarefas cognitivas, para os quais humanos são melhores e possuem, três componentes fundamentais: representação, raciocínio e aprendizagem.

As redes neurais artificiais (RNA) são um marco da IA. Seu funcionamento possibilita o aprendizado de padrões em situações muitas vezes complexas a partir de elementos simples que simulam o comportamento de neurônios (MEDEIROS, 2018).

As redes neurais artificiais são modelos computacionais que se baseiam no sistema nervoso para realização do processamento de informações em unidades denominadas neurônios artificiais (SILVA; SPATTI; FLAUZINO, 2010).

Usando algoritmos, elas podem reconhecer padrões escondidos e correlações em dados brutos, agrupá-los e classificá-los, e com o tempo, aprender e melhorar continuamente.

As características envolvidas numa rede neural artificial são a adaptação por experiência, adquirida por sucessivos exemplos de amostras com obtenção de aquisição de conhecimento; capacidade de aprendizado por método de treinamento; generalização do conhecimento; organização de dados; tolerância a falhas; armazenamento distribuído e facilidade de prototipagem (SILVA; SPATTI; FLAUZINO, 2010).

Em certas aplicações de RNAs somente o conjunto de padrões de entradas está disponível. Porém, as amostras de entrada podem possuir informações relevantes do comportamento do sistema. Essas redes se auto-organizam usando métodos de treinamento competitivo, detectando similaridades e correlações montando agrupamentos ou clusters.

Uma rede neural é um processador maciçamente distribuído constituído de unidades de processamento simples, que tem a propensão natural de armazenar conhecimento experimental e torná-lo disponível para uso (HAYKIN, 2017, p.28).

O trabalho de modelagem do sistema nervoso começou com o trabalho de McCulloch e Pitts (MCCULLOCH; PITTS, 1943) e após este trabalho muitos cientistas se interessaram pelo campo de pesquisa. Donald Hebb descreve no livro *Organization of Behavior* de 1949 o processo de treinamento e funcionamento da sinapse. Baseado neste estudo Frank Rosenblatt apresentou o modelo de sinapse humana (ROSENBLATT, 1958).

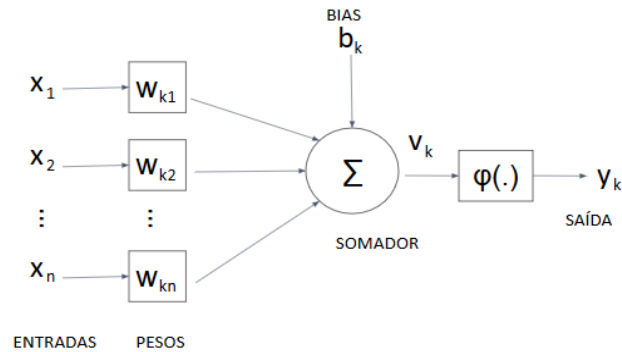
Em 1969, Minsk e Papert demonstraram que o modelo de Rosenblatt não tinha capacidade para resolver problemas linearmente separáveis (solução obtida dividindo o espaço de entrada em duas regiões por uma reta) (MINSKY; PAPERT, 1969). Mais tarde David E. Rumelhart e James L. McClelland, apresentaram um trabalho com um modelo computacional que permitia o treinamento dos neurônios artificiais, reacendendo o interesse no assunto e contornando os problemas apontados por Minsk & Paper (FREITAS; FREITAS, 2020).

O cérebro humano é constituído por bilhões de unidades denominadas neurônios constituídos de um corpo celular, denominado soma onde fica localizado o núcleo. No corpo da célula partem filamentos chamados dendritos e o filamento mais prolongado chama-se axônio. Os dendritos permitem a ligação do neurônio as células que o circundam, o axônio conecta as células que estão mais distantes, sendo estas conexões com outras células denominadas sinapses (SILVA; SPATTI; FLAUZINO, 2010).

O impulso nervoso é transmitido entre neurônios por reações eletroquímicas, lançadas pelas sinapses e entram pelos dendritos, aumentando ou diminuindo o potencial elétrico do corpo celular. Esse potencial chega a um limiar e então um pulso elétrico ou potencial de ativação é transmitido pelo axônio se espalhando ao longo das conexões do axônio, chegando a outras sinapses e reativando o mesmo processo em outros neurônios.

O modelo precursor de um neurônio artificial baseado no modelo biológico foi feito por Warren McCulloch e Walter Pitts no ano de 1943, representado na figura 7.

Figura 7- Neurônio artificial



Fonte: O autor baseado em HAYKIN (2017,p.36)

O comportamento das sinapses é emulado através de pesos w_{k1} , w_{k2} até w_{kn} acoplados as entradas dos neurônios, cujo valor pode ser excitatório ou inibitório. O efeito de uma sinapse i no neurônio pós-sináptico é dado por x_{ki} multiplicado por w_{ki} .

As entradas podem também receber as saídas de outros neurônios, outras entradas externas, um bias ou qualquer combinação destes elementos. O bias (Tendência) é um erro sistemático diferentemente de erro aleatório. Um ou mais componentes do erro sistemático podem contribuir para a tendência. Uma grande diferença sistemática em relação ao valor de referência aceito reflete-se em um grande valor de tendência. (“O Neurônio Artificial”, [S.d.]

A função de ativação é dada pelo limiar representado por φ (1) função degrau e outras representadas pela figura 8.

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (1)$$

Figura 8- Gráficos de funções de ativações



Fonte: O autor (2020)

Segundo Haykin (2017), conforme figura 8 tem-se a função limiar, função linear (rampa), função sigmoide para parâmetro de inclinação a variável. Outro elemento presente é o campo local induzido pelo neurônio, representado por v_k :

$$v_k = \sum_{j=1}^N w_{kj}x_j + b_k \quad (2)$$

No somatório w_{kj} é um número real que representa a conexão sináptica da entrada j -ésimo neurônio. A conexão sináptica é conhecida como excitatória se $w_{kj}>0$ ou inibitória caso $w_{kj}<0$. Após a determinação do somatório, o valor da ativação do neurônio é atualizado através da função de ativação e finalmente, o valor de saída do neurônio é produzido através da função de saída f_s .

Haykin (2007) reconhece seis propriedades que caracterizam uma RNA segundo Medeiros:

1. Não Linearidade - Os neurônios podem ser lineares ou não lineares, permitindo aproximações de funções robustas de funções de mapeamento que tenham característica não linear. **2. Mapeamento de entrada-saída** - A rede aprende a partir de exemplos, estabelecendo mapeamento entre os padrões apresentados na entrada com saídas dadas pelos exemplos. **3. Adaptabilidade** – Redes neurais podem ser treinadas e armazenar o conhecimento nos pesos sinápticos, podendo adaptar-se caso o conjunto de amostras utilizado para o treinamento se modifique ao longo do tempo. **4. Resposta a evidências** - Uma rede neural pode fazer uma tarefa de seleção de um padrão, mas também informar o grau de confiança ou crença no padrão escolhido. **5. Informação contextual** - o conhecimento é armazenado na própria estrutura e pela avaliação da rede neural. **6. Tolerância a falhas:** Redes implementadas em *hardware* são tolerantes a falhas, em caso de neurônios ou conexões que possam ser danificados, ou mesmo em *softwares* que utilizem técnicas de poda de redes que reduzem a quantidade de neurônios ou sinapses, mantendo a mesma condição de performance. (MEDEIROS, 2018, p. 129).

Ainda, Medeiros (2018) apresenta o conteúdo quanto a construção de RNAs os elementos básicos são apresentados por Haykin (2007).

1. Número de camadas -As RNAs têm pelo menos 1 camada de entrada, de onde recebem os sinais ou as características das amostras; e uma camada de saída, que apresenta os padrões ou as classes mapeadas para os conjuntos de treinamentos. Também podem apresentar uma ou mais camadas ocultas, como no caso do perceptron multicamada e das redes de base radial. **2. Quantidade de neurônios por camada** – A quantidade de neurônios depende da natureza do problema abordado. A camada de entrada terá tantos neurônios quantas forem as características das amostras do conjunto de treinamento. A camada de saída terá os neurônios referentes às classes a que pertencem as amostras do conjunto de treinamento. A camada oculta ou escondida, por seu turno, pode ter uma quantidade variável de neurônios, conforme a característica do mapeamento que se deseja. **3. Tipo de função de transferência** – A função transferência define a forma como o neurônio é ativado. Para isso, podem ser

utilizadas funções discretas (como a função degrau, empregada no perceptron [...]) ou contínuas (como a função sigmoide para o perceptron multicamada). **4. Métodos de treinamento** - Ao longo dos anos, têm-se desenvolvido diversos métodos ou algoritmos de treinamento, entre os quais o mais utilizado é o algoritmo de retro propagação (*backpropagation*), que aplica informação do erro na atualização dos pesos (MEDEIROS, 2018, p. 130).

Quanto aos métodos de aprendizagem as principais são:

1. Aprendizagem mediante correção de erros – A informação do erro é utilizada para modificar os pesos sinápticos. Encontramos esse tipo de aprendizagem, por exemplo, no perceptron de Rosenblatt (1958) e no Adaline de Widrow e Hoff (1960). **2. Aprendizagem baseada em memória** – Armazena-se um grande número de exemplos de entrada e saída, e uma amostra é comparada com sua vizinhança, a fim de que se identifique a que classe pertence. **3. Aprendizagem hebbiana** – Com base nos estudos de Donald Hebb, emprega uma regra associativa, que aumenta a força dos pesos positivamente correlacionados ou diminui a daqueles negativamente correlacionados. **4. Aprendizagem competitiva** – Os neurônios competem entre si para tornarem-se ativos. O vencedor estará ativo durante certo instante (aprendizagem também denominada *winner-takes-all*). **5. Aprendizagem de Boltzmann** – Apresenta características estocásticas, derivando-se das ideias da mecânica estatística. [...] A atualização dos pesos ocorre por correlação, operando em duas condições: uma presa, [...] e outra livre [...] (MEDEIROS, 2018, p. 131).

A aprendizagem é caracterizada de duas formas, supervisionada e não supervisionada. Na supervisionada ocorre o *feedback* da rede para orientar o treinamento, realizando esta ação com uma amostra de dados de treinamento. Na não supervisionada a rede aprende de forma auto-organizada, utilizando os próprios dados de entrada para extrair características e correlações.

Dentre as possibilidades de RNA estão reconhecimento de padrões, aproximação de funções, controle, filtragem (SILVA; SPATTI; FLAUZINO, 2010).

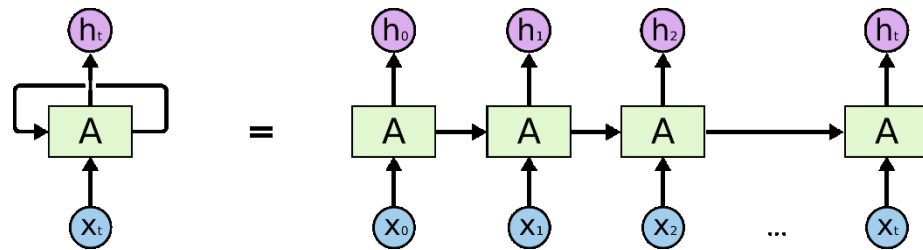
2.2.2 Redes Neurais Recorrentes e LSTM

Redes neurais recorrentes (RNN) são redes neurais com *loops*. Essas estruturas de processamento representam uma grande variedade de comportamentos dinâmicos. Permite a criação de representações internas e dispositivos de memória capazes de processar e armazenar informações temporais e sinais sequenciais (OLAH, 2015).

São diferentemente das redes neurais tradicionais (que não armazenam informações no tempo) em suas estruturas. As RNNs possuem um estado interno (*h*) ou estado interno do

neurônio recorrente. Uma rede neural recorrente pode ser considerada como várias cópias da mesma rede, cada uma passando uma mensagem a um sucessor. (SOUSA, 2018).

Figura 9- Representação de camadas LSTM



Fonte: (OLAH, 2015)

A figura 9 representa um *loop* permite que as informações sejam transmitidas de uma etapa da rede para a próxima, A representa a rede neural recorrente, ou um conjunto de camadas escondidas e sobrepostas com n neurônios e combinações. Uma RNN pode ser considerada como várias cópias da mesma rede, cada uma passando uma mensagem a um sucessor. O conceito de recorrência é definido na equação 3, sendo h o novo estado após cada loop, h é o estado interno do neurônio recorrente ou da rede recorrente, a função de w é a função dos parâmetros de entrada w, h no tempo t-1 representa o estado anterior, x representa o input.

$$h_t = fw(h_{t-1}, x_t) \quad (3)$$

Para a equação 3, teremos como função de ativação *tanh*, e de acordo com a quantidade de estados ou tempos que a rede precise reconsiderar teremos n matrizes de estado interno e n matrizes de inputs que irão compor os novos estados internos, formando assim um gradiente para o treinamento da rede em retropropagação (SOUSA, 2018).

As RNNs têm a forma de uma cadeia de módulos repetidos e esse módulo de repetição terá uma estrutura simples, como uma única camada de tangente h.

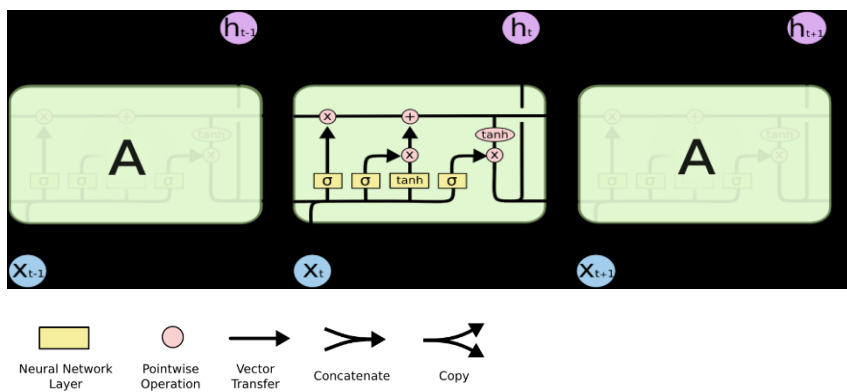
Alguns problemas dessa técnica é que aumento a quantidade de estados h, aumenta-se muito o número de parâmetros e atualizações de matrizes nas estruturas. Outro ponto é para o treinamento e atualização dos pesos (retropropagação) é necessário que seja feito a derivada parcial do erro em relação a entrada em k vezes os estados anteriores. Para esse gradiente o estado atual é corrigido e para os demais estados anteriores serão o produto das derivadas parciais de k estados, e para múltiplas saídas acontece a mesma situação. Desta forma é possível

ter treinos ruidosos, dificuldade de dependência de longo prazo e dificuldade de lidar com esses ruídos (SOUSA, 2018).

Hochreiter & Schmidhuber (1997) introduziram um conceito chamado *Long Short Term* (LSTM) ou em tradução literal redes de memória de longo prazo (HOCHREITER; SCHMIDHUBER, 1997). É um tipo especial de RNN, pois a relação entre os neurônios que estão distantes é calculada, envolvendo a multiplicação da matriz Jacobiana. É uma arquitetura de RNN capaz de manter um estado de memória.

O LSTM muda o peso no *loop* aumentando o limite de entrada, alterando limite de ajuste e limite de saída. Com isso os parâmetros podem ser alterados em momentos diferentes dinamicamente (LIU *et al.*, 2019). Na figura 10 é representado os blocos de um LSTM.

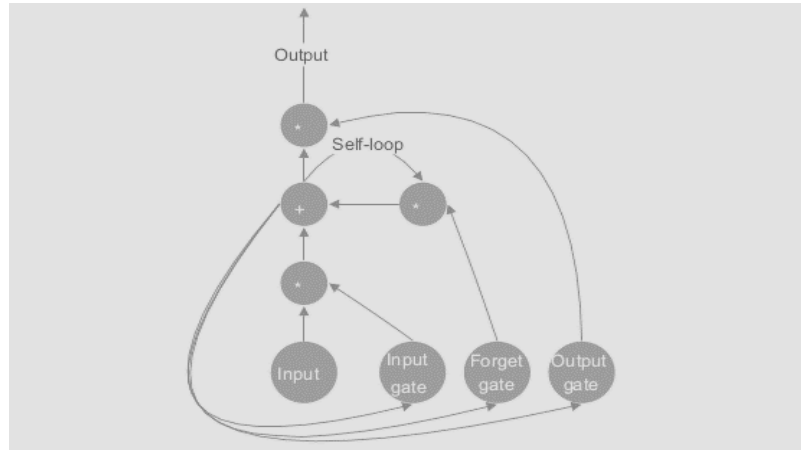
Figura 10- Representação das etapas de funcionamento de um LSTM



Fonte: (OLAH, 2015)

As LSTMs com uma camada, possuem um módulo de repetição de estrutura muito simples, como uma única camada passando pelo termo tangente h. Outra é em vez de uma única camada de rede neural, existem quatro, interagindo de uma maneira diferente. A estrutura de etapas executadas no LSTM é representada com outra visão na figura 11.

