



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
DIRETORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES

WELLINGTON DANTAS DE ALMEIDA

**MÉTODOS DE APRENDIZAGEM DE MÁQUINAS
PARA DETECÇÃO DE ERRO EM COMUNICAÇÃO DE DADOS**

Fortaleza - Ceará

Junho de 2021

WELLINGTON DANTAS DE ALMEIDA

MÉTODOS DE APRENDIZAGEM DE MÁQUINAS
PARA DETECÇÃO DE ERRO EM COMUNICAÇÃO DE DADOS

Dissertação submetida à Coordenação
de Pós-Graduação em Engenharia de
Telecomunicações do Instituto Federal do Ceará,
como requisito parcial para obtenção do grau de
Mestre em Engenharia de Telecomunicações.

Área de pesquisa: Inteligência Artificial

Orientador: Dr. Ajalmar Rêgo da Rocha Neto

Coorientador: Dr. Francisco José Alves de Aquino

Fortaleza - Ceará
Junho de 2021

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

A447m Almeida, Wellington Dantas de.
MÉTODOS DE APRENDIZAGEM DE MÁQUINAS PARA DETECÇÃO DE ERRO EM
COMUNICAÇÃO DE DADOS / Wellington Dantas de Almeida. - 2021.
80 f. : il.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Engenharia de Telecomunicações,
Campus Fortaleza, 2021.

Orientação: Prof. Dr. Ajalmar Rego da Rocha Neto.

Coorientação: Prof. Dr. Francisco José de Aquino.

1. Novo método de detecção de erros. 2. Classificação com opção de rejeição. 3. Máquina de aprendizado extremo. I. Título.

CDD 621.382



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
TELECOMUNICAÇÕES

WELLINGTON DANTAS DE ALMEIDA

Esta Dissertação foi julgada adequada para a obtenção do Grau de Mestre em Engenharia de Telecomunicações, sendo aprovada pela Coordenadoria de Pós-Graduação em Engenharia de Telecomunicações, Campus Fortaleza do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

Prof. Dr. Ajalmar Rêgo da Rocha Neto
Instituto Federal do Ceará - IFCE
(Orientador)

Prof. Dr. Francisco José Alves de Aquino
Instituto Federal do Ceará - IFCE
(Coorientador)

Prof. Dr. João Paulo Pordeus Gomes
Universidade Federal do Ceará - UFC

Fortaleza, 09 de Junho de 2021

Este trabalho é dedicado à minha família, meus professores, meus amigos e a todos aqueles que buscam o conhecimento.

Agradecimentos

A Deus, acima de tudo.

Ao meu orientador, Prof. Dr. Ajalmar R. Rocha Neto, pelos ensinamentos, amizade, confiança e dedicação depositados ao longo deste trabalho. Seu conhecimento, ajuda e paciência foram fundamentais para o sucesso desta etapa.

Ao meu coorientador, Prof. Dr. Francisco José A. Aquino, por me auxiliar na vida acadêmica, pelos feedbacks e contribuições importantes, desde os primeiros semestres da graduação ao mestrado. Agradeço pela confiança depositada em mim.

Aos meus pais e minha irmã pelo apoio incondicional, conselhos, força, amor e amizade sem igual durante todos esses anos. Sem eles nada disto seria possível.

A todos os professores do mestrado, em especial os professores Dr. Daniel Xavier, Dr. Regis C. P. Marques e Dr. Auzuir R. de Alexandria, e a todos aqueles que de alguma forma contribuíram para minha formação.

Ao professor avaliador deste trabalho Dr. João Paulo Pordeus Gomes, agradeço pelo seu tempo em avaliar a contribuição deste trabalho.

A todos os meus amigos de pesquisa do Laboratório de Inteligência Artificial (LIA) e Processamento Digital de Sinais (PDS). E a todos os amigos do mestrado em Engenharia de Telecomunicações e Ciências da Computação.

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) pelo suporte financeiro.

“A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original”.