Figura 11- Representação das interações de um LSTM



Fonte: (LIU et al., 2019)

Na figura 11, cada linha carrega um vetor inteiro, desde a saída de um nó até as entradas de outros. Círculos rosa representam operações pontuais, como agregação de vetores, enquanto as caixas amarelas são camadas de rede neural aprendidas. A fusão de linhas denota concatenação, enquanto uma bifurcação de *linh* denota seu conteúdo sendo copiado e como cópias indo para locais diferentes. Esta representação será utilizada para abordar as etapas de funcionamento conforme (OLAH, 2015).

O LSTM pode remover ou adicionar informações ao estado da célula, regulado por estruturas chamadas de portas, uma forma opcional de permitir a passagem de informações. São compostos de uma camada de rede neural sigmoide (primeira rota paralela do diagrama na figura 10) e uma operação de multiplicação pontual. Resulta em números entre zero e um, descrevendo quanto de cada componente deve ser deixado passar. Um valor zero significa "não deixe nada passar", enquanto um valor um significa "deixe tudo passar" (OLAH, 2015).

Outra forma de se pensar esta ação é o quanto deve “esquecer” de informação e o quanto ela deve manter ao próximo estado (SOUSA, 2018).

Esta etapa é representada matematicamente pela função 4:

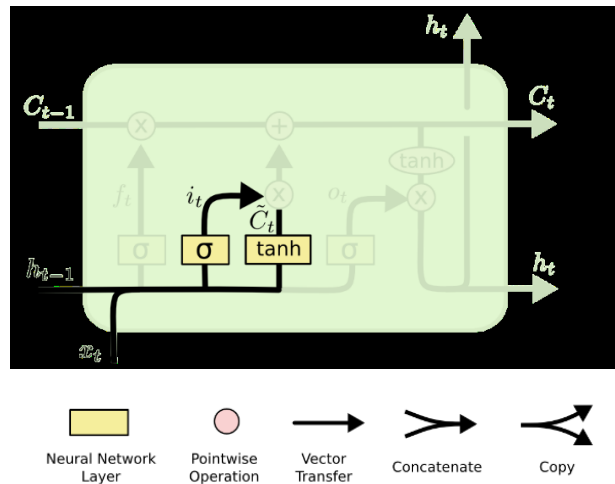
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (4)$$

O próximo passo é decidir o armazenamento de informações úteis no estado da célula, essa etapa é denominada *Input Gate*. Primeiro, uma camada sigmoide, chamada de camada de porta de entrada, decide quais valores serão atualizados. Em seguida, uma camada tangente h cria um vetor de novos valores candidatos, após será atualizado para o estado novo. Os valores

do vetor e os valores regulados são multiplicados para obter as informações úteis (OLAH, 2015).

Essa etapa decide qual informação que vem da entrada vai ser inserida. É combinada com o candidato a novo C que é \tilde{C}_t e pode ser representada conforme figura 12.

Figura 12- Segunda etapa LSTM, Input Gate



Fonte: (OLAH, 2015)

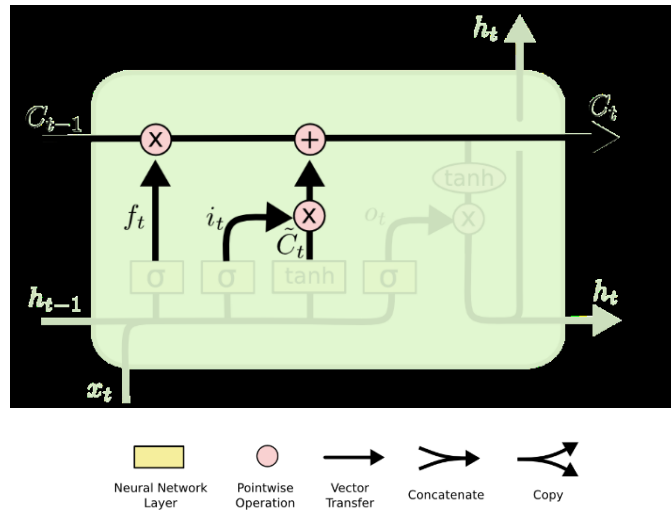
As ações podem ser representadas matematicamente conforme as equações 5 e 6.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (5)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (6)$$

A atualização então do estado da célula antiga C_{t-1} para o novo estado C_t e feita numa etapa então denominada *Cell Update* e representada pela figura 13.

Figura 13- Atualização do estado C_t , Cell Update



Fonte: (OLAH, 2015)

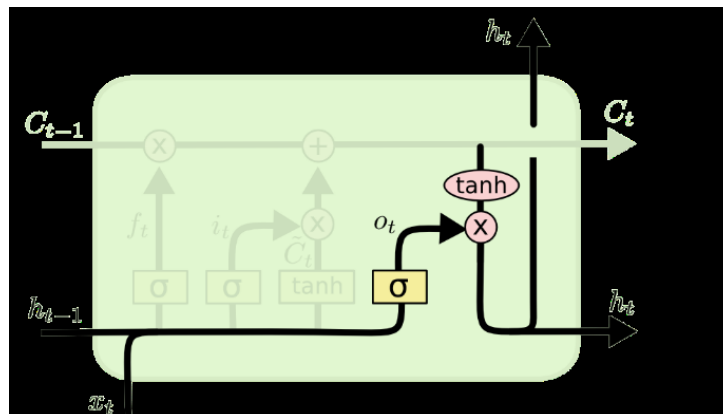
A função matemática que representa o estado C_t e dado pela equação 7.

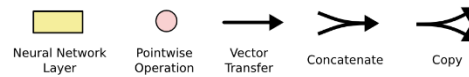
$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (7)$$

A saída será baseada no estado de célula, é executado uma camada sigmoide que define quais partes do estado da célula por meio de tangente h (ficando entre -1 e 1), esse valor então é multiplicado pela saída da porta sigmoide (OLAH, 2015).

Primeiro determina-se quais partes do *Cell State* será enviado para a saída. Então usa-se *uma tangente h para* gerar saídas que serão multiplicadas por (8) o_t conforme figura 14 (SOUSA, 2018).

Figura 14- Atualização da etapa de saída LSTM





Fonte: (OLAH, 2015)

Sua representação matemática é dada pelas equações 8 e 9.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = o_t \times \tanh(C_t) \quad (9)$$

A rede neural LSTM é um tipo rede neural cíclica especial. É adequada para processamento e prever eventos importantes com intervalo relativamente longo e atraso na sequência. Serve para fazer uso das características de eventos no passado para prever as características futuras. (LIU *et al.*, 2019)

Outros trabalhos que também complementam o embasamento prático na área de aplicação de RNN em situações de tecnologia da informação e comunicação tem-se (RUSEK *et al.*, 2020); (HUANG; CHUNG; TANG, 2018); em (MIRANDA JÚNIOR *et al.*, 2019) e (TELES VIEIRA; LEMOS; LEE, 2003).

2.2.3 Self-Organizing Maps

Em certas aplicações de RNAs somente o conjunto de padrões de entradas está disponível, ou seja, não existem dados para treinamentos sucessivos. Porém, as amostras de entrada podem possuir informações relevantes do comportamento do sistema. Essas redes se auto-organizam usando métodos de treinamento competitivo, detectando similaridades e correlações montando agrupamentos ou *clusters*.

Dentre as estruturas de redes artificiais auto-organizáveis destaca-se o trabalho do finlandês Teuvo Kohonen (1984). Essas redes foram inspiradas no córtex cerebral, que é ativa por frente a determinados estímulos (SILVA; SPATTI; FLAUZINO, 2010).

O aprendizado de padrões que possam levar a análise de dados clusterizados possuem diversas aplicações, sejam acadêmicas quanto para análises financeiras, reconhecimento de padrões, análises de falhas e centenas de outras aplicações.

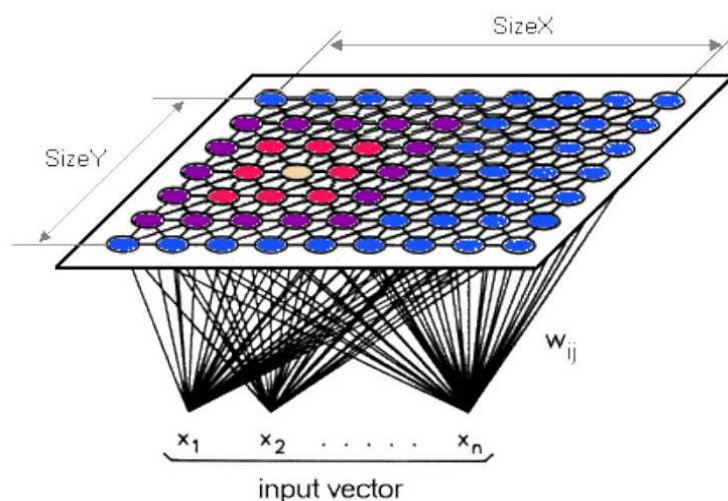
As redes *self-organizing maps* (SOM), são mais semelhantes as estruturas neurobiológicas que as redes supervisionadas, sua estrutura básica é uma camada de entrada,

fonte, camada de saída, mas também possuem modelos multicamadas (BRAGA; LUDEMIR; CARVALHO, 2014).

Seu funcionamento é que quando um padrão de entrada é apresentado, a rede procura a unidade mais parecida com esta entrada (demais dados). Durante o treinamento a rede aumenta a semelhança entre os nós escolhidos. Assim a rede constrói um mapa topológico onde os nós estão topologicamente próximos e são semelhantes quanto seus padrões de entrada (KOHONEN, 1990; SHARMA; DWIVEDI, 2018).

Os nós da rede são organizados em grade bidimensional, as vezes unidimensional, organizados em linhas e colunas. Cada nó recebe todas as entradas e funciona como um extrator de características onde quanto maior a semelhança dos dados com o vetor de pesos desse nó, maior o valor de sua saída. Um exemplo de sua estrutura seria a figura 15.

Figura 15- Representação do mapa de Kohonen e seus vetores de entrada



Fonte: (CIABURRO; VENKATE, 2020)

Do ponto de vista de aprendizagem, os neurônios de saída da RNA competem entre si para se tornar ativos, assim somente um neurônio de saída está ativo em um determinado instante. Neurônios com mesma estrutura, mas pesos diferentes têm respostas diferentes a uma entrada. Um limite é imposto sobre a força de cada neurônio e existe um mecanismo de competição entre neurônios, de forma que um neurônio é vencedor em um dado instante (CAMBUIM, 2017).

Outra característica é que seu aprendizado é não supervisionado, ou seja, não requer previamente as saídas desejadas. Como fonte de dados de seu treinamento são utilizados apenas seus valores de entrada. A rede trabalha esses dados classificando-as (MORAL, 2008).

Seus pesos iniciais são aleatórios, o que pode gerar muitos ciclos lentos, podendo então para isso ser inicializado com vetores de pesos iguais e modificação dinâmica de raio de vizinhança. O treinamento é competitivo e não supervisionado. Seus pesos são atualizados pela equação:

$$\begin{aligned} \omega_{ij}(t+1) &= \{w_{ij}(t) + \eta(t)(x_i(t) - \omega_{ji}(t)), \text{ se } j \\ &\in \Lambda(t) \omega_{ij}(t), \text{ caso contrário} \end{aligned} \quad (10)$$

O funcionamento SOM possui etapas distintas, etapa competitiva define o neurônio mais adequado (*Best Matching Unit*). A escolha da melhor correspondência entre o vetor de entrada e o vetor peso é feita por meio do critério da menor distância (euclidiana) entre o vetor de pesos por ela armazenado e o vetor de entrada conforme (AFFONSO, 2011).

$$i(x) = \|x - w_j\| \quad j = 1, 2, \dots, n \quad (11)$$

Onde $i(x)$ é a representação do neurônio da entrada x , e w_j é o vetor peso. Das funções de distâncias uma das mais empregadas é a distância Euclidiana (AFFONSO, 2011).

$$D_E = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (12)$$

Aqui x_n são as coordenadas dos vetores de entrada e y_n são as coordenadas dos vetores-protótipo (pesos das redes auto-organizáveis). Outros tipos de distâncias são as de *Minkowski*, e distância de *Manhattan*.

$$D_{Minkowski} = \sqrt[p]{\sum_{k=0}^n |x_k - y_k|^p} \quad (13)$$

$$D_{Manhattan} = \sum |X - Y| \quad (14)$$

Na etapa cooperativa, são definidos os vizinhos dentro de uma distância obtida a partir da BMU (*Best Matching Unit*) obtida na primeira etapa, competitiva.

O treinamento da rede, consiste na otimização da distância entre os neurônios. Um neurônio ativado tende a excitar os neurônios em sua vizinhança imediata. A atribuição de novos valores e distâncias abrangendo toda a rede é chamada de época. Os neurônios nessa vizinhança são atualizados a cada iteração (AFFONSO, 2011) .

Na aplicação de SOM pode-se citar outros trabalhos de embasamento por (SHARMA; SHARMA; DWIVEDI, 2018); (PILLAI *et al.*, 2019); (AFFONSO, 2011).

3 METODOLOGIA DA APLICAÇÃO

Como metodologia de desenvolvimento deste trabalho, foram aplicados os conceitos de RNNs e SOM. Como aplicação foram utilizados para automatizar a análise de dois casos reais de uma operadora de telecomunicações.

Esses dois modelos foram escolhidos por algumas características de seus conceitos teóricos e de aplicação e por serem aplicações reais e de praxe de análise por especialistas da área de engenharia, operações e planejamento de uma operadora. Cada uma apresenta um determinado modelo de relatório de coleta de dados, sendo que estes dados se assemelham as características descritas na bibliografia e de trabalhos aplicados nos casos do SOM e de LSTM (EREMENKO; PONTEVES, 2018). Também são situações levantadas por entrevista com especialistas das respectivas áreas e que foram indicadas para o desenvolvimento de pesquisa de automações para os casos.

O SOM permite extrair as características de uma base e possivelmente encontrar padrões anormais. Essa funcionalidade propicia a análise de inconsistências, desvios e padrão de comportamentos. Outra característica é que utiliza treinamento não supervisionado, ou seja, não é necessário fornecer uma base de treinamento ao algoritmo. Pode então ser aplicado em casos que não se tenha uma base treinamento. Além disso se torna uma ferramenta de análise exploratória dos dados.

O algoritmo não supervisionado, por uma definição simplista, entende os padrões nos dados e gera grupos (clusters). Esses clusters são dados que possuem similaridades entre si, (características que auxiliam o algoritmo a classificá-los). Por estes motivos e pelo exemplo aplicado (EREMENKO; PONTEVES, 2018) foi utilizado na análise de falhas de massivas de redes de acesso.

Falhas de rede são problemas indesejáveis. Mas uma vez ocorrido deve ser tratado de forma ágil e estudado para que as equipes de operação de rede possam evitá-las futuramente e agir de forma preventiva.

Na segunda situação de aplicação será utilizado RNNs. Estas diferentemente do SOM são redes neurais com aprendizado supervisionado. É preciso fornecer uma base de dados ao algoritmo para que seja feito o treinamento do aprendizado.

O macro detalhamento das aplicações testadas estão na tabela 5:

Tabela 5- Grupos de testes executados

GRUPO	Teoria de IA aplicada	Situação a ser aplicado	Objetivo
1	SOM	Falhas de campo em GPON	Análise de falha através de padrões dos dados
2	RNN	Monitoramento de interface de BNG/BRAS	Predição futura de tráfego através do aprendizado pela base de treinamento histórica.

Fonte: O autor (2020).

Uma característica importante das RNNs é a possibilidade de utilização com dados temporais, ou seja, que variam de acordo com tempo. Por este motivo são utilizadas em diversas aplicações como previsão de valores de ações da bolsa de valores. Essa característica as tornam um bom caso de aplicações em telecomunicações como previsão de tráfego.

O tráfego da rede varia de acordo com horários e dias, além de ser influenciado por fatores externos a rede, como o comportamento dos usuários, falhas, aplicações e outras das mais diversas. Mas nessas situações os especialistas possuem ferramentas baseadas em históricos de utilização e cálculos de crescimento estatístico para tentar prever o futuro comportamento do tráfego e atuar também de forma preditiva.

Desta forma foi aplicado um código de RNN com o intuito de tentar fazer a previsão e avaliar a qualidade da aplicabilidade desta teoria auxiliando na automatização de tarefas. Espera-se chegar a uma previsibilidade de comportamento do tráfego, coletado em relatórios, destinando parte dos dados como treinamento e parte como teste.

3.1 Levantamento dos dados reais para aplicação em algoritmo SOM

Para a aplicação dos conceitos desenvolvidos neste trabalho foram utilizados dados reais cedidos em parceria com uma empresa de telecomunicações, que está presente em mais de 350 cidades e que possui as mais diversas tecnologias de rede, telefonia e *datacenters*.