Albert Einstein

Resumo

O correto recebimento das informações de bits ‘0’ ou ‘1’ é uma tarefa importante e um dos principais problemas dos sistemas de recebimento de dados. Como os canais dificilmente são ideais a ocorrência de erros de bits pode afetar a qualidade da mensagem recebida. Uma solução é na implementação de mecanismos de detecção de erros para permitir a identificação dos bits corrompidos. Por ser um problema binário, algoritmos de classificação são eficientes em definir a saída como -1 ou +1 com base nas suas estimativas. Porém, até onde sabemos, as abordagens existentes ainda dependem da aplicação de detectores de erro que inserem informações redundantes e aumentam o custo computacional para verificação de pacotes, porque o classificador sozinho não consegue dar a certeza em relação à informação recebida de um canal. Com isso, podemos aplicar o método de opção de rejeição aplicado a algoritmos de classificação de padrões para automatizar o processo de detecção de erros sem o uso das abordagens existentes. Desse modo, procuramos desenvolver um método de detecção de erro que (i) use a classificação de padrões para fazer o papel de um equalizador dos canais de comunicação; (ii) use a técnica da opção de rejeição para calcular custos de risco mínimo de erro; (iii) propor um verificador de erros simples chamado Detecção de Erro Neural Independente (INED) e sua variante Detecção de Erro Neural Combinado (CNED), que são simples e eficientes. Os métodos INED e CNED são desenvolvidos com base na técnica da opção de rejeição. Os métodos desenvolvidos são testados em seis ambientes ruidosos diferentes e são comparados aos algoritmos de detecção de erros existentes. Os resultados indicam que os métodos independentes e combinados são comparáveis e em alguns casos superiores aos detectores conhecidos. Além disso, este trabalho apresenta resultados com a rede neural Máquina de Aprendizado Extremo (ELM) que mostrou apresentar resultados mais eficientes em comparação aos outros algoritmos usando a opção de rejeição.

Palavras-chave: Novo método de detecção de erros; Classificação com opção de rejeição; Máquina de aprendizado extremo.

Abstract

The correct receipt of bit information '0' or '1' is an important task and one of the main problems of data receiving systems. As the channels are hardly ideal, the occurrence of bit errors can affect the quality of the received message. One solution is to implement error detection mechanisms to allow the identification of corrupted bits. As a binary problem, classification algorithms are efficient in defining the output as -1 or +1 based on their estimates. However, as far as we know, the existing approaches still depend on the application of error detectors that insert redundant information and increase the computational cost for verifying packets because the classifier alone is not able to provide certainty regarding the information received from a channel. Thus, we can apply the rejection option method applied to pattern classification algorithms to automate the error detection process without using existing approaches. In this way, we seek to develop an error detection method that (i) uses the pattern recognition classifier to play the role of an equalizer in the communication channels; (ii) use the rejection option technique to calculate minimum risk of error costs; (iii) propose a simple error checker called Independent Neural Error Detection (INED) and its variant Combined Neural Error Detection (CNED), which are simple and efficient. The INED and CNED methods are developed based on the rejection option technique. The developed methods are tested in six different noisy environments and are compared to the existing error detection algorithms. The results indicate that the independent and combined methods are comparable and in some cases superior to the known detectors. In addition, this work presents results with the ELM (Extreme Learning Machine) algorithm, which has shown to present more efficient results in comparison to the other algorithms using the rejection option.

Keywords: New error detection method; Classification with rejection option; Extreme learning machine.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

Lista de Abreviações

1	Introdução	20
1.1	Motivação para a Dissertação	23
1.2	Objetivo Geral	23
1.3	Objetivos Específicos	23
1.4	Organização da Dissertação	24
2	Fundamentação Teórica	25
2.1	Códigos Detectores de Erro	25
2.1.1	Single Parity Check (SPC)	26
2.1.2	CHECKSUM	27
2.1.3	Cyclic Redundancy Check (CRC)	28
2.2	Classificadores de Padrões	31
2.2.1	Breve História sobre Inteligência Artificial	31
2.2.2	Multilayer Perceptron (MLP)	31
2.2.3	Radial Base Function (RBF)	33
2.2.4	Extreme Learning Machine (ELM)	34

2.2.5	Naive Bayes classifier (Bayes)	36
2.3	Classificação com Opção de Rejeição	37
2.3.1	Regra da Decisão Ótima	38
2.3.2	Abordagens para Classificação Binária com Opção de Rejeição	42
2.3.3	Abordagem com um Único Classificador Padrão	44
2.4	Conclusão	45
3	Método Proposto	48
3.1	Definição do INED e CNED	48
3.2	Desenvolvimento do Modelo de Detecção de Erro usado no INED e CNED	49
3.2.1	Etapa de Treinamento do Limiar de Decisão	49
3.2.2	Etapa de Predição	50
3.2.3	Etapa de Detecção do Erro	51
3.3	Diagrama do Modelo Geral	52
3.4	Exemplo da Detecção de Erro Neural Independente (INED)	52
3.5	Exemplo do Detector de Erro Neural Combinado (CNED)	53
3.6	Conclusão	55
4	Experimentos e Resultados	56
4.1	Experimentos	56
4.1.1	Base de Dados dos Experimentos	56
4.1.2	Testes Realizados	58
4.1.3	Validação Cruzada e Parâmetros da Classificação	59
4.1.4	Simulação Computacional em Hardware	60
4.1.5	Constituição do Frame de Comunicação	61
4.2	Resultados	63
4.2.1	Análise Exploratória dos Dados	63

4.2.2	Comparação de Desempenho entre os Classificadores de Padrões com Opção de Rejeição	65
4.2.3	Resultados dos Algoritmos de Detecção de Erro com ELM	66
4.2.4	A Eficiência da Classificação com Opção de Rejeição em Detectar Erro de Bit	69
4.2.5	Comparação de Detecção de Erro entre SPC, CHECKSUM, CRC e INED	70
4.2.6	Detecção de Erro com CNED	72
4.2.7	Resultados da Simulação em Hardware	73
5	Conclusão e Trabalhos Futuros	75
	Referências Bibliográficas	76

Lista de Figuras

1	Diagrama representando o processo de comunicação de dados. TX (transmissão) e RX (recepção).	22
2	Modelo de codificação utilizando o código SPC para paridade ímpar e para paridade par (cor vermelha).	26
3	Exemplo de erro ocasionado na recepção (cor azul) pelo canal ou por ruído, porém identificado pelo código SPC para paridade ímpar e para paridade par (cor vermelha).	27
4	Exemplo com dois bits de erro ocasionado na recepção (cor azul) pelo canal ou por ruído, e não identificado pelo código SPC para paridade ímpar e para paridade par (cor vermelha).	27
5	Exemplo de codificação com a técnica CHECKSUM, preparação e cálculo para envio da mensagem, com o valor acrescido do checksum (cor vermelho).	28
6	Exemplo de decodificação com a técnica CHECKSUM, processo de cálculo aritmético e detecção de erro (cor verde).	29
7	Representação do cálculo de verificação do código polinomial para o exemplo de codificação usando a técnica CRC.	30
8	Arquitetura de uma rede neural MLP.	32
9	Regiões de rejeição e aceitação da regra ótima de reconhecimento. Fonte: (ROCHA NETO, 2011).	40
10	Ilustração da curva de compromisso entre o erro e a rejeição. Fonte: (ROCHA NETO, 2011).	40
11	Ilustração da curva hipotética de compromisso entre o erro e a rejeição. Fonte: (ROCHA NETO, 2011).	41

12	(a) Classificação binária convencional que divide o espaço em duas regiões: C1 ou C2; (b) classificação binária com opção de rejeição que divide o espaço em três regiões: C1, C2 ou Rejeição.	42
13	Regiões de decisão obtidas pela utilização de um único classificador para um problema binário. Fonte: (SOUSA; MORA; CARDOSO, 2009).	43
14	Regiões de decisão obtidas pela utilização de dois classificadores independentes para um problema binário. Fonte: (SOUSA; MORA; CARDOSO, 2009) (modificado).	44
15	Variação do tamanho da região de rejeição para o limiar igual a zero (Classificação convencional) e limiar diferente de zero (Opção de rejeição). . .	45
16	Taxa de acerto e taxa de rejeição para um certo limiar e n realizações. Fonte: (OLIVEIRA, 2016) (modificado).	46
17	Exemplo de escolha de melhor desempenho quando $W_r = 0.24$. Fonte: (OLIVEIRA, 2016).	46
18	Diagrama que representa o processo de treinamento, predição e detecção de erros com o método INED.	52
19	Diagrama representando a detecção de erros pelo método INED na transmissão de mensagens de dados.	53
20	Diagrama representando a detecção de erros pelo método CNED na transmissão de mensagens de dados.	54
21	Representação de convolução entre canal e sinal [Y] e formação do conjunto de padrões [X D].	57
22	Representação da influência da relação sinal-ruído na mensagem do bloco que é transmitido.	58
23	Ilustração representando a simulação de transmissão binária entre Tx e Rx. . .	60
24	Módulos utilizados na simulação para comunicação sem fio. Fonte:(FILIFELOP,)	61
25	Conjunto transmissor e receptor do sistema de comunicação sem fio.	61
26	Esquema de enquadramento usado na comunicação sem fio.	61

27	Projeto de implementação dos detectores de erro na formação da informação útil usado na transmissão.	62
28	Representação visual do conjunto de padrões binário variando de 0 a 15 dB. . .	64
29	Curvas de desempenho A-R com SNR variando de 0 dB a 15 dB.	66
30	Monitor serial do Arduino: Análise dos resultados de saída do receptor. . . .	74