O conteúdo dos dados levantados não apresenta informações das quais possam trazer riscos a operação da empresa como IPs dos equipamentos, localização, dados pessoais ou que de qualquer forma venham a colocar usuários em risco ou em desacordo da Lei Geral de Proteção de dados (LGPD) (SERPRO, 2020).

O relatório real de falha massiva em rede GPON é gerado por especialistas de operações de rede com a coleta de dados de sistemas, medições de campo e concatenados a fim de descrever as causas de falha massiva (especificamente em clientes banda larga de tecnologia de acesso GPON). Este relatório conta com as classificações de cada equipamento final (cliente) que estava numa determinada área que sofreu uma falha massiva (falha de funcionamento da rede que afeta uma grande parte da base ou região) e as características coletadas de rede em campo.

Seu principal objetivo é ser fonte de informação para a investigação das causas, se tornar histórico para análises futuras e fornecer indicadores para as equipes de incidentes, engenharia e planejamento.

O relatório de análise original consiste em uma tabela que pode ser em banco de dados, ou planilha (como utilizado neste trabalho). Contendo os seguintes campos:

- ONU *name*: equipamento final.
- Tecnologia: fabricante da OLT e ONU.
- ONU *Serial number*: serial da ONU.
- OLT RX *power*: nível de potência óptica na porta da OLT.
- ONU RX *power*: nível de potência óptica na ONU.
- ONU *distance*: referência de distância da ONU na rede.
- Circuito: número do circuito lógico cadastrado a ONU.
- CTO: caixa de terminação óptica cadastrada vinculada ao cliente.
- Porta: porta utilizada na CTO.
- Banda média: última referência de banda consumida.
- Quantidade de CTOs: número de CTOs conectadas a mesma porta da OLT.
- Quantidade de Portas: número de portas totais de acesso conectadas.
- Massivas: sim ou não (se o equipamento terminal esteve em algum evento de falha).
- *Region Name*: região geográfica ou localidade, neste caso 3 regiões metropolitanas.
- *Device name*: nome da OLT concentradora principal.

A elaboração original deste relatório é feita pelos especialistas de operação de campo e NOC.

Para a aplicação no algoritmo som foram inicialmente desconsiderados valores com erros de digitação ou coleta, dados inconsistentes e valores em branco.

O formato de exportação dos dados foi através de relatório no formato xls e csv fornecidos pelo especialista que colaboraram com este trabalho. Foram selecionados 8 atributos com o intuito de diminuir a dimensionalidade de dados, preservar informações das quais tenham sigilo e que não tenham relevância de para a análise conforme com a visão do especialista.

O indexador é a coluna que se torna a referência para o *dataset*, neste caso o campo *Onu Client*, que são os equipamentos finais dos clientes e determinam cada linha do relatório. Para os demais atributos de análise foram selecionados respectivamente conforme tabela 6.

Tabela 6- Tipos e quantidades de dados reais utilizados por categoria

Onu Client	Region Name	Device Name	Tecnologia	OLT RX Power	ONT RX Power	ONU distance	Circuito
Indexador	Atributo	Atributo	Atributo	Atributo	Atributo	Atributo	Atributo
767	3	30	1	767	767	767	767

Fonte: O autor (2020).

Para os atributos *ONU CLIENT*, *Region Name*, *Device Name* e *Tecnologia* foram substituídos os valores originais por números de 1 a n conforme tabela 6. Esse relatório então após o tratamento ficou com 767 linhas por 8 colunas, se tornando a base de dados para o algoritmo SOM.

3.2 Ambiente de desenvolvimento SOM

Para o desenvolvimento da aplicação usando algoritmo SOM foi utilizado como hardware um notebook HP G42-215br com processador Core-i3, 2Gb de memória RAM, SSD 128 Gb com sistema operacional Ubuntu 20.04 versões 64 bits. Conexão com internet banda larga GPON 50 mbps.

Como desenvolvimento foi utilizado a linguagem Python na versão 3.7 através da plataforma Anaconda. Dentre as opções da plataforma Anaconda, foi utilizado o compilador *Spyder* para desenvolvimento e testes.

Também foi utilizado o ambiente *Google Colaboratory* comumente também chamado de *Google Colab*, ou apenas *Colab*. É um ambiente de *notebooks* Jupyter que não requer configuração e é executado na nuvem (“Google Colab | Plataforma de Ciência de Dados aplicada à Saúde”, [s.d.]).

Como o ambiente é aberto através do *browser* conectado via internet direto as máquinas do Google, não é preciso realizar configurações especiais. Assim se torna uma opção para aplicações que necessitem de mais suporte de hardware, simplifica a aplicação teórica e abre oportunidade a pesquisadores que não tenham acesso a ferramentas avançadas de processamento. O *browser* utilizado foi o Google Chrome 84.0.4147.105 (Versão oficial) 64 *bits*.

Para o desenvolvimento do código foram utilizadas referências do código aplicado em (EREMENKO; PONTEVES, 2018). As principais bibliotecas utilizadas para o desenvolvimento foram a biblioteca *numpy* que é usada principalmente para realizar cálculos em *Arrays* Multidimensionais (“NumPy”, 2015).

Biblioteca *pandas* (“Pandas Library”, 2020), biblioteca de código aberto, que fornece estruturas de dados, onde trabalha-se com tabelas usando estruturas chamadas *datasets*.

O *scikit-learn*, é uma biblioteca de aprendizado de máquina de código aberto que suporta aprendizado supervisionado e não supervisionado. Fornece ferramentas para ajuste de modelo, pré-processamento de dados, seleção e avaliação de modelo (“Scikit-learn”, 2020).

Foi utilizado a biblioteca MiniSOM 2.2.6, para implementação do algoritmo SOM (“MiniSom PyPI”, 2020) , a biblioteca *pylab* foi utilizada para geração de gráficos 2D (“Pyplot”, 2020) .

3.3 Aplicação *Self-Organizing Maps*

O código aplicado para clusterização de informações de falha de rede GPON está publicado no Anexo A deste trabalho e as bases e código digital no link público também no mesmo anexo.

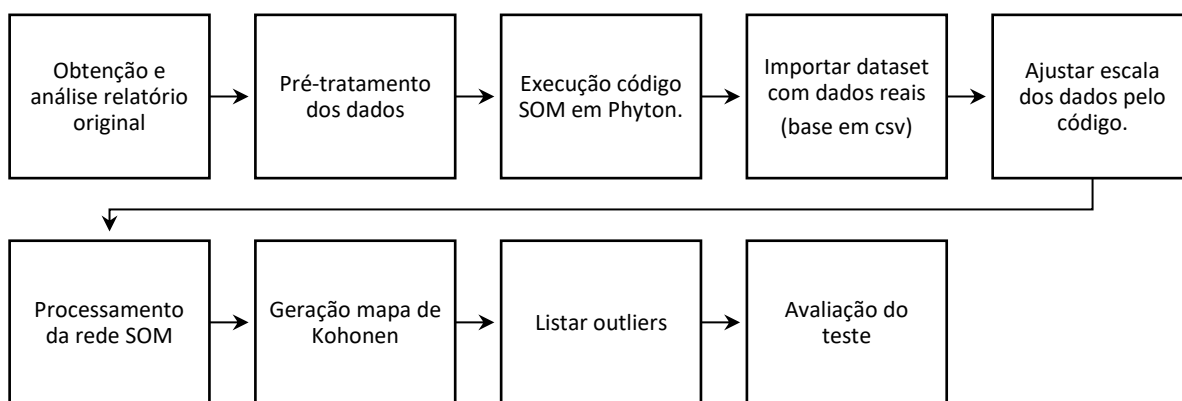
Após o recebimento dos dados foi realizado o estudo e validação da base, retirando os valores nulos, valores com erros de digitação e padronização (Pré-tratamento dos dados).

O Pré-tratamento também consiste em retirar dados sensíveis (Nomes, ou dados que possam expor clientes ou configurar algum risco). Também a exclusão de linhas em branco ou colunas vazias. Esta etapa é executada em ferramenta *Microsoft Excel*, ou outra ferramenta livre de leitura de planilhas. Após a finalização é salvo o arquivo em formato *csv*.

Uma vez com os dados de base prontos e salvos, foi desenvolvido o código (Anexo A), carregado as bibliotecas necessárias (Execução do código SOM em Python) e apontando os diretórios de salvamento e busca da base de dados.

Para leitura dos dados pelo código SOM, foi executada a importação do arquivo em formato *csv* (*Comma-separated values*). Após a importação é realizado o processamento e geração de resultado, que conforme o algoritmo da biblioteca MiniSom é realizado (em execução do código) um ajuste de escala dos dados, a execução do algoritmo SOM e a geração dos grupos de similaridade (clusters) conforme a figura 16 a seguir:

Figura 16- Etapas de execução do teste usando SOM



Fonte: O autor (2020).

Neste trabalho utilizando a ferramenta *Google Colab* foi possível executar linha a linha e observar as etapas do código. Inicialmente deve-se importar as bibliotecas de apoio e código de importação da base em *csv*, indicando o caminho do arquivo a ser lido pelo código e liberando o acesso a drives (caso arquivo esteja em nuvem).

Já com o código em execução e com base de dados inserido, na etapa de treinamento da rede SOM é feito a análise de entrada dos dados pelo algoritmo, calculado os vetores de distância euclidiana, e após construído um mapa de aproximação dos dados.

Nessa aplicação após a geração dos mapas será observado um mapa de 2 dimensões que possui uma escala de cores de acordo com proximidade de um valor central. Quanto mais escuro na escala de cores (tendendo ao preto) mais próximo do centro de distância dos dados. Desta forma são definidos os clusters, e sua quantidade depende das características em comum dos dados de entrada.

A fim de testar o modelo em diferentes situações foi realizado a variação de seus parâmetros: *input_len* que é a quantidade de atributos que caracterizam os dados, *sigma* que é

o raio dos centroides e *learning_rate* que é a taxa de aprendizado conforme realizado por (EREMENKO; PONTEVES, 2018). Os testes propostos foram os seguintes conforme tabela 7:

Tabela 7- Conjunto de testes aplicado usando SOM

Teste	Ação	Parâmetros	Execuções	Objetivo
1.1	Utilizar parâmetros mistos e conforme referência.	input_len = 15 sigma=1.0 learn_rate = 0.5	1	Apresentar metodologia de análise e avaliar outliers.
1.2.1	Alterar variável sigma e Learn Rate	Sigma=1/learn_rate=0,5	5	Avaliar alterações de resultado.
1.2.2	Alterar variável sigma e Learn Rate	Sigma=1/learn_rate=1	5	Avaliar alterações de resultado.
1.2.3	Alterar variável sigma e Learn Rate	Sigma=0,5/learn_rate=0,5	5	Avaliar alterações de resultado.
1.2.4	Alterar variável sigma e Learn Rate	Sigma=0,5/learn_rate=1	5	Avaliar alterações de resultado.
1.3	Alterar valores negativos para positivos	Utilizar dados em módulo. Sigma=1/learn_rate=0,5	5	Avaliar alterações de resultado.
1.4.1	Alterar para 1 atributos	Retirada atributos: Sigma=1/learn_rate=0,5	5	Avaliar alterações de resultado.
1.4.2	Alterar para 8 atributos	Retirada atributos: Sigma=1/learn_rate=0,5	5	Avaliar alterações de resultado.
1.4.3	Alterar para 4 atributos	Retirada atributos: Sigma=1/learn_rate=0,5	5	Avaliar alterações de resultado.
1.5.1	Alterar taxa de aprendizado para 2	Sigma=1/Learn_rate = 2	5	Avaliar alterações de resultado.
1.5.2	Alterar taxa de aprendizado para 5	Sigma=0,5/Learn_rate = 2	5	Avaliar alterações de resultado.

Fonte: O autor (2020).

Ao todo são 51 testes que nesta aplicação, além do mapa e a tabela dos dados, será apresentado para o item 1.1 uma lista dos principais dados discrepantes, o que indicará que a análise para o atributo de falha massiva foi ou não considerado correto pelo classificador.

Com a lista, seria possível indicar aos especialistas quais os equipamentos não foram corretamente avaliados e indicar possíveis ações de manutenção preditiva.

3.4 Levantamento dos dados reais para aplicação em algoritmo LSTM

Para os testes de aplicação usando RNNs foi utilizado um relatório de coleta de tráfego em uma interface de 10Gbps de um *broadband remote access* (BRAS) hoje chamado de *broadband network gateway* (BNG). Para o desenvolvimento do trabalho de pesquisa, foram disponibilizados pela operadora relatórios de 1 dia, 1 semana e 1 mês (todos no espaço de medição de 1 mês) de um relatório de medição real de equipamento ativo na planta.

Este relatório real utilizado possui um total de 719 coletas. Os dados são referentes ao período de 01 de julho de 2020 até 31 de julho de 2020.

Este equipamento (BRAS/BNG) possui diversas portas de dados onde trafegam dados de *download* e *upload* do equipamento ou também chamado tráfego *Incomming* e *Outcoming*. Como são interfaces que possuem alto tráfego e são responsáveis pelo tráfego de VLANs de várias cidades e autenticação, são equipamentos que precisam de monitoramentos constantes e ações de curto prazo.

Os especialistas precisam fazer o monitoramento de curto, médio e longo prazo, para que seja feito o correto planejamento de ações de migração de tráfego (engenharia de tráfego) ou ampliações de capacidade, o que é trabalhado pela projeção de capacidade e através de dados históricos. Caso uma única interface deste equipamento chegue à utilização máxima centenas de clientes terão perdas de pacotes, má percepção do serviço e dificuldade de uso em diversas aplicações, podendo ir de centenas a milhares de clientes. Por este motivo, é de importância que seja monitorado e executado a predição de valores, para que possa ser planejado ações de ampliação e aquisição de equipamentos com antecedência para adequação de tráfego.

Como forma de automação e predição desses dados foi proposto a utilização de RNN, uma vez que são também utilizadas para aplicações com dados temporais. Nesta aplicação será utilizado para predição de comportamento de 1 semana.

O relatório fornecido pelo especialista consiste em um csv e xls, com dados de medição a cada 15 minutos pelo sistema de gerência do equipamento que é coletado por um servidor por protocolo SNMP. Esses dados são então armazenados em um servidor de monitoramento e então apresentados em uma interface de relatório gráfico aos especialistas para análise gráfica.

Nestas estruturas é possível a extração de relatórios contendo então a descrição do equipamento, a interface (ou porta) e a medida em bits trafegados (*download* e *upload*) no momento descrito como data e hora.

A fim de facilitar a visualização dos dados foram adicionadas 2 colunas convertendo os valores de bits para *gigabit*. O formato original do relatório é mostrado na figura 17.

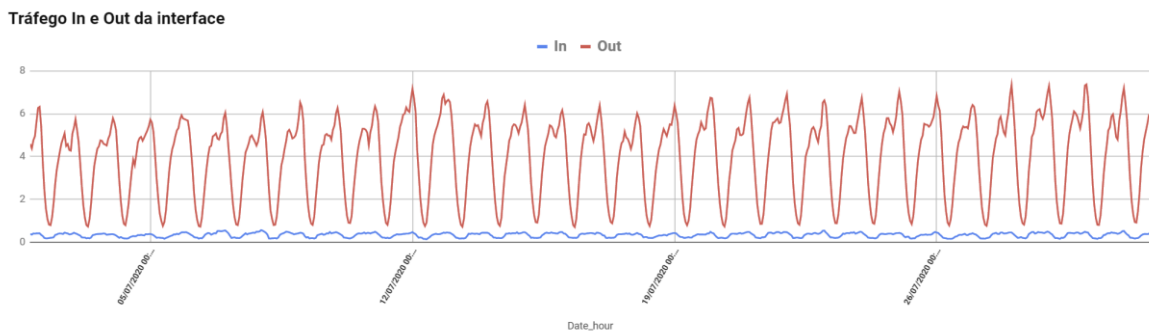
Figura 17- Amostra dos dados utilizados em csv

Date_hour	Incoming traffic HC on interface xe-0/0/0:2	Outbound traffic HC on interface xe-0/0/0:2
01/07/2020 19:00:00	384180808	4584601979
01/07/2020 20:00:00	359939420	4381835856
01/07/2020 21:00:00	412546839	4762524676
01/07/2020 22:00:00	403383216	4953842330
01/07/2020 23:00:00	417092516	5627095395
02/07/2020 00:00:00	416502139	6271301262
02/07/2020 01:00:00	419026885	6325037096
02/07/2020 02:00:00	345988783	5408603937
02/07/2020 03:00:00	293642792	3774824955
02/07/2020 04:00:00	205972255	2478039015
02/07/2020 05:00:00	184872914	1586747544
02/07/2020 06:00:00	186206180	1089928567
02/07/2020 07:00:00	197173736	825607721
02/07/2020 08:00:00	210971556	799866978
02/07/2020 09:00:00	217416578	1180481421
02/07/2020 10:00:00	236302008	1850180202
02/07/2020 11:00:00	343466377	2742522689
02/07/2020 12:00:00	365758490	3388871777

Fonte: O autor (2020).