Lista de Tabelas

1	Polinômios de CRC mais comuns.	29
2	Função de ativação comumente usada no ELM.	36
3	Descrição dos datasets.	56
4	Característica do módulo RF 433MHz	62
5	Balanceamento dos padrões das classes do dataset Data1.	64
6	Acurácia e desvio padrão de classificadores de padrões (%).	67
7	Acurácia e média de rejeição em classificadores de padrões com opção de rejeição (%).	67
8	Tempo em segundos da performance para 1 realização e com SNR em 0 dB. . .	67
9	Acurácia e desvio padrão de simulação de transmissão com SPC (%).	68
10	Acurácia e desvio padrão de simulação de transmissão com CHECKSUM (%). .	68
11	Acurácia e desvio padrão de simulação de transmissão com CRC (%).	69
12	Eficiência do ELM com a técnica de opção de rejeição em detectar erros na transmissão de bits (50 realizações).	70
13	Eficiência do ELM com a técnica de opção de rejeição em detectar erros na transmissão de bits (1 realização).	70
14	Acurácia e desvio padrão de simulação de transmissão com INED (%).	71
15	Desempenho dos detectores SPC e INED sob diferentes condições de ruído. . .	71
16	Desempenho para os detectores CHECKSUM e INED sob diferentes condições de ruído.	72
17	Desempenho para os detectores CRC-8-CCITT e INED sob diferentes condições de ruído.	72
18	Acurácia e desvio padrão de simulação de transmissão com SPC-CNED (%). .	73

19	Desempenho dos detectores SPC e SPC-CNED sob diferentes condições de ruído.	73
20	Simulação de transmissão de 250 blocos no teste prático (média de 215 certos e 35 errados).	74

Lista de Símbolos

Z	variável aleatória
$\mathbb{R}$	conjunto dos números reais
t	tempo contínuo
n	tempo discreto
$f(z)$	função densidade de probabilidade
$F(z)$	função de distribuição acumulada
σ	desvio padrão
μ	média ou esperança matemática
∇	operador gradiente
Σ	operador de somatório
Π	operador de introdutório
$\sqrt{}$	operador de raiz quadrada
$\int$	operador de integral
dx	derivada da função
x_i	atributos do padrão
b_i	bias do padrão
$\sqrt{}$	raiz quadrada
t	limiar de rejeição
$E(t)$	taxa de erro para um limiar
$R(t)$	taxa de rejeição para um limiar

$A(t)$	taxa de acerto para um limiar
W_r	custo de rejeição
W_c	custo de acertar
W_e	custo de errar
X	conjunto de pontos de entrada
Y	conjunto de pontos de saída
$g(\cdot)$	função de ativação
w	pesos da camada de entrada
β_i	pesos da camada de saída
H	matriz de saída do ELM
$H^\dagger$	Moore–Penrose inversa
C_{+1}	classe de um padrão positiva
C_{-1}	classe de um padrão negativa

Lista de Abreviações

SPC	Single Parity Check
CRC	Cyclic Redundancy Check
MLP	Miltilayer Perceptron
RBF	Radial Base Function
ELM	Extreme Learning Machine
ISS	Interferência intersimbólica
IDE	Integrated Development Environment
LMS	Least Mean Square
RLS	Recursive Least Square
DFE	Decision Feedback Equalizer
EDA	Exploratory Data Analysis
ASK	Amplitude Shift Keying

INTRODUÇÃO

A demanda por sistemas de comunicação, que transmitem grande quantidade de informações em alta velocidade e confiabilidade, representa um grande desafio. Canais ruidosos sujeitos a interferência intersimbólica (ISI), atenuação de sinal e ruído degradam a qualidade da mensagem transmitida, causando muitos erros de bit que são recebidos pelo receptor. No processo real de comunicação do computador, a transmissão de informações estará inevitavelmente sujeita a um certo grau de interferência (ZHONG; HU, 2020). A demanda pela aplicação de métodos eficientes de detecção de erros tem aumentado na última década (KWON; LEE, 2019). Assim, quase todos os sistemas de comunicação adotam controle de detecção de erros para melhorar a qualidade da transmissão de dados na linha de comunicação (ALRSHAH *et al.*, 2015).

No campo de detecção de erros, os algoritmos mais utilizados em sistemas de comunicação são o Código de Paridade Simples (SPC), Verificador de Redundância Cíclica (CRC) e Soma de Verificação (CHECKSUM). Nestes algoritmos, bits redundantes são adicionados na etapa de codificação da mensagem, causando o aumento do pacote. Na recepção, os pacotes são novamente verificados com detectores de erros e comparados. Se forem iguais, o pacote de dados continuará a ser recebido de acordo com o protocolo. Se houver um erro, o pacote deve ser enviado novamente. Os códigos de detecção de erros não eliminam a possibilidade de erros nas mensagens de dados enviadas para o receptor. Isso ocorre porque os próprios códigos de controle estão sujeitos à corrupção por ruído. Dependendo da relação sinal-ruído (SNR), nenhum dos métodos retorna uma precisão meticulosa. Porém, a detecção de erros do algoritmo melhora os indicadores da taxa de erro de bit (BER).