Do relatório total fornecido com 719 coletas, 605 foram separadas para treinamento o que equivale a 3 semanas de medição. Os demais 114 foram separados para testes, que consistem em comparar o valor real coletado pelo equipamento e a predição de 1 semana de dados. Desta forma haverá condições de comparar graficamente o comportamento aplicado. A visão similar a que os especialistas tem do comportamento é apresentado na figura 18.

Figura 18- Tráfego incoming e outgoing utilizado (todas as amostras de 1 mês)



Fonte: O autor (2020).

O código utilizado para predição foi o LSTM, que possui características e aplicabilidade em situações de dados em linha do tempo e aplicado em predições conforme (EREMENKO; PONTEVES, 2018)

A base de dados após etapa de tratamento contém então 5 colunas ou atributos: *Date_hour*, *In*, *Out*, *Incoming traffic HC on interface*, *Outbound traffic HC on interface*.

As colunas *In* e *Out* são os dados em Gigabit/s e das colunas *Incoming* e *Outbound* em *bit/s*. Estas duas últimas colunas são os dados brutos fornecidos pela gerência proprietária e armazenadas em servidores de monitoramento SNMP, analisadas e extraídas por profissionais de monitoramento de rede. A coluna *date_hour* se refere ao dia e hora da coleta do dado.

3.5 Ambiente de desenvolvimento RNN

Para a aplicação de RNN como preditor de comportamento de tráfego de interface de BRAs, o ambiente de desenvolvimento seguiu as mesmas características de plataforma (*Spyder* e *Google Colab*) e mesmo *hardware* conforme o experimento com SOM.

3.6 Aplicação de LSTM e testes.

Na aplicação do LSTM como preditor de comportamento de tráfego de interface de BRAs, o ambiente de desenvolvimento foi o *Google Colab*.

O código foi inspirado nas referências de (EREMENKO; PONTEVES, 2018). Neste são demonstradas aplicações semelhantes de predições de casos de bolsas de valores se mostram semelhantes as atividades de análise realizadas pelos especialistas que lidam com a necessidade de realizar predições de comportamento de tráfego de uso de rede.

Outros trabalhos são como os de (RUSEK et al., 2020); (HUANG; CHUNG; TANG, 2018); (TELES VIEIRA; LEMOS; LEE, 2003); (TIAN et al., 2020); o uso em predição de tráfego conforme (ZHAO et al., 2018).

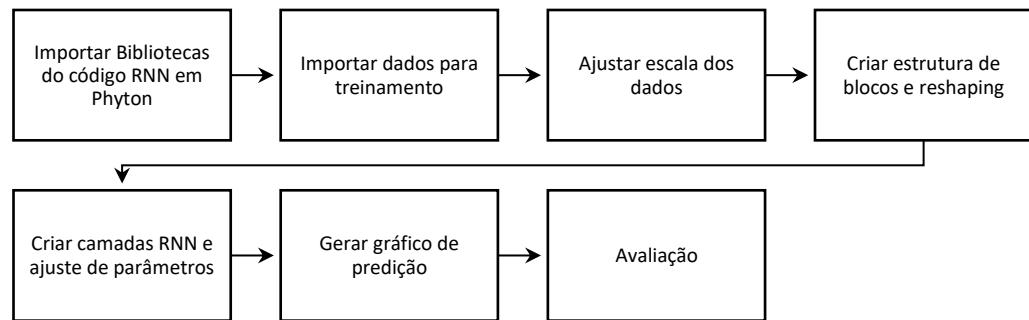
As principais bibliotecas utilizadas para o desenvolvimento foram a biblioteca *numpy* (“NumPy”, 2015) . Biblioteca *pandas* (“*Pandas Library*”, 2020), *scikit-learn* (“*Scikit-learn*”, 2020).

A biblioteca principal utilizada para aplicação do RNN foi a *keras* (“*Keras*”, 2020), especificamente para criação da rede neural a biblioteca *Sequential*, para adicionar a camada de

saída a biblioteca *Dense*, para criar as camadas LSTM a biblioteca de mesmo nome (LSTM) e *Dropout* para função de descarte no gradiente de treinamento (ida e volta) pelas camadas.

O fluxo do planejamento de execução ocorre conforme figura 19:

Figura 19- Etapas realizadas para pelo código usando RNN e em cada execução



Fonte: O autor (2020).

O código desenvolvido (no anexo B) importa as bibliotecas e através do diretório indicado e previamente liberado pela conta *Google* onde está salvo o arquivo com a base de medições no formato *csv* (também é possível trabalhar em outros formatos como *xlsx*).

Além do código, as tabelas de dados utilizados para treinamento e teste estão disponíveis em um drive público também no anexo B. A previsão dos dados de tráfego de BNG será apresentado de forma gráfica, comparando o real e o predito.

Os processos de teste começam tratando a base de dados em formato *csv* (através de planilhas ou planilhas online como o *Google Sheets*) uma vez que é neste formato que os dados foram obtidos.

Os dados fornecidos pelos especialistas (719 amostras) foram então separados em 2 lotes, 1 com 605 medições referentes ao treino e 114 para teste (comparação do valor predito e o valor real).

Uma vez desenvolvido, os dados e códigos foram salvos no mesmo diretório utilizando o *Google drive* e executando o código *Python* pelo *Google Colab* em navegador *Mozilla Firefox*. Neste código é executado a importação das bibliotecas necessárias e o arquivo de base de dados que será então carregado como um *dataset*.

Os dados terão um ajuste de escala e serão executados os parâmetros e algoritmo LSTM com a criação das camadas, neurônios e processos de previsão.

Os dados preditos são então plotados junto aos que foram medidos pelo equipamento (1 semana). Para se encontrar as melhores características e testar a aplicação de ajuste do modelo

foram executados os seguintes testes conforme sugerido em (EREMENKO; PONTEVES, 2018) e pelos testes realizados durante a execução do código (tabela 8):

Tabela 8- Testes com LSTM em predição de interface BRAS

Teste	Item testado	Objetivo	Referência no código	Valores testados
2.1	Time step (passos)	Avaliar influência e melhor valor	Criando Timesteps e 1 saída	5, 20, 30, 50, 60, 80, 100.
2.2	Camadas LSTM	Avaliar influência e melhor valor	Adicionando camadas LSTM e parâmetro Dropout.	4, 8, 12, 16.
2.3	Quantidade de neurônios	Avaliar influência e melhor valor	units	5, 20, 50, 80, 100, 200.
2.4	Épocas	Avaliar influência e melhor valor	epochs	20, 50, 100, 150, 200, 300.
2.5	Predição de tráfego Incoming	Testar a melhor configuração e desempenho da predição	Código Otimizado pasta em anexo.	Melhores valores de 2.1, 2.2, 2.3.
2.6	Predição de tráfego Outbound	Testar a melhor configuração e desempenho da predição	Código Otimizado pasta em anexo.	Melhores valores de 2.1, 2.2, 2.3.

Fonte: O autor (2020).

Assim, conforme tabela 8 foram aplicados 25 testes a fim de se chegar aos melhores parâmetros e ao objetivo de aplicação LSTM. Para cada teste ocorre a coleta do gráfico de predição x tráfego real e curvas de convergência de erro por época. Ou seja, é plotado em gráfico e comparado o resultado predito com o resultado real do comportamento da curva de tráfego de interface do BRAS e calculado os parâmetros estatísticos para avaliar sua correlação e desempenho.

Os dados detalhados de coletas, código e diretório web com mais dados utilizados estão contidos no anexo B.

4 RESULTADOS E DISCUSSÃO

Os testes foram divididos em 2 grupos, um grupo para cada aplicação conforme a tabela 5. A partir destes grupos são detalhados os resultados e observações do experimento conforme os subitens a seguir.

4.1 Testes usando Som

A partir do console do *Spyder* foi executado o código presente no anexo A conforme figura 35 iniciando conforme descrito no método para aplicação usando SOM e para o teste 1.1.

Os dados inseridos se tornam um *dataset*, que pode ser visualizado através do explorador de variáveis do *Spyder*.

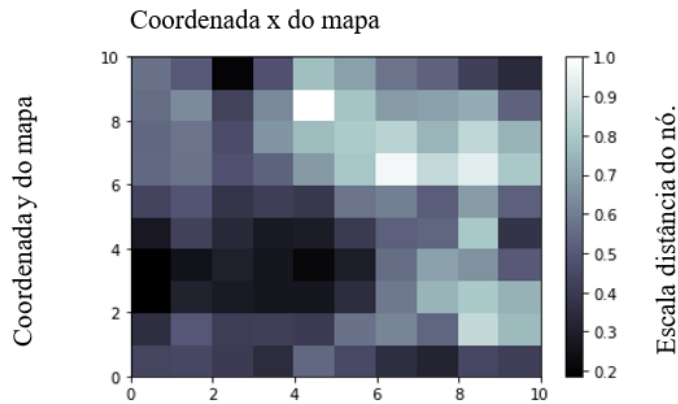
As colunas do *dataset* serão os mesmos dados característicos do relatório original. Cada coluna é um atributo medido ou coletado pela operadora.

No código SOM (anexo A), foram utilizados os seguintes parâmetros: `som = MiniSom (x=10, y=10, input_len = 15, sigma= 1.0, learning_rate = 0.5)`. O `input_len` será 15 devido a quantidade de atributos inserida (de 0 a coluna 14), 1 coluna indexada (referências de cada linha) e mais 14 de atributos. O `x` e `y` são arbitrários, e serão o tamanho da grade do mapa. `Sigma 1` e `learning rate` serão utilizados como 1 e 0,5 por padrão.

Após a normalização dos dados do *dataset*, serão utilizados com um vetor `X`, a mudança em relação a entrada original entre 0 e 1.

Com a execução da etapa do código SOM, onde é calculado as distâncias dos dados através da distância euclidiana, é então gerado (através das etapas e utilizando a o *Spyder*) o mapa SOM com o grid 10x10 como exemplo na figura 20 (escolhido arbitrariamente pelas configurações do *MiniSom* em `x` e `y`).

Figura 20- Mapa SOM com grid 10x10 representando a visão 2D dos dados conforme semelhança.

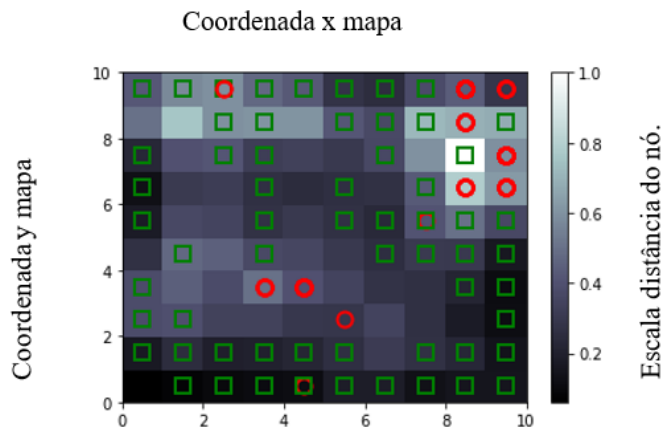


Fonte: O autor (2020)

Aqui quanto mais próximo do valor 1 na escala de branco, mais distante do *cluster* é o valor, e, portanto, será um *outlier* isto indica que o valor analisado pelo agrupamento de suas características divergiu da que era esperada. Quanto mais próximo da escala 0,2 mais um valor central é, e então os valores mais próximos deste valor central serão em tom mais escuro.

É possível observar 5 pontos centrais, sendo que pelo menos 2 possuem grandes quantidades de outros setores. Mas a visualização apenas em escala de cinza pode ser algo difícil de entender, desta forma são adicionados marcadores para direcionar o entendimento do mapa. No caso, são referentes aos casos classificados como sim ou não na coluna de falhas massivas. Que serão quadrados (Massivas) e círculos (Não massivas). Um novo código foi executado e é obtido o resultado (figura 21):

Figura 21- Mapa SOM com auxílio de marcadores conforme as duas saídas do *dataset*.



Fonte: O autor (2020)

Neste caso o mapa dos *outliers* agora está na posição $x=8$ e $y=7$ que corresponde ao quadrado branco (maior escala de classificação).

Para avaliar quais linhas do *dataset* correspondem ao ponto encontrado, através da linha `mappings=som.win_map(X)` é criada uma lista dos valores das notas atribuídos a classificação dos SOM.

Utilizando o *Spyder* e o visualizador de variáveis é possível então saber onde estão os dados mais discrepantes, conforme figura 22. Utilizando uma transformada inversa para reorganizar os dados chega-se aos seguintes equipamentos classificados como *outliers*: 111, 149, 179, 198, 200, 202, 203, 281, 282.

Figura 22- Lista de outliers conforme dataset inserido, visão Spyder

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	111	1	4	1	-29.2	-31.5	160496	5.65696e+06	111	13	100	64	5	4	1
1	149	1	5	1	-26.4	-24.5	1.1767e+06	6.75876e+06	149	9	100	64	1	3	1
2	179	1	6	1	-27.3	-25.6	0.98127	7.53273e+06	179	46	100	64	1	1	1
3	198	1	7	1	-26.5	-26.3	2.34185e+06	6.52776e+06	198	3	100	64	3	1	1
4	200	1	7	1	-32	-26.3	1.99352e+06	5.66038e+06	200	8	40	64	2	2	1
5	202	1	7	1	-27.6	-26.5	27474	6.68173e+06	202	16	100	64	3	2	1
6	203	1	7	1	-31.1	-33.9	3.01597e+06	6.18174e+06	203	8	100	64	3	2	1
7	281	1	9	1	-30	-26.1	2.02238e+06	6.29419e+06	281	3	60	48	2	1	1
8	282	1	9	1	-29	-25.6	1.8191e+06	7.48425e+06	282	10	60	48	2	1	1

Fonte: O autor (2020)

Para a primeira aplicação utilizando o SOM, conseguiu fazer as devidas reduções de dimensionalidade, plotagem do mapa de Kohonen e através do código adaptado a partir de (EREMENKO; PONTEVES, 2018) obter os *outliers*.

Estes *outliers* significam os ONTs que possivelmente foram classificados de forma diferente do que o SOM clusteriza através das características imputadas. Para avaliar esse resultado, através do relatório original foi observado que todos estes valores se encontravam na mesma cidade, todos utilizam o mesmo fabricante de OLT, todos estão em diferentes CTOs, 2 utilizam CTOs de 48 portas e o restante em 64 portas, todos possuem classificação “Normal”

nos relatórios de cadastro de rede, estão distribuídos em 5 OLTs diferentes. Modelo do relatório é apresentado na figura 23.

Figura 23- Visão relatório csv

ONU CLIENT	Region Name	Device Name	Tecnologia	OLT RX Power	ONT RX Power	ONU Distance
111	1	4	1	-29,2	-31,5	160496
149	1	5	1	-26,4	-24,5	1176702
179	1	6	1	-27,3	-25,6	0,98127
198	1	7	1	-26,5	-26,3	2341848
200	1	7	1	-32	-26,3	1993518
202	1	7	1	-27,6	-26,5	27474
203	1	7	1	-31,1	-33,9	3015966
281	1	9	1	-30	-26,1	2022384
282	1	9	1	-29	-25,6	1819098

Fonte: O autor (2020).

Pelo menos 3 destes estão com valores de medição de ONT RX *power* fora do padrão, ao isolar as 5 OLTs utilizadas na OLT com mais dados apontados (4 classificados), todos estavam em ramos de rede com divisão óptica de 64 enquanto os demais estavam classificados com 56. Esse último fato pode indicar sob uma visão mais holística que na classificação geral a maioria das massivas tenha sido dado em redes com maior nível de divisão óptica, o que pode ser correto, pois uma vez que o cabo primário venha a ser rompido ou algum defeito acarreta em maior quantidade de clientes impactados.

Para outro caso, a ONU *distance* por ser muito menor que as demais acabou sendo um ponto de atenção para classificar como outlier. Esse fato poderia ter ocorrido por um erro de medição no relatório ou o fato de por estar mais perto, deveria apresentar um conjunto de valores diferentes do apresentado. Para os demais foram por possuir algum valor muito discrepante em um dos atributos.

Uma vez avaliado o teste 1.1, foi executado a sequência proposta na metodologia, alterando no código e os parâmetros. Para os demais testes foram executados, ajustado no código as coordenadas apontadas no mapa como outliers e executado o laço de execução que lista da base quais itens foram classificados. A lista é composta pelos itens da coluna ONU *CLIENT*. Os dados foram coletados e avaliado se seriam os mesmos nos diversos testes. Como um sistema de avaliação automático espera-se que não ocorressem variações de resultado. Isto pode por não trazer confiabilidade no sistema.

Em cada teste foi também coletado a figura referente ao mapa gerado, no anexo B o *link* do drive contém os códigos, todos os dados coletados e mapas gerados para os testes. Na tabela

9 estão dispostas as quantidades de outliers por teste, como é possível ver os testes apresentaram grande variação entre os testes.