Algoritmos de detecção de erro são constantemente estudados e aplicados em soluções reais visando a diminuição da taxa de erro e garantindo que a informação recebida seja confiável. Trabalhos recentes sobre a aplicação de algoritmos de detecção de erros são apresentados em: (ZHONG; HU, 2020) que adiciona a implementação do CRC em diminuir a taxa de erro de bit em soluções com IoT; em (DONG; HE, 2020) são criadas estratégias e métodos de modificação do CRC visando aprimoramento do algoritmo; em (HUO; LIU, 2018) apresenta um design de

comunicação de alto desempenho e velocidade com a aplicação de algoritmos de detecção de erro e de criptografia; em (AYDOS; FEY, 2015) apresenta um software com a implementação de paridade com modificações e economiza entre 29% a 35% a sobrecarga de redundância.

Algoritmos de aprendizado de máquina, mais especificamente, os de redes neurais são um conjunto de algoritmos usados para classificações e regressões de padrões. Em problemas envolvendo equalização, os algoritmos de aprendizado de máquina podem substituir os equalizadores porque filtram o sinal recebido classificando suas amostras geradas após a passagem pelo canal. Um dos algoritmos utilizados para predição do sinal é o ELM, muito utilizado neste trabalho de dissertação por causa da sua eficiência e rapidez, é também um dos modelos de aprendizado de máquina mais recentes. Foi proposto por Huang (HUANG; ZHU; SIEW, 2006), tornou-se um dos algoritmos de redes neurais feedforward mais populares e pesquisados em inteligência artificial. É aplicado para sistemas que exigem treinamento rápido, alta generalização e uma alta taxa de precisão (ALENCAR; ROCHA NETO; GOMES, 2016).

Em 2017 foi anunciado pelo Google Scholar, que o trabalho de Huang (HUANG; ZHU; SIEW, 2006) no ELM apresentado em 2006, ficou em segundo lugar entre os dez artigos científicos de aprendizado de máquina mais citados daquele ano, o que mostra que este algoritmo de inteligência artificial teve boa aceitação pela comunidade científica e que muitos estudos têm aplicado a rede neural ELM. Entre os trabalhos relacionadas como um equalizador de canais foram encontrados trabalhos relatados em (LIM, 2008) que apresenta um equalizador ELM que usa o método de mínimos quadrados de dados (DLS), onde a melhoria nos problemas de equalização do canal de desempenho alcançou resultados eficientes. Em (YANG *et al.*, 2018) apresenta uma estrutura do ELM para aplicar na equalização cega de sistemas de transmissão usando modulação em amplitude em quadratura (QAM). Em (LIU *et al.*, 2019) propôs a mesma estrutura de equalização que (YANG *et al.*, 2018), mas usando sistemas OFDM.

No processo de envio de dados, o sinal passa por um canal barulhento que pode distorcer sua forma, causando interferência entre símbolos (ZHAO *et al.*, 2011a), (ZHAO *et al.*, 2011b). No entanto, um processo de filtragem no sinal é feito para recuperá-lo, com o emprego de equalizadores lineares ou não-lineares que têm a função de compensar o sinal não ideal. Os equalizadores são eficientes em compensar o sinal não ideal, principalmente o não-linear, pois é adaptativo para ajustar os coeficientes com a variação do canal (SU; CHENG; YU, 2010). Exemplos de equalizadores, como o *Decision Feedback Equalizer* (DFE) e o *Linear Transversal Equalizer* (LTE), podem ser usados com os algoritmos *Recursive Least Squares* (RLS) e *Least Mean Squares* (LMS). A aplicação de classificadores de padrões (CP) pode ser uma boa estratégia, pois seus pesos se atualizam com a variação do canal. A Figura.1 apresenta

o modelo de comunicação em bloco para exemplificar os conceitos mostrados até aqui.