Tabela 9- Quantidade de outliers identificados de acordo com os testes

Teste	Subteste 1 (total de outliers coletados)	Subteste 2 (total de outliers coletados)	Subteste 3 (total de outliers coletados)	Subteste 4 (total de outliers coletados)	Subteste 5 (total de outliers coletados)	Média de outliers por tipo de teste
1.1	9	Não se aplica	Não se aplica	Não se aplica	Não se aplica	Não se aplica
1.2.1	21	93	13	6	8	28,2
1.2.2	228	11	98	1		84,5
1.2.3	7	2	1	5	2	3,4
1.2.4	24	17	14	20	22	19,4
1.3	31	164	13	1	14	44,6
1.4.1	2	13	59	57	21	30,4
1.4.2	2	13	18	3	1	7,4
1.4.3	212	98	3	46	358	143,4
1.5.1	1	19	5	1	24	10
1.5.2	43	2	11	30	9	19

Fonte: O autor (2020).

Quanto as coordenadas nos mapas foram coletadas em cada teste conforme tabela 10:

Tabela 10- Coordenadas observadas nos mapas que indicaram outliers (x e y)

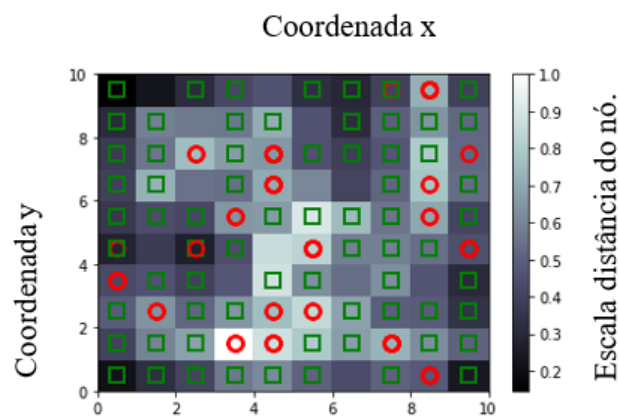
Teste	Subteste 1 (Coordenada no mapa)	Subteste 2 (Coordenada no mapa)	Subteste 3 (Coordenada no mapa)	Subteste 4 (Coordenada no mapa)	Subteste 5 (Coordenada no mapa)
1.1	8,7	-	-	-	-
1.2.1	8,8	2,1	1,8	2,8	8,9
1.2.2	6,1	(5,7)(6,8)	(8,1)(3,1)	2,1	
1.2.3	3,1	6,3	6,5	(5,6)(7,6)	8,6
1.2.4	6,2	3,1	(8,7)(8,8)	5,4	3,1
1.3	8,7	2,1	1,7	1,8	2,1
1.4.1	8,7	2,8	0,8	2,3	1,1
1.4.2	8,8	8,3	(1,7)(1,8)	1,8	1,1
1.4.3	9,5	6,7	3,1	7,7	8,7

1.5.1	8,6	2,2	(7,3)(8,3)	5,1	5,8
1.5.2	(2,1)(6,1)	5,7	3,8	8,8	8,5

Fonte: O autor (2020).

Quanto a visualização nos mapas é possível notar a variação entre os testes conforme as figuras 24 a figura 25 (demais figuras disponíveis no diretório do anexo A):

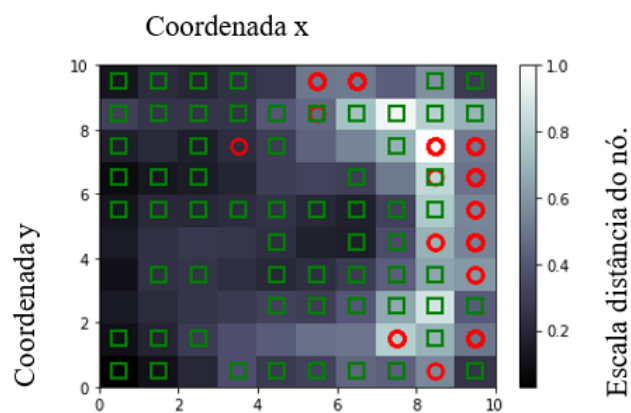
Figura 24- Mapa do teste 1.2.3.1



Fonte: O autor (2020).

Mapa dos testes 1.3.1 indicado na figura 25, nota-se como o ponto dos *outliers* muda à medida que novos mapas são gerados.

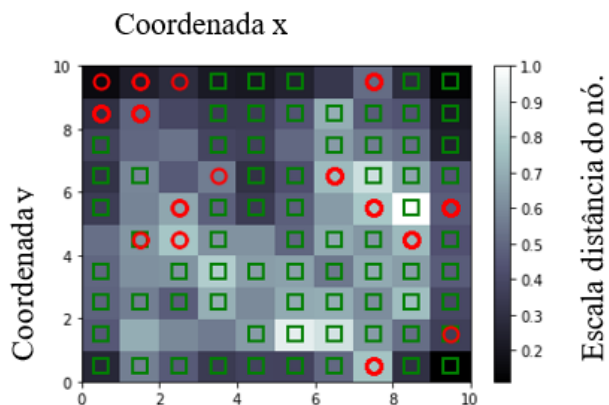
Figura 25- Mapa do teste 1.3.1



Fonte: O autor (2020).

Da mesma forma observa-se na figura 25, novamente a mudança na posição do *grid* do mapa quanto a posição dos outliers, além de que agora a classificação do grupo foi como de não massiva como grupo dominante e de *outliers*.

Figura 26- Mapa teste 1.5.2.5



Fonte: O autor (2020).

Pelos valores coletados nos testes, tem-se a variação dos dados quanto a outliers, que foram distribuídas conforme tabela 11:

Tabela 11- Distribuição estatística de quantidade de outliers por teste

Teste	Média de outliers por tipo de teste	Desvio padrão	Coefficiente de Variação
1.1	Não se aplica	Não se aplica	Não se aplica
1.2.1	28,2	36,684	130,08%
1.2.2	84,5	105,117	124,40%
1.2.3	3,4	2,510	73,82%
1.2.4	19,4	3,975	20,49%
1.3	44,6	67,597	151,56%
1.4.1	30,4	26,092	85,83%
1.4.2	7,4	7,635	103,18%
1.4.3	143,4	143,195	99,86%
1.5.1	10	10,770	107,70%
1.5.2	19	16,956	89,24%

Fonte: O autor (2020).

Observa-se pela tabela 11 que ocorreram grandes variações em geral ao longo dos 5 subtestes, mesmo com os mesmos parâmetros. Os testes 1.2.4 e 1.2.3 apresentaram uma menor variação, o que poderia indicar que um sigma 0,5 seria um bom valor de ajuste.

A mudança dos valores negativos da base não auxiliou na melhoria de estabilidade de resposta como visto no teste 1.3 e em seus 5 subtestes com mesma configuração. A tabela 12 apresenta uma visão aberta de como foi o comportamento da classificação nos grupos.

Tabela 12- Distribuição estatística considerando quantidade de itens classificados por teste e subteste

Teste	Subteste 1		Subteste 2		Subteste 3		Subteste 4		Subteste 5	
	Não massiva	Massiva	Não massiva	Massiva	Não massiva	Massiva	Não massiva	Massiva	Não massiva	Massiva
1.1	0%	100%	-	-	-	-	-	-	-	-
1.2.1	95%	5%	0%	100%	0%	100%	100%	0%	0%	100%
1.2.2	3%	97%	27%	73%	85%	15%	100%	0%	-	-
1.2.3	100%	0%	0%	100%	100%	0%	80%	20%	100%	0%
1.2.4	0%	100%	100%	0%	100%	0%	100%	0%	0%	100%
1.3	100%	0%	100%	0%	100%	0%	0%	100%	100%	0%
1.4.1	50%	50%	23%	77%	20%	80%	25%	75%	0%	100%
1.4.2	0%	100%	0%	100%	0%	100%	0%	100%	0%	100%
1.4.3	39%	61%	31%	69%	67%	33%	37%	63%	35%	65%
1.5.1	0%	100%	95%	5%	20%	80%	0%	100%	100%	0%
1.5.2	100%	0%	0%	100%	91%	9%	100%	0%	0%	100%

Fonte: O autor (2020).

Conforme tabela 12, mesmo dentro do mesmo teste, com mesmos parâmetros ocorreram grandes variações, seja da classificação do grupo delimitado como *outlier* e também pela quantidade. Os testes de variação de atributos esperavam-se que auxiliassem a melhorar a resposta uma vez que possíveis atributos não essenciais poderiam estar atrapalhando nos agrupamentos, porém também não surtiram em resultados de melhora da resposta. Entre os testes do 1.4.1, 1.4.2 e 1.4.3 mesmo diminuindo os atributos ainda foi coletado grande variação.

Os testes em 1.5.1 e 1.5.2 seguiram da mesma forma que os resultados anteriores com grandes variações.

Esses dados podem indicar que o modelo não é adequado (não acurado) para a situação real proposta, tanto como os dados podem não estar adequados para uma análise em cluster. Seria necessário novos dados, com menos atributos ou buscar melhorias no código aplicado.

Outra forma seria avaliar as causas de falha massiva na rede GPON considerando informações de gerência da OLT. Estas informações trariam dados adicionais não considerados nestes testes onde podem indicar que a falha não ocorreu devido a problemas na rede óptica em campo, mas por problemas de software nas ONUs, alguma falha de processamento da OLT ou até mesmo algum problema na rede de *backhaul*.

4.2 Testes usando RNN

Seguindo a metodologia, o primeiro teste executado no *Spyder* foi feito com 50 neurônios e 100 épocas. Já neste teste ocorreram falhas com alto tempo de execução do código e simultaneamente os 4 núcleos de processamento do laptop utilizado chegaram a 98% de utilização, com aumento significativo de temperatura, utilização de memória RAM.

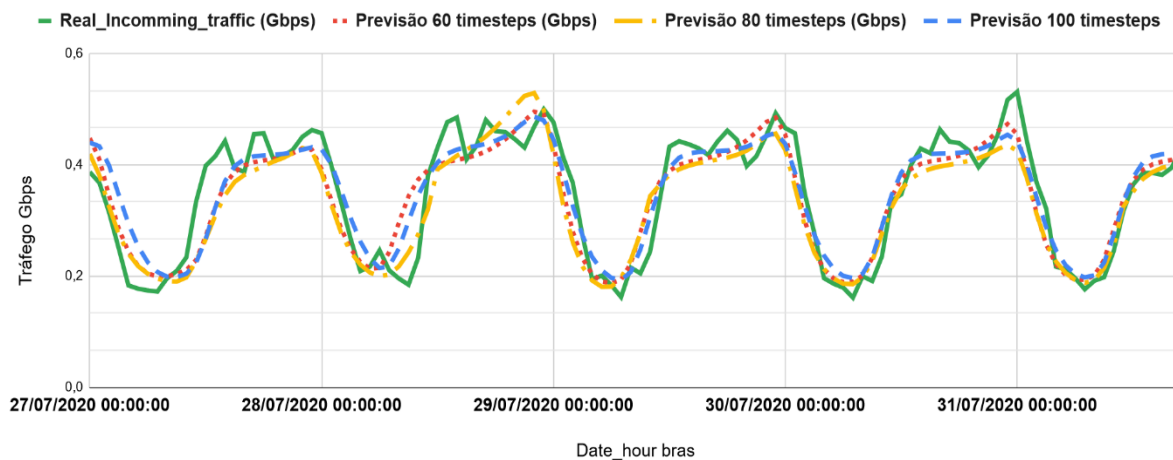
Após seguidos testes com os mesmos parâmetros e falhas, foi reduzindo as épocas a metade foi possível a execução no ambiente, porém demandando muito tempo de execução. Os sucessivos testes, apresentaram as mesmas falhas.

Para que fosse possível a realização dos experimentos, foi adotado como parâmetro a execução no ambiente do *Google Colab*, aplicando os testes de acordo com o proposto na metodologia (3.6).

4.2.1 Teste RNN 2.1 variando timesteps

Para os testes executados na proposta 2.1, com valores abaixo de 60 *timesteps* não foram obtidas curvas de predição. Para os valores de 60, 80 e 100 foram obtidas as predições no valor de 114 amostras, que representam 114 horas futuras em relação aos dados de treinamento gerando as curvas da figura 27. A curva de traçado contínuo em verde representa os valores reais medidos pelo equipamento entre o dia 27 de julho de 2020 e 01 de agosto de 2020. Plotado em vermelho a curva predita para 60 *timesteps*, em amarelo 80 *timesteps* e em azul 100 *timesteps*.

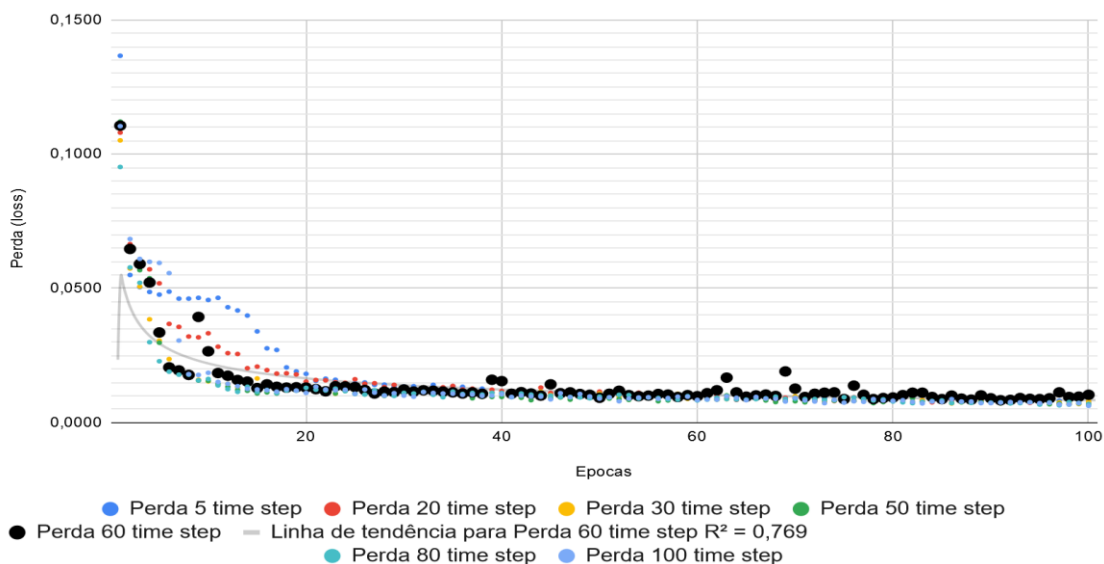
Figura 27 - Gráfico Real x previsões de tráfego *incomming* teste 2.1



Fonte: O autor (2020).

Para cada época também foi coletado a perda e para cada teste executado sendo gerado as curvas da figura 28. É possível perceber que após 20 épocas existe um padrão de convergência do sistema e antes variando de acordo com o valor utilizado no teste.

Figura 28- Curvas perda por épocas para testes de variação de *timestep*



Fonte: O autor (2020).

Em relação ao comportamento analisado e métricas de aferição a curva de predição com utilização de 60 *timesteps* apresenta boa convergência. Estatisticamente foram levantados por

cálculo os seguintes dados através dos valores preditos em cada teste comparado a curva de medição real (sendo y_i valor real e \hat{y}_i valor predito) erro médio (15), erro médio quadrático (16), Raiz do erro médio quadrático (RMSE) (17), coeficiente de determinação ou R^2 (indica quanto o modelo foi capaz de explicar os dados coletados) (18), coeficiente de Pearson ou R (mede o grau da correlação entre duas variáveis) (19).

$$ME = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i) \quad (15)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (16)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (17)$$

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (18)$$

$$R = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \cdot \sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}} \quad (19)$$

A tabela 13 apresenta os indicadores das equações 15 a 19.

Tabela 13 - Indicadores para testes variando timesteps

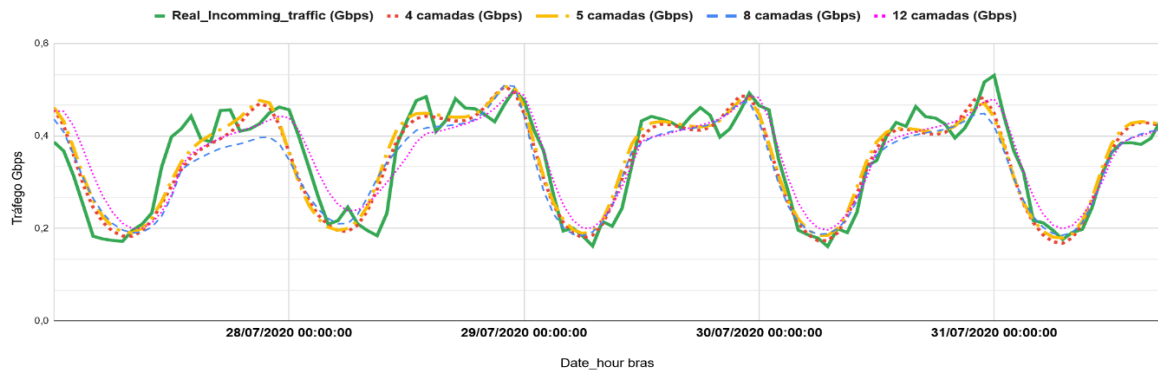
Indicador	Previsão 60 <i>timesteps</i>	Previsão 80 <i>timesteps</i>	Previsão 100 <i>timesteps</i>
Erro médio (ME)	0,48%	1,52%	-0,31%
Erro médio quadrático (MSE)	0,21%	0,22%	0,18%
RMSE_RELATIVO (RMSE_R)	0,04%	0,04%	0,04%
Raiz do erro médio quadrático (RMSE)	4,60%	4,64%	4,26%
R^2	82,60%	84,01%	85,63%
R	90,88%	91,66%	92,53%

Fonte: O autor (2020).