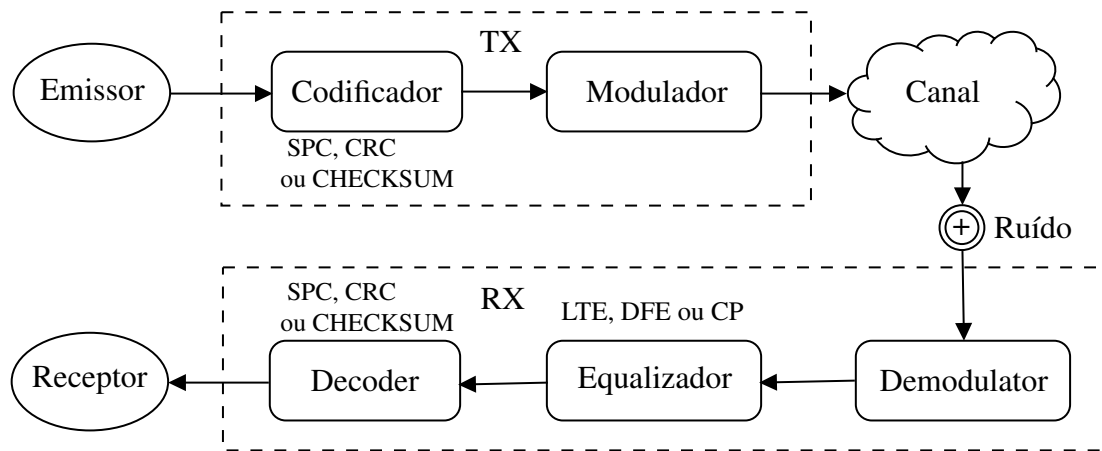


Figura 1: Diagrama representando o processo de comunicação de dados. TX (transmissão) e RX (recepção).

A classificação com opção de rejeição foi proposta por (CHOW, 1970), onde apresenta um classificador ideal que pode ser descrito pelo equilíbrio entre sua taxa de rejeição e a taxa de acerto. Uma estratégia de rejeição procura rejeitar padrões que têm uma alta chance de serem classificados incorretamente, para serem enviados para a avaliação de um especialista humano posteriormente ou, mais precisamente, sujeitos a outros métodos. Isso aumenta a possibilidade de que padrões não rejeitados possam ser classificados corretamente.

A aplicação da opção de rejeição aumenta consideravelmente a métrica de avaliação dos classificadores, mas o método não descarta a possibilidade de erro. Nas tarefas de classificação binária, as saídas de rede são codificadas para as saídas 0 ou 1; No entanto, em certas situações, a decisão pode ser adiada pelo sistema de rejeição. O desenvolvimento de classificadores binários com a terceira saída pode automatizar as decisões corretas (SOUSA *et al.*, 2015); em situações críticas a amostra pode ser repassada a um especialista humano para análise. A classificação com opção de rejeição está sendo aplicada em muitas áreas de pesquisa. Exemplos podem ser vistos para aplicações médicas (ROCHA NETO *et al.*, 2011), em classificação de imagens (CONDESSA *et al.*, 2013), em (SOUSA; MORA; CARDOSO, 2009) o método aprende a região de rejeição em dados complexos e em (MESQUITA *et al.*, 2016) é aplicado o método para previsão de defeitos de software.

Outros trabalhos sobre classificação com opção de rejeição foram aplicados a rótulos de aplicativos onde a precisão do classificador não é satisfatória (PILLAI; FUMERA; ROLI, 2013), em (AHMADI *et al.*, 2004) apresenta um classificador de notas e avalia a confiabilidade da nota pela probabilidade da classe vencedora através de um valor de limite de rejeição. A classificação binária é aplicada com a opção de rejeitar para otimização de perda convexa com o algoritmo

Support Vector Machines (SVM) (BARTLETT; WEGKAMP, 2008).

1.1 Motivação para a Dissertação

O presente trabalho tem como diferencial em relação aos citados anteriormente o fato de apresentar um detector de erro que implementa uma estratégia de rejeição de padrões. Os métodos se apresentam como uma abordagem inovadora, uma vez que não são encontrados trabalhos similares na literatura.

No trabalho de mestrado, introduzimos dois novos métodos para detecção de erros de bits, usando o ELM como modelo de aprendizado de máquina e aplicando classificação com a opção de rejeição para treinar custos e limiares de decisão e consequentemente, detectar bits vulneráveis. Os métodos consistem em calcular com um certo grau de certeza a chance dos bits serem bem classificados. Os métodos são chamados de INED e CNED e apresentam resultados promissores em relação às técnicas de detecção conhecidas.

1.2 Objetivo Geral

O objetivo geral deste trabalho é avaliar a aplicabilidade de técnicas de classificação com opção de rejeição usando redes neurais artificiais para criar um método de detecção de erro automático com base na regra da decisão ótima e, assim, verificar a ocorrência de erro na recepção de bits. Neste ponto, busca-se novas estratégias de detecção com desempenho igual ou superior às demais técnicas convencionais.

1.3 Objetivos Específicos

Os objetivos específicos deste trabalho são apresentados a seguir:

- Aplicar classificadores de padrões no problema de equalização de canais de comunicação;
- Utilizar a técnica de classificação com opção de rejeição na equalização de canais de comunicação;
- Avaliar o desempenho dos classificadores ao utilizar o processo de classificação com opção de rejeição, quando aplicados em problemas binários;

- Propor o método principal INED e sua variante CNED para detecção de ocorrência de erros em bits;
- Avaliar os métodos INED e CNED com outras abordagens por meio de simulações computacionais.

1.4 Organização da Dissertação

O restante deste trabalho é formado por quatro capítulos que descrevem os diversos conceitos relacionados à classificação com opção de rejeição e sua aplicação com algoritmos de aprendizado de máquina.

O Capítulo 2 contém uma fundamentação teórica em três seções: apresenta uma abordagem dos algoritmos detectores de erros conhecidos no mercado; apresenta os conceitos técnicos de classificadores de padrões e conteúdos relativos à classificação com opção de rejeição utilizando a regra da decisão ótima.

O Capítulo 3 é um capítulo de apresentação de propostas e métodos. Contém uma descrição sobre como unir duas técnicas: algoritmos de classificação e a técnica de opção de rejeição. E como a partir dessas duas técnicas podemos criar uma nova solução para detecção de erros de bit, neste capítulo são apresentados os conceitos e técnicas para implementação dos métodos simples, porém eficientes INED e sua variante CNED.

O Capítulo 4 apresenta a formulação dos experimentos realizados para avaliação dos métodos propostos. E apresenta os resultados e as discussões dos experimentos isolados.

O Capítulo 5 aborda algumas considerações finais sobre esta exposição, comentando e analisando os resultados obtidos. Ressalta as contribuições que auxiliaram na elaboração deste trabalho e as propostas para trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os aspectos relacionados a detecção de erro com classificadores de padrões e opção de rejeição. Uma visão geral sobre os métodos de detecção de erro tradicionais aplicados no mercado também são apresentados para justificar a criação de novo método de detecção de erro.

Este capítulo está organizado em quatro seções.

Na seção 2.1, são apresentados três algoritmos de detecção de erros mais comuns da literatura.

Na seção 2.2, são apresentados algoritmos de classificação de padrões aplicados na resolução de problemas.

Na seção 2.3, é apresentado a regra da decisão ótima de Chow e a regra da classificação com opção de rejeição.

Na seção 2.4, é apresentada a conclusão sobre o capítulo apresentado.

2.1 Códigos Detectores de Erro

Uma condição muito comum em transmissões de dados são a ocorrência de erros, isto é, bit 0 que muda para bit 1, ou o contrário. Os sinais digitais sofrem um ruído que pode introduzir erros nos bits binários que viajam do remetente para o receptor, para adicionar proteção contra erros em tais casos, são empregados códigos de detecção de erro.

Entre os mais conhecidos, podemos destacar os códigos de detecção de erro SPC (*Single Parity Check*), CRC (*Cyclic Redundancy Check*) e soma de verificação CHECKSUM. Seja qual for, todos esses três detectores têm algo em comum, todos precisam acrescentar bits nos pacotes de dados para futuros cálculos de confiabilidade dos dados. Com isso, o objetivo desta seção é apresentar um breve introdutório sobre esses códigos, pois eles são a base de comparação com os métodos desenvolvidos nesta dissertação.

2.1.1 Single Parity Check (SPC)

O código de verificação de paridade simples é um dos códigos de detecção de erros mais simples e fáceis de implementar. Resumidamente, um bit extra é adicionado a cada unidade de dados (por exemplo, byte, palavra, bloco, segmento, quadro, célula e ou pacote) do código original, isto é, k bits é alterada para uma palavra de código de n bits, em que $n = k + 1$. De forma que a nova palavra de código exibe sempre um número par ou ímpar de 1's, com o objetivo de tornar a sequência gerada (incluindo o bit de paridade) uniforme (PEDRONI, 2010).

Na implementação do SPC, a paridade par (onde a quantidade total de bits 1's deve ser par) é normalmente usada para transmissão síncrona e paridade ímpar (onde a quantidade total de bits 1's deve ser ímpar) para transmissão assíncrona (GRAMI, 2016). A distância mínima de Hamming para esta categoria é $D_{min} = 2$, por consequência, o código pode detectar um erro, porém não pode corrigi-lo.

Devido à sua simplicidade, a paridade é usada em muitos periféricos de hardware, como por exemplo o PS/2 (*personal system version 2*), para comunicação entre PCs e periféricos como teclado, mouse ou alto-falante. Onde a conexão pode ser repetida em caso de dificuldade ou onde a simples detecção do erro é útil.

No exemplo da Figura 2 são apresentados os dois modelos de transmissão utilizando o código SPC, com paridade ímpar e paridade par. Observa-se que como a palavra "M" de código binário 01001101_b , consiste em oito bits de dados (um byte), aos quais é acrescentado um bit de paridade. Como os bits de dados são um conjunto de bits par, quatro bits 0's e quatro bits 1's, são acrescentados e o bit igual a '1' para paridade ímpar e ou bit igual a '0' para paridade par.