4.2.2 Teste RNN 2.2 variando camadas

No teste 2.2 foram variadas as quantidades de camadas na rede LSTM. Aqui foi mantido a quantidade de 60 *timesteps*, 50 neurônios, *dropout* em 0,2 e 100 épocas. Na figura 29, é possível observar a variação de comportamento de acordo com o ajuste de parâmetro.

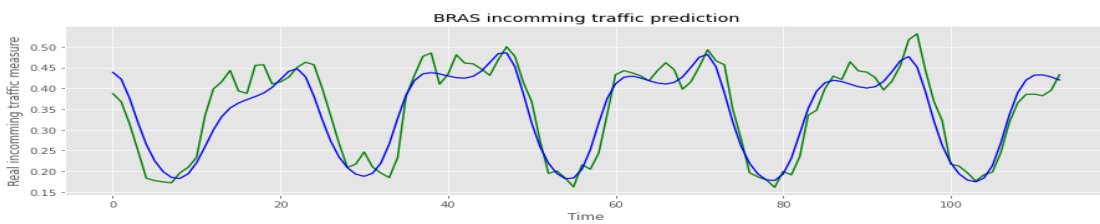
Figura 29- Curvas do teste 2.2 concatenadas



Fonte: O autor (2020).

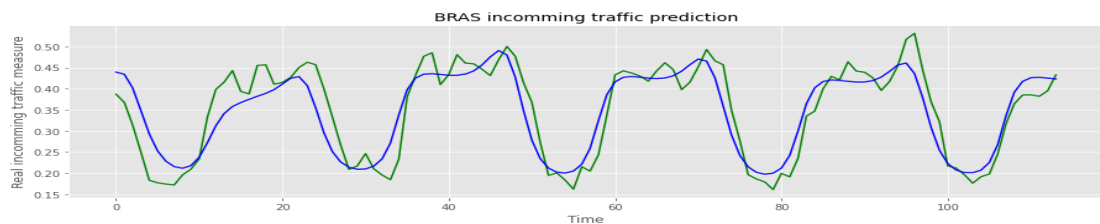
Na figura 46 a 49 é mostrado a visão da plotagem no Python junto com a curva real.

Figura 30- Teste LSTM 2.2 com 4 camadas



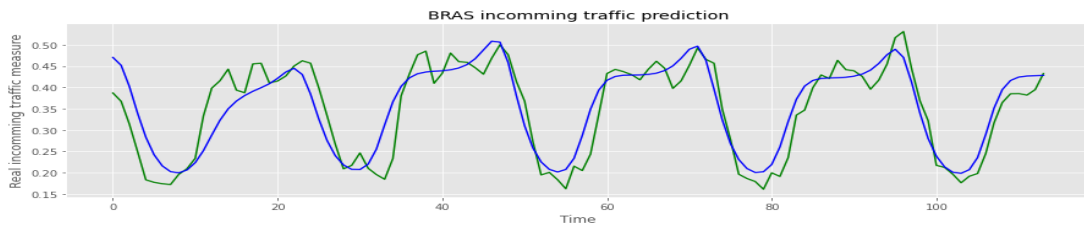
Fonte: O autor (2020).

Figura 31- Teste LSTM com 5 camadas



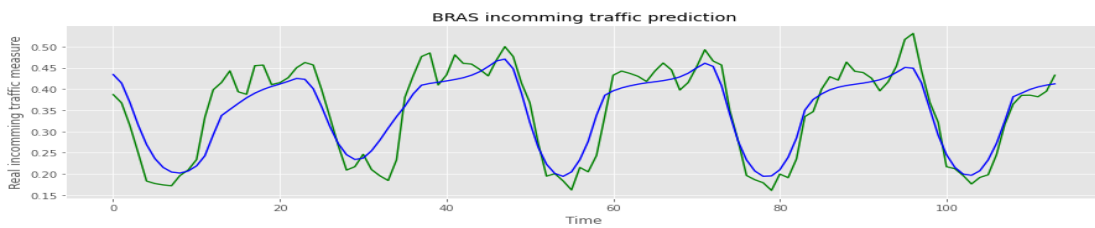
Fonte: O autor (2020).

Figura 32- Teste LSTM com 8 camadas



Fonte: O autor (2020).

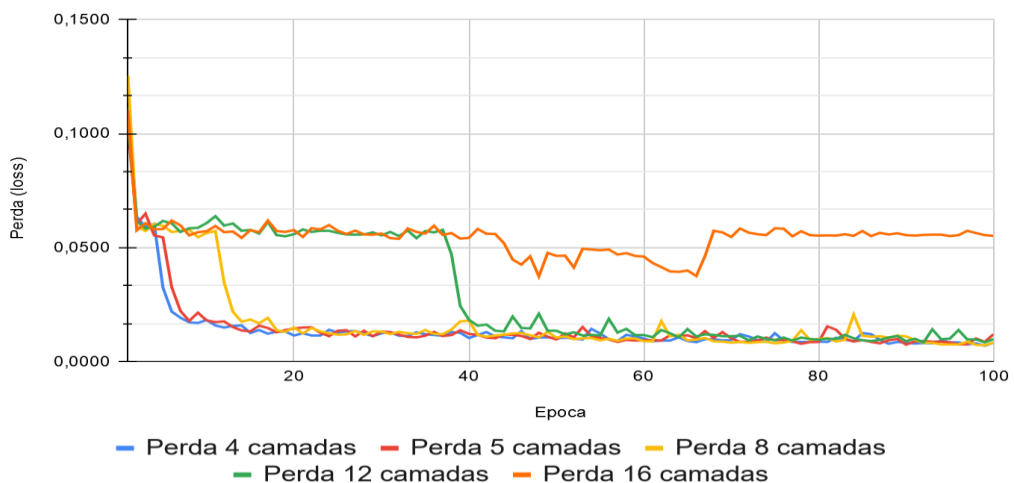
Figura 33- Teste LSTM com 12 camadas.



Fonte: O autor (2020).

As curvas de perdas (ou loss) por época variando quantidade de camadas, são apresentadas na figura 54:

Figura 34- Curvas perdas por época variando quantidade de camadas.



Fonte: O autor (2020).

Tabela 14 - Indicadores testes variando camadas

Indicador	4 camadas	5 camadas	8 camadas	12 camadas
Erro médio (ME)	0,59%	-0,08%	1,35%	-0,49%
Erro médio quadrático (MSE)	0,17%	0,19%	0,20%	0,20%
RMSE_RELATIVO (RMSE_R)	0,04%	0,04%	0,04%	0,04%
Raiz do erro médio quadrático (RMSE)	4,18%	4,33%	4,46%	4,46%
R ²	85,70%	84,29%	84,48%	84,48%
R	92,58%	91,81%	91,91%	91,91%

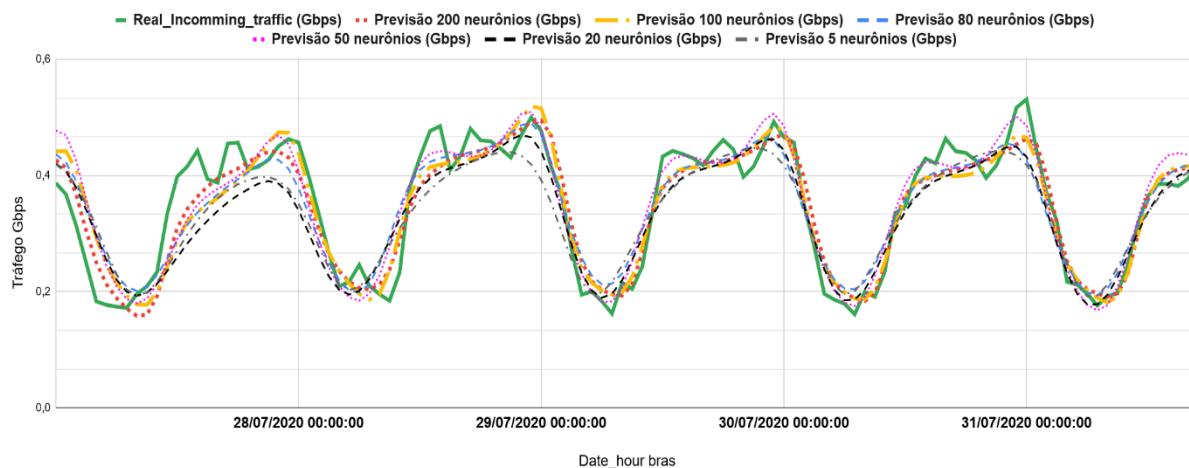
Fonte: O autor (2020).

Pelo comportamento apresentado nos testes 4 camadas apresenta um bom desempenho quanto a predição dos dados e uma melhor convergência quanto a perda ao passar das épocas.

4.2.3 Teste RNN 2.3 variando quantidade de neurônios

O teste 2.3 consiste em variar a quantidade de neurônios, mantendo-se os parâmetros anteriores e obtendo-se as seguintes curvas:

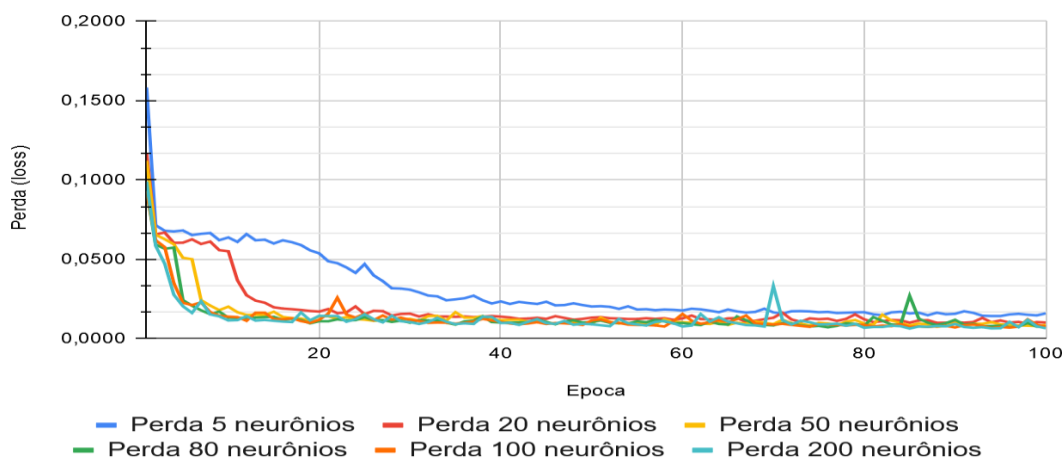
Figura 35- Curvas de predição variando quantidade de neurônios



Fonte: O autor (2020).

Os testes mostram que com 50 neurônios já existe uma boa relação de predição (figura 35), uma grande quantidade de neurônios e camadas exigem mais capacidade computacional para execução. Na figura 36 com poucos neurônios o erro demora mais épocas para estabilizar.

Figura 36- Visão de perda por épocas com variação de neurônios



Fonte: O autor (2020).

Os indicadores dos testes obtidos são apresentados na tabela 15.

Tabela 15- Indicadores dos testes variando quantidade de neurônios

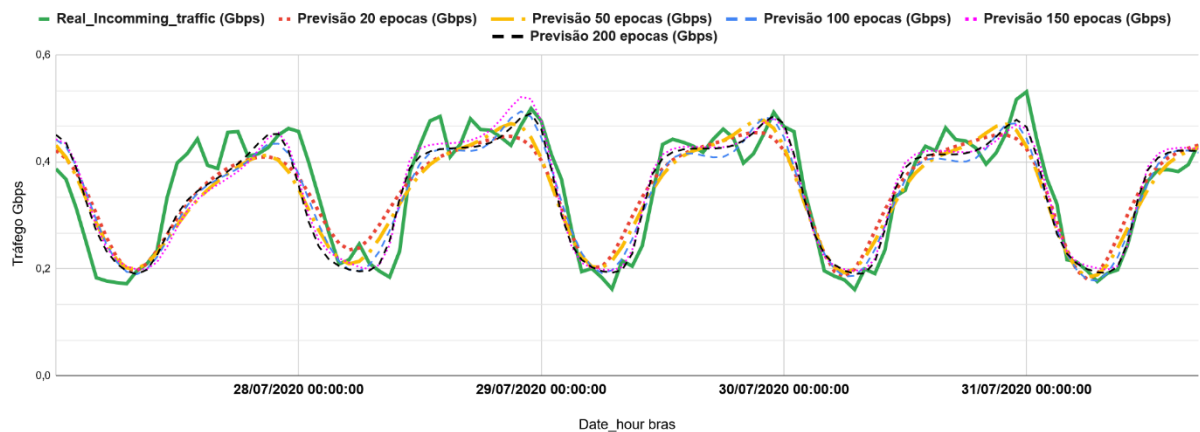
Indicador	Previsão 200 neurônios	Previsão 100 neurônios	Previsão 80 neurônios	Previsão 50 neurônios	Previsão 20 neurônios	Previsão 5 neurônios
Erro médio (ME)	0,27%	0,22%	0,00%	-0,49%	1,47%	1,26%
Erro médio quadrático (MSE)	0,12%	0,16%	0,18%	0,18%	0,26%	0,28%
RMSE_RELATIVO (RMSE_R)	0,03%	0,04%	0,04%	0,04%	0,04%	0,05%
Raiz do erro médio quadrático (RMSE)	3,51%	4,01%	4,24%	4,24%	5,11%	5,27%
R ²	89,90%	86,61%	85,31%	85,31%	80,66%	79,91%
R	94,82%	93,07%	92,36%	92,36%	89,81%	89,39%

Fonte: O autor (2020).

4.2.4 Teste RNN 2.4 variando quantidade de épocas

No teste 2.4 foi proposto a variação das épocas e a observação de sua influência. Assim foram mantidos os demais parâmetros (4 camadas, 50 neurônios, 60 *timesteps*) obtendo-se as seguintes curvas:

Figura 37- Curvas de previsão variando quantidade de épocas

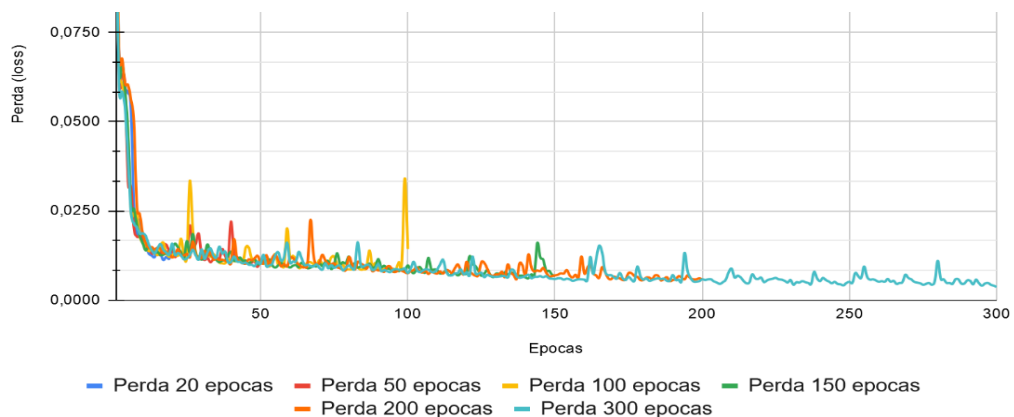


Fonte: O autor (2020).

No teste 2.4 com 150 épocas é obtido um bom parâmetro em que a curva apresenta boa similaridade ao que seria o tráfego real. Porém quanto mais épocas mais tempo o para finalizar a previsão, é preciso ponderar o custo benefício.

Quanto ao *loss* foram obtidos os seguintes valores:

Figura 38- Perda por época no teste 2.4



Fonte: O autor (2020).

Os valores dos indicadores obtidos são apresentados na tabela 16. Apesar de com 200 épocas ser obtido bons indicadores o tempo de execução do código é maior em relação aos demais.

Tabela 16- Indicadores dos testes variando quantidade de épocas

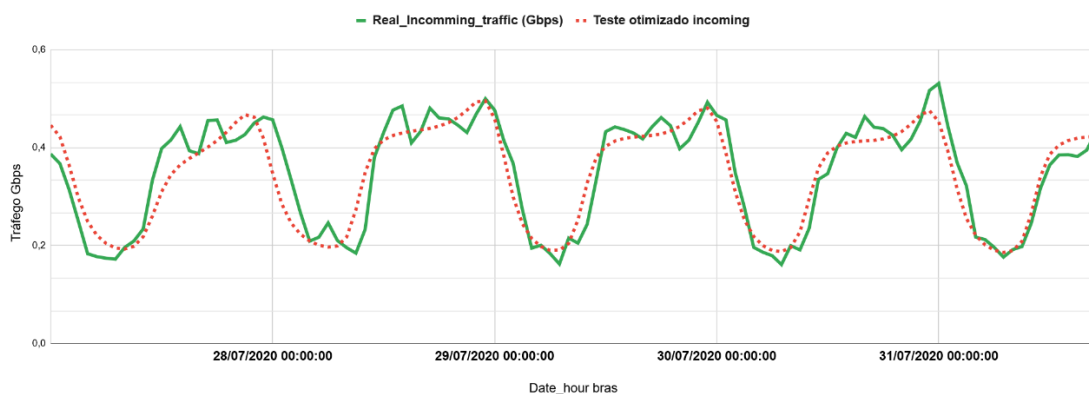
Indicador	Previsão 20 épocas	Previsão 50 épocas	Previsão 100 épocas	Previsão 150 épocas	Previsão 200 épocas
Erro médio (ME)	0,04%	0,64%	0,60%	0,03%	0,40%
Erro médio quadrático (MSE)	0,24%	0,24%	0,19%	0,19%	0,17%
RMSE_RELATIVO (RMSE_R)	0,04%	0,04%	0,04%	0,04%	0,04%
Raiz do erro médio quadrático (RMSE)	4,92%	4,87%	4,34%	4,34%	4,13%
R ²	81,37%	81,11%	84,22%	84,22%	85,87%
R	90,21%	90,06%	91,77%	91,77%	92,66%

Fonte: O autor (2020).

4.2.5 Teste 2.5 predição de tráfego *incomming*

Uma vez aplicado individualmente os testes de 2.1 a 2.4 e avaliado os melhores parâmetros em cada variável, no teste 2.5 foi configurado 60 *timesteps*, 4 camadas, 50 neurônios, 150 épocas e obtido a seguinte curva:

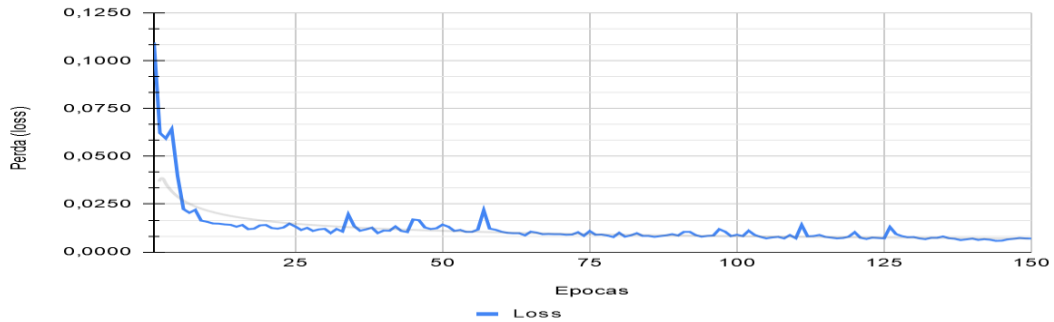
Figura 39 - Curvas para tese otimizado de tráfego *incomming*



Fonte: O autor (2020).

A curva de convergência da perda por épocas foi conforme a figura 40:

Figura 40- Perda por épocas teste 2.5



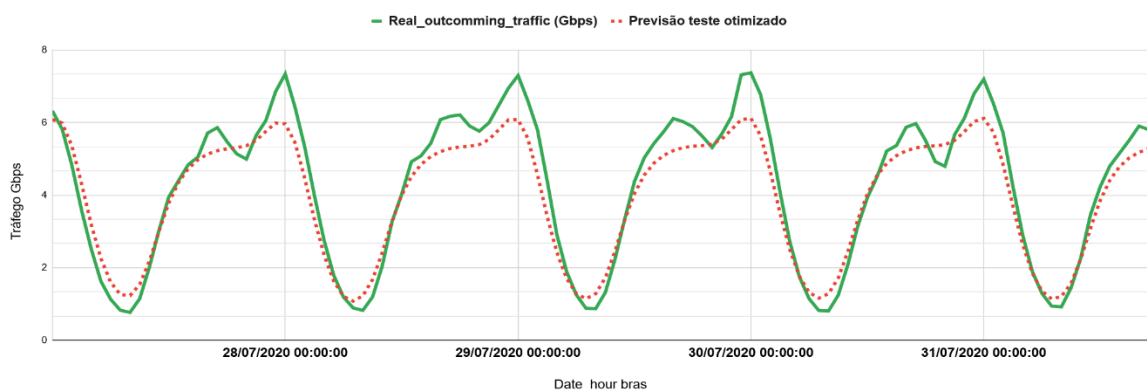
Fonte: O autor (2020).

Este modelo apresentou boa convergência do erro em relação as épocas e o padrão predito apresentam similaridade ao dado real coletado pela gerência do equipamento. O erro médio foi de 0,28%, MSE 0,17%, RMSE relativo foi de 0,04%, RMSE 4,18%, R² ficou em 85,46% e R no valor de 92,44%.

4.2.6 Teste 2.6 predição de trafego outcomming

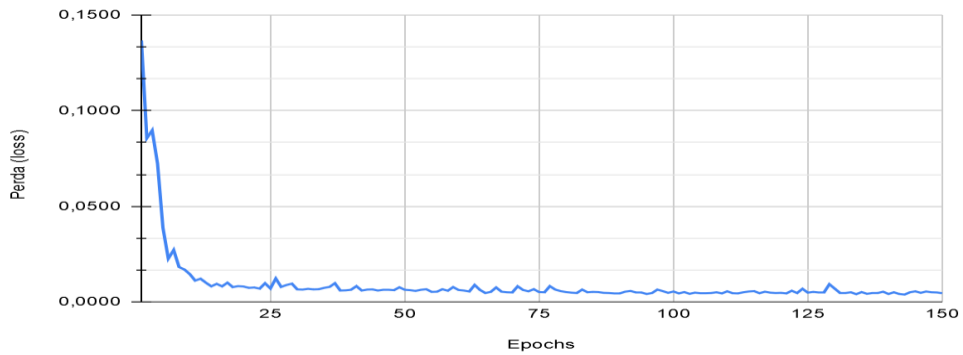
Para se validar a aplicação com outros dados de tráfego foi utilizado o tráfego *out* da mesma interface, mas que possui valores maiores. Da mesma forma e parâmetros do teste 2.5 foram obtidas as seguintes curvas (figura 41):

Figura 41- Predição de trafego teste 2.6 outcomming tráfico otimizado



Fonte: O autor (2020).

Figura 42- Perda por época teste 2.6



Fonte: O autor (2020).

Como nos demais testes a figura 42 apresenta a perda por época. O erro médio encontrado foi de 24,6%, MSE 31,41%, RMSE relativo foi de 0,49%, RMSE 56,05%, R^2 ficou em 96,17% e R no valor de 98,07%.

Assim como nos demais casos para o tráfego *outcomming* indica-se que pode ser feito ajustes específicos para este caso.

5 CONCLUSÃO

Foi possível com este trabalho aplicar dois conceitos propostos de IA para auxiliar a automatização de parte de atividades manuais do dia a dia de uma operadora de telecomunicações (utilizando dados reais).

A primeira situação proposta, foi criar grupos (clusters) em um mapa de Kohonen utilizando a base real de um relatório de análise de falha massiva GPON. Neste caso foi encontrado dados discrepantes pelo teste 1.1. Os demais resultados apontam que utilizando o SOM, parte dos clientes estão com valores de potência óptica fora do padrão (possível defeito de campo).

Sob uma visão mais ampla, é possível que a maioria das massivas tenham ocorrido em redes com maior nível de divisão óptica (128 portas de CTO por porta PON), o que indica acerto, uma vez que caso o cabo primário tenha rompido haveria mais clientes impactados.

Os *outliers* indicam que os dados coletados podem ter sofrido erro humano no levantamento. Outra hipótese é que a falha não foi devida a uma causa exclusiva, mas algo pontual. Outros problemas que poderiam já ter ocorrido, além dos dados utilizados, seria defeito do ONT e de conectores. Não necessariamente estes *outliers* estariam sob uma falha massiva, mas já estavam com outro problema anteriormente.

Nos demais testes foram encontradas variações da quantidade de *outliers* coletados, o que gera uma visão inconclusiva sobre sua utilização geral. Apesar deste fato foi possível a obtenção de um mapa de duas dimensões conforme proposto nos testes e que observando os outliers dividiu os grupos em massivas e não massivas. Porém não manteve uma boa acurácia nas repetições de testes com mesmos parâmetros.

Como trabalhos futuros, pode ser utilizado nova base de dados ou a redução de atributos utilizando a correlação de atributos. O uso do algoritmo SOM possui potencial em outras situações de análise de malhas.

Como trabalhos futuros, com redes SOM e considerando o mesmo cenário de avaliação de um relatório de falhas de massivas GPON em campo, poderia ser aplicado a outras bases de dados com melhor consistência. Neste caso seria feito pelo algoritmo a clusterização dos dados pelas características próximas e assim o operador possivelmente encontraria o padrão de operação de sua planta ou aplicação além de conseguir listar os outlier, ou dados distantes do padrão. Esses outliers poderiam indicar anomalias em configurações de plantas industriais, ponto de extração dos melhores valores de calibração de processos ou controle aplicado.

Na robótica, a aplicação do SOM pode ser feita para cálculo de trajetória de deslocamento de robôs. Outra aplicação seria para tipologia de curvas de carga de transformadores. Também a classificação de motores pela mineração de dados de curva de desempenho.

Em de mineração de dados, o SOM pode ser aplicado para classificação de bases de dados de desempenho de alunos, auxiliando a encontrar aqueles que tem dificuldades de aprendizado e suas possíveis causas. Também pode auxiliar na seleção de candidatos a vagas de trabalho devido a características similares a especificação de uma vaga. Pode auxiliar na mineração de melhores ações de vendas, marketing e direcionar produtos adequados aos melhores públicos. Outra aplicação para melhorar o relacionamento entre clientes e empresas pelo melhor conhecimento e extração de características pelo perfil ou pelo tipo de produto.

Na segunda situação proposta, automatizar a predição de comportamento do tráfego de dados em interfaces reais de broadband network access utilizando o conceito de rede neural recorrente LSTM, foram obtidas as curvas de predição e comparadas aos dados reais conforme proposto no método.

A aplicação de LSTM utilizando um notebook apresentou falhas críticas, sendo posteriormente aplicável ao utilizar ferramentas de processamento em nuvem. Os resultados foram satisfatórios de acordo com a proposta apresentada, demonstrando um potencial para trabalhos futuros.

Pelos gráficos foram observadas boas respostas do modelo predito comparado ao real. Um ponto de grande atenção para trabalhos futuros é o dimensionamento das características de quantidade de neurônios, camadas sobrepostas e épocas. Uma baixa quantidade destes elementos pode levar a baixa performance do modelo enquanto um excesso pode levar a utilização elevada de recursos de hardware e tempo alto de processamento e resposta.

O uso adequado de recursos por empresas de telecomunicações deve ser avaliado criteriosamente. Um custo operacional incorreto ou muito alto pode inviabilizar um projeto. Projetos que não são bem estruturados e planejados utilizando ferramentas de antecipação de problemas demandam maior custo. Assim, o trabalho pode contribuir com uma avaliação mais assertiva, ou como base para trabalhos na área.

Como trabalhos futuros, o LSTM pode ser aplicado em situações que possuam dados históricos ou sequenciais. Podem ser citadas análise e predição tráfego, projeção de comportamento de equipamentos (processamento, uso de memória, aquecimento), previsão de crescimento de consumo de energia. Além da área de processamento de dados é possível a

utilização em projeção de crescimento de carga em linhas de transmissão usando histórico de monitoramento, projeção e previsão de sobrecarga em transformadores.

Ainda como trabalhos futuros, o LSTM no campo dos negócios poderia ser utilizado para *chatbots* de atendimento, prevendo a melhor resposta para um atendimento automatizado considerando o histórico de palavras ou experiências anteriores dos clientes de operadoras. Podem ser utilizadas para previsão de aumento ou queda de receita, vendas, provisão de estoques.

Esta pesquisa durante seu desenvolvimento resultou além desta tese, até o presente momento, o artigo publicado em conferência internacional:

W. S., Silva, A. S. Morais, W. O. Silva, “Proposta de uso de redes neurais para clusterização de dados no contexto de análise qualitativa de informações de reclamações em serviços de telecomunicações”. 9º Congresso Ibero-americano de Investigación Cualitativa (CIAIQ2020).

REFERÊNCIAS

AFFONSO, G. S. **Mapas Auto-organizáveis de Kohonen (SOM) aplicados na avaliação dos parâmetros da qualidade da água**. 2011. 89 f. Universidade de São Paulo, 2011.

ANACONDA. **The World's Most Popular Data Science Platform**. Disponível em: <<https://www.anaconda.com/>>. Acesso em: 13 ago. 2020.

ANATEL. **Panorama - Reclamações 2019**. Disponível em: <https://sei.anatel.gov.br/sei/modulos/pesquisa/md_pesq_documento_consulta_externa.php?eEP-wqk1skrd8hSlk5Z3rN4EVg9uLJqrLYJw_9INcO60_nY9GnnWvxl_B97punLsqsSpwlLKycbBJttIA0I-QzSN0aivoJFXfv1hghzUup_fZlw86VfcrGmgZNCLm49A>. Acesso em: 13 ago. 2020.

ANATEL. **Banda Larga Fixa**. Disponível em: <<https://www.anatel.gov.br/paineis/ acessos/banda-larga-fixa>>. Acesso em: 11 ago. 2020.

ARAGON-ZAVALA, A. Capacity Planning and Dimensioning. **Indoor Wireless Communications: From Theory to Implementation**. 1º ed. [S.l.]: Wiley Telecom, 2017. p. 257–292. <https://doi.org/10.1002/9781119004547>

BACKBONE RCT-SC passa a ser gerenciado pelo PoP-SC. **Memória da Internet Acadêmica de Santa Catarina**. Disponível em: <<https://memoria.pop-sc.rnp.br/2015/01/backbone-rct-sc-passa-a-ser-gerenciado-pelo-pop-sc/>>. Acesso em: 11 ago. 2020.

BRAGA, A. DE P.; LUDEMIR, T. B.; CARVALHO, A. C. P. L. F. **Redes Neurais Artificiais Teoria e Aplicações**. Rio de Janeiro: LTC, 2014.

CAMBUIM, L. **Mapas Auto-Organizáveis de Kohonen**. Disponível em: <<https://www.cin.ufpe.br/~lfsc/cursos/introducaoainteligenciaartificial/IA-Aula11-MapasAutoOrganizaveis.pdf>>. Acesso em: 30 maio 2020.

CIABURRO, G.; VENKATE, B. **Kohonen SOM**. Disponível em:

<<https://www.oreilly.com/library/view/neural-networks-with/9781788397872/24ba0001-ddea-493f-96b8-aa9be7c921d3.xhtml>>. Acesso em: 13 ago. 2020.

COPPIN, B. **Inteligência Artificial**. 1. ed. Rio de Janeiro: LTC, 2013.

EREMENKO, K.; PONTEVES, H. **Machine Learning A-Z**. Disponível em:

<<https://www.superdatascience.com/pages/machine-learning>>. Acesso em: 04 abril. 2020.

F.JACINTO, J. N. **Aspectos gerais sobre IPTV**. Disponível em:

<http://www.img.lx.it.pt/~fp/cav/ano2012_2013/Trabalhos_MEEC_2012_2013/Artigo14/55098_55146_Site_IPTV/Site_55098_55146/arquitectura.html>. Acesso em: 11 ago. 2020.

FARIAS, G. **Introdução à Computação**. [S.l.]: UFPB, [S.d.]. Disponível em:

<<http://producao.virtual.ufpb.br/books/camyle/introducao-a-computacao-livro/livro/livro.chunked/index.html>>. Acesso em: 5 ago. 2020.

FOROUZAN, B. A. **Comunicação de Dados e Redes de Computadores**. 4. ed. São Paulo: McGrall Hill, 2010.

FREITAS, M. A. DE S.; FREITAS, G. B. **Inteligência Artificial e Machine Learning: Teoria e Aplicações**. 1. ed. [S.l: s.n.], 2020.

HARDTEC. ARPANET: **As Origens da Internet**. Disponível em:

<<https://hardtec.com.br/arpamet-as-origens-da-internet/>>. Acesso em: 11 ago. 2020.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. São Paulo: Bookman, 2007.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997. <https://doi.org/10.1162/neco.1997.9.8.1735>

HUANG, Y.; CHUNG, W.; TANG, X. A temporal recurrent neural network approach to detecting market anomaly attacks submission type: Short paper. **2018 IEEE International Conference on Intelligence and Security Informatics**, ISI 2018, p. 160–162, 2018.

<https://doi.org/10.1109/ISI.2018.8587397>

IEEE. **IEEE Xplore**. Disponível em: <<https://ieeexplore.ieee.org/Xplore/home.jsp>>. Acesso em: 11 ago. 2020.

ZHAO, J; et. al. Towards traffic matrix prediction with LSTM recurrent neural networks, *Electronics Letters*, vol. 54, pp. 566-568, 2018. <https://doi.org/10.1049/el.2018.0336>

ITU.T. G.983.1. **International Telecommunication Union**. 2003. Disponível em: <<https://www.itu.int/rec/T-REC-G.983.1-200303-S!Amd2/en>>. Acesso em: 11 ago. 2020.

ITU.T. G.984.1. **International Telecommunication Union**, 2012. Disponível em: <<https://www.itu.int/rec/T-REC-G.984.1-201204-I!Amd2/en>>.

KEISER, G. **Comunicações por Fibras Ópticas**. 4. ed. Sao Paulo: McGrall Hill, 2013.

KOHONEN, T. The Self-Organizing Map. **Proceedings of the IEEE**, 1990. <https://doi.org/10.1109/5.58325>

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet**. 6. ed. São Paulo: Pearson, 2013.

LINS, A. Como ir do IPv4 para o IPv6, passando pelo CGNAT e NAT64. **Forum IXbr**. Disponível em: <<https://regional.forum.ix.br/files/apresentacao/arquivo/596/Transicao IPv6 - IXbr-SaoPaulo.pdf>>. Acesso em: 8 ago. 2020.

LIU, T. et al. Application of neural network in fault location of optical transport network. **China Communications**, v. 16, n. 10, p. 214–225, 2019. <https://doi.org/10.23919/JCC.2019.10.014>

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, dez. 1943. Disponível em: <<https://link.springer.com/article/10.1007/BF02478259>>. Acesso em: 11 ago. 2020. <https://doi.org/10.1007/BF02478259>

MEDEIROS, L. F. DE. **Inteligência artificial aplicada: uma abordagem introdutória**. Curitiba: Editora Intersaberes, 2018.

MINSKY, M.; PAPERT, S. **Perceptrons: An Introduction to Computational Geometry**. **M.I.T. Press.**, 1969. Disponível em: <<https://psycnet.apa.org/record/1969-35017-000>>. Acesso em: 12 out. 2020.

MIRANDA JÚNIOR, G. et al. **Estimativa de Qualidade de Serviço em Estações Wi-Fi Utilizando Redes Neurais Recorrentes**. v. 11, p. 57–70, 2019.
<https://doi.org/10.5753/sbrc.2019.7350>

MONTE sua Rede com tecnologia GPON ou EPON. **eFurukawa**. Disponível em: <<https://www.efurukawa.com/br/monte-sua-rede/resultado-preconf?solutionId=400002#modal-subscribe>>. Acesso em: 11 ago. 2020.

MONTOURO, F. **Infraestrutura de Rede Interna: Fibra Óptica ou Cabo de Cobre?** Disponível em: <<http://fabiomontoro.blogspot.com/2013/03/fibra-optica-ou-cabo-de-cobre.html>>. Acesso em: 11 ago. 2020.

MORAL, E. DEL. **Mapas Auto Organizáveis SOMs Mapas de Kohonen**. **LSI USP**. Disponível em: <www.lsi.usp.br/ICONE>. Acesso em: 6 jul. 2020.

MX960 Images - Juniper Networks. **Juniper**. Disponível em: <<https://www.juniper.net/us/en/company/press-center/images/image-library/mx960/>>. Acesso em: 11 ago. 2020.

NEURÔNIO. **Anatomia do Sistema Nervoso Central**. Disponível em: <<https://sites.google.com/site/anatomiasistemamanervosocentral/home/neuronio>>. Acesso em: 12 ago. 2020.

NUMPY. **Numpy**. Disponível em: <<https://numpy.org/>>. Acesso em: 13 ago. 2020.

O NEURÔNIO Artificial. **GSigma**. Disponível em:
<https://www.gsigma.ufsc.br/~popov/aulas/rna/neuronio_artificial/index.html>. Acesso em:
12 ago. 2020.

O QUE muda com a LGPD — LGPD - Lei Geral de Proteção de Dados Pessoais. **Serpro**.
Disponível em: <<https://www.serpro.gov.br/lgpd/menu/a-lgpd/o-que-muda-com-a-lgpd>>.
Acesso em: 17 out. 2020.

O MODELO TCP/IP. **Pagina.Fe**. Disponível em:
<https://paginas.fe.up.pt/~mrs01003/TCP_IP.htm>. Acesso em: 11 ago. 2020.

OLAH, C. **Understanding LSTM Networks**. Disponível em:
<<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Acesso em: 13 ago. 2020.

OLIVEIRA, E. V. DE. **Análise de Desempenho de Método Baseado em Rede LSTM para Classificação de Falhas em um Processo de Controle de Nível**. 2020. 78 f. Universidade Federal do Rio Grande do Norte, 2020.

PAING, S. Y. BROADBAND REMOTE ACCESS SERVER. **IHS Jane's Defence Weekly**,
v. 53, n. 16, p. 24–27, 2016.

PANDAS Library. **Pandas**. Disponível em: <<https://pandas.pydata.org/>>. Acesso em: 13 ago.
2020.

PETER, S.; RUSSEL, N. **Inteligência artificial**. 3. ed. São Paulo: GEN LTC, 2013.

PILLAI, N. S. R. et al. Satellite image classification using self organizing map and ensemble classifiers. **2019 IEEE International Conference on System, Computation, Automation and NETWORKING, ICSCAN 2019**, p. 1–5, 2019.

PINHEIRO, J. M. DOS S. **Redes ópticas de acesso em telecomunicações**. 1. ed. Rio de Janeiro: Elsevier, 2016.

HASHMI, U. DARBANDI, A. IMRAN, A. Enabling proactive self-healing by data mining network failure logs, **2017 International Conference on Computing, Networking and Communications (ICNC)**, Santa Clara, CA, pp. 511-517, 2017.

<https://doi.org/10.1109/ICCNC.2017.7876181>

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, p. 23, 1958.

<https://doi.org/10.1037/h0042519>

SANTOS, M. T. et al. **Gerência de Redes de Computadores**. Rio de Janeiro: Escola Superior de Redes RNP, 2015.

SELMANOVIC, F.; SKALJO, E. **GPON in telecommunication network**. 2010, Moscou: ICUMT, 2010. p. 1012–1016. <https://doi.org/10.1109/ICUMT.2010.5676500>

SHARMA, A.; SHARMA, M. K.; DWIVEDI, R. K. Analyzing facebook data set using self-organizing map. **Proceedings of the 2018 International Conference on System Modeling and Advancement in Research Trends, SMART 2018**, p. 109–112, 2018.

<https://doi.org/10.1109/SYSMART.2018.8746984>

SILVA, R. F. DA. **A qualidade do serviço de acesso a internet para os consumidores brasileiros: um estudo sobre os elementos mais relevantes para a qualidade percebida pelos clientes**. São Paulo: Fundação Getúlio Vargas, 2020.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas**. 2. ed. São Paulo: Artliber, 2010.

SOLUÇÕES, I. Tecnologia GPON substitui cabeamento estruturado na transmissão de dados. **i3CSoluções**. Disponível em: <<https://i3csolucoes.com.br/tecnologia-gpon-substitui-cabeamento-estruturado-na-transmissao-de-dados/>>. Acesso em: 2 ago. 2020.

SOUSA, R. T. DE. Redes Neurais Recorrentes. **Deep learning Brasil**. Disponível em: <www.deeplearningbrasil.com.br>. Acesso em: 2 out. 2020.

PREVIEW, Scopus. Sources. Scopus. Disponível em: < <https://www.scopus.com/sources>>.

Acesso em: 13 dez. 2020.

TANENBAUM, A. S. **Redes de computadores**. 5. ed. São Paulo: Pearson, 2011.

TEIXEIRA, J. DE F. T. **O que é inteligência artificial**. Brasília: Editora Brasiliense, 1990.

TIAN, Y. et al. A Recommender System for Metaheuristic Algorithms for Continuous Optimization Based on Deep Recurrent Neural Networks. **Journal of IEEE transactions on artificial intelligence**. p 1-15. 2020.

TELES VIEIRA, F. H.; LEMOS, R. P.; LEE, L. L. Alocação dinâmica de taxa de transmissão em redes de pacotes utilizando redes neurais recorrentes treinadas com algoritmos em tempo real. **IEEE Latin America Transactions**, v. 1, n. 1, p. 70–75, 2003.

WANG, B.; QIN, Y.; HUANG, Q. The prospective of library spurred by artificial intelligence-China survey. **Proceedings - 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering, AEMCSE 2020**, p. 175–179, 2020. <https://doi.org/10.1109/AEMCSE50948.2020.00044>

WANG, J. et al. One new BRAS model based on Trusted Network. **Proceedings of the International Symposium on Electronic Commerce and Security, ISECS 2008**, p. 398–401, 2008. <https://doi.org/10.1109/ISECS.2008.62>

WELCOME to Python.org. **Python**. Disponível em: <<https://www.python.org/>>. Acesso em: 13 ago. 2020.

ZAPATER, M. et al. Self-organizing maps versus growing neural gas in detecting anomalies in data centres, **Logic Journal of the IGPL**, vol. 23, p 495-505, 2015.

<https://doi.org/10.1093/jigpal/jzv008>

ZHANG, Q. et al. Dynamic Neural Network Enabled 50 Gb/s PAM-4 IM/DD Transmissions Based on 10G-Class Optical Devices. **2019 18th International Conference on Optical**

Communications and Networks, ICOCN 2019, p. 1–3, 2019.

<https://doi.org/10.1109/ICOCN.2019.8934670>

YANG, P. et al. An Outlier Detection Approach Based on Improved Self-Organizing Feature Map Clustering Algorithm, **IEEE Access**, vol. 7, p 1-13, 2019.

<https://doi.org/10.1109/ACCESS.2019.2922004>

ANEXO A – DADOS E CÓDIGO APLICADO PARA APLICAÇÃO SOM

Os dados utilizados, tabelas, e demais figuras e dados detalhados estão disponíveis no seguinte [link](https://drive.google.com/drive/folders/1UF9biKRFhpJsLpTAXhTa6eyM2K7A3x1d?usp=sharing)
<<https://drive.google.com/drive/folders/1UF9biKRFhpJsLpTAXhTa6eyM2K7A3x1d?usp=sharing>>

Código SOM aplicado neste trabalho:

```
#Importação das bibliotecas utilizadas
import numpy as np #biblioteca matemática
import matplotlib.pyplot as plt #biblioteca para elaboração de gráficos
import pandas as pd #biblioteca para manipulação de dados como dataset

#Importação da fonte de dados no formato de dataset
dataset = pd.read_csv('massiva.csv') #massiva é o nome dado relatório utilizado

X= dataset.iloc[:, :-1].values #Subconjunto x, conjunto menos coluna 'massivo', sera
utilizado para treinamento nao supervisionado
Y= dataset.iloc[:, -1].values #Subconjunto y, coluna 'massivo', sera utilizado para
comparar ao teste e observar os outliers.

#Normalização dos dados para melhor desempenho devido a qtde dimensoes
from sklearn.preprocessing import MinMaxScaler
sc=MinMaxScaler(feature_range = (0,1)) # Delimitado entre 0 e 1
X = sc.fit_transform(X)

#Treinamento SOM
from minisom import MiniSom

#Nas configurações do SOM temos, x e y são as dimensões (10x10, 20x20), em seguida
temos imput len
```

#input len aqui será 15 que vem de X, sigma é o raio das diferentes periferias. A taxa de aprendizado será 1

#hiper parâmetro decide quanto os pesos são atualizados durante cada iteração, maior a taxa menor o aprendizado e mais tempo para construção.

#A função de decaimento melhora a convergência.

```
som=MiniSom(x = 10, y = 10, input_len = 15, sigma=1, learning_rate=0.5)
```

#Para inicialização precisamos inicializar os pesos com valores próximos a 0

```
som.random_weights_init(X) #inicia de forma randomica os pesos para o treinamento
```

```
som.train_random(data=X, num_iteration = 100) #O treinamento é a repetição do laço de treinamento definindo a quantidade de vezes que será iterado
```

#Apresentação do mapa auto organizável

```
from pylab import bone, pcolor, colorbar, plot, show
```

```
bone()
```

```
pcolor(som.distance_map().T) #Media das distancias internacionais através de cores usando matriz de vencedores Transposta.
```

```
colorbar() #Quanto maior a média das distâncias mais próximo do branco
```

#Quanto mais escuro mais ao centro, ou seja, a vizinhança se forma em torno do ponto mais escuro, mais branco significa outliers.

```
markers=['o','s']
```

```
colors=['r','g']
```

#Marcação no mapa

```
for i, x in enumerate(X): #x será o vetor de cada cliente
```

```
w = som.winner(x) #w coordenadas do no vencedor, 0.5 ajusta o marcador ao centro do quadro.
```

```
plot(w[0] + 0.5,
```

```
      w[1] + 0.5,
```

markers[Y[i]], #Y possui a variável dependente de como foi classificado a situação original, o código vai comparar com esse dado e marcar.

```
markeredgecolor = colors[Y[i]],  
markerfacecolor = 'None',  
markersize=10, #tamanho do marcador  
markeredgewidth= 2) #largura da aresta do marcador
```

```
show()
```

```
#Listando outliers
```

```
mappings = som.win_map(X)
```

```
#outliers = np.concatenate((mappings[(7,1)],mappings[(8,0)]),axis=0) #usar para + de  
#1 quadro de outlier (nó)
```

```
outliers=mappings[(9,9)] #usar para 1 quadro outlier (nó)
```

```
outliers = sc.inverse_transform(outliers)
```

```
#lista de outliers
```

```
print('Outliers listados (x,y)')
```

```
for i in outliers[:, 0]:
```

```
    print(int(i))
```

ANEXO B – DADOS E CÓDIGO APLICAÇÃO RNN

Para o teste utilizando RNN foi considerado a seguinte tabela de dados de entrada, usando um arquivo csv.

Os códigos e arquivos usados, e resultados coletados estão disponíveis no seguinte link <<https://drive.google.com/drive/folders/1JUJ7-sbgREu6Fo6FEdnTf9g9Iw3qddqp?usp=sharing>>

Código aplicado

```
# Recurrent Neural Network LSTM
# Etapa 1 - Processamento de dados

# Importar bibliotecas
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import math

#Bibliotecas de ajuste de escala de dados
from sklearn.preprocessing import MinMaxScaler

# Bibliotecas aplicadas a arquitetura RNN
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout

# Importação dos dados de treinamento
dataset_train = pd.read_csv('BRAS_Train.csv')
training_set = dataset_train.iloc[:, 1:2].values

# Feature Scaling
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
```

```

# Criar estruturas com 60 timesteps e 1 output
X_train = []
y_train = []
for i in range(60, 1258):
    X_train.append(training_set_scaled[i-60:i, 0])
    y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)

# Reshaping
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))

# Part 2 - Construindo a RNN

# Inicializando a RNN
regressor = Sequential()

# Adicionando a primeira camada LSTM e Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1],
1)))
regressor.add(Dropout(0.2))

# Adicionando a segunda camada LSTM e Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adicionando a terceira camada LSTM e Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adicionando a quarta camada LSTM e Dropout regularisation

regressor.add(LSTM(units = 50))

```

```

regressor.add(Dropout(0.2))

# Adicionando a output layer
regressor.add(Dense(units = 1))

# Compilando a RNN
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

# Fitting RNN ao treino
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)

# Part 3 - Fazendo as previsões e apresentando resultados #
dataset_test = pd.read_csv('BRAS_Test.csv')
real_bras_measure = dataset_test.iloc[:, 1:2].values

# Predição de 1 semana
dataset_total = pd.concat((dataset_train['Date_Hour'], dataset_test['In']), axis = 0)
inputs = dataset_total[len(dataset_total) - len(dataset_test) - 60:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
X_test = []
for i in range(60, len(dataset_test)):
    X_test.append(inputs[i-60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
predicted_bras_traffic = regressor.predict(X_test)
predicted_bras_traffic = sc.inverse_transform(predicted_bras_price)

# Visualização dos resultados
plt.plot(real_bras_measure, color = 'red', label = 'Real BRAS Traffic')
plt.plot(predicted_bras_traffic, color = 'blue', label = 'Predicted BRAS traffic')
plt.title('BRAS traffic')
plt.xlabel('Time')

```

```
plt.ylabel('BRAS traffic prediction')  
plt.legend()  
plt.show()
```