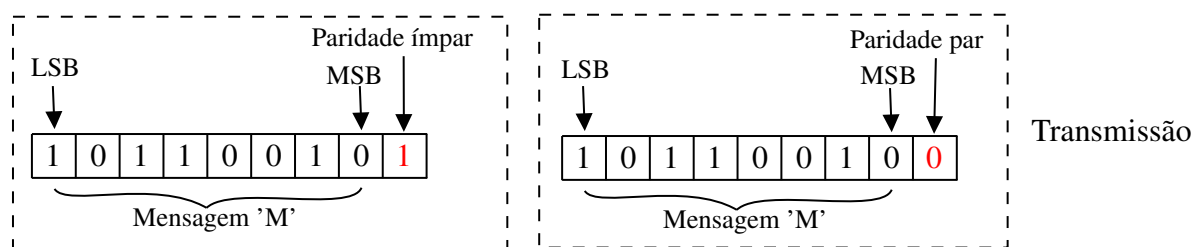


Figura 2: Modelo de codificação utilizando o código SPC para paridade ímpar e para paridade par (cor vermelha).

Para determinar se ocorreu um erro, o decodificador usa uma equação de verificação de paridade, que corresponde a uma adição de módulo 2 (XOR), como representado em:

$$c_1 \oplus c_2 \oplus c_3 \oplus c_4 \oplus c_5 \oplus c_6 \oplus c_7 \oplus c_8 \oplus = 1 \quad (\text{paridade ímpar}) \quad (2.1)$$

$$c_1 \oplus c_2 \oplus c_3 \oplus c_4 \oplus c_5 \oplus c_6 \oplus c_7 \oplus c_8 \oplus = 0 \quad (\text{paridade par}) \quad (2.2)$$

onde somente será verdadeira a mensagem que houver um número ímpar de 1's em c para paridade ímpar e somente será verdadeira a mensagem que houver um número par de 0's em c para paridade par.

A Figura 3 apresenta um caso onde um dos bits da mensagem 'M' chega errado na recepção, e é facilmente identificado pelo código SPC. Porém, na Figura 4 apresenta uma das principais limitações desta técnica de detecção de erro. Se a quantidade de erros for um número par o código SPC erroneamente identifica como certo.

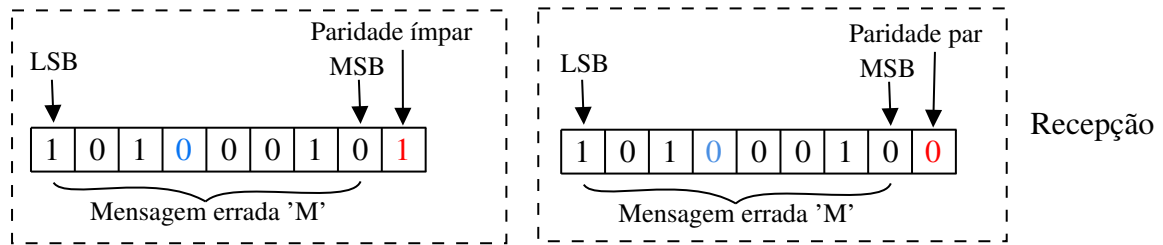


Figura 3: Exemplo de erro ocasionado na recepção (cor azul) pelo canal ou por ruído, porém identificado pelo código SPC para paridade ímpar e para paridade par (cor vermelha).

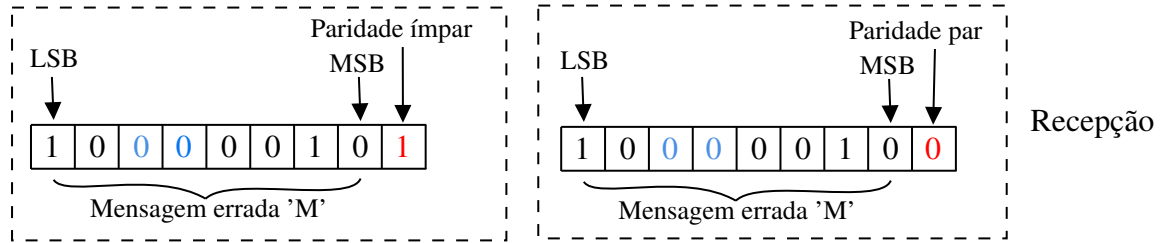


Figura 4: Exemplo com dois bits de erro ocasionado na recepção (cor azul) pelo canal ou por ruído, e não identificado pelo código SPC para paridade ímpar e para paridade par (cor vermelha).

2.1.2 CHECKSUM

A soma de verificação consiste em somar uma certa quantidade de palavras que, a serem transmitidas, as palavras de código são enviadas junto com o resultado invertido dessa soma (BARBARÁ; GOEL; JAJODIA, 2000), (SCHILLER *et al.*, 2008). Neste esquema, os dados são divididos em K palavras para cada um dos M bits e as palavras são adicionadas usando a aritmética do complemento 1 para obter a soma de verificação na codificação, a soma de verificação é invertida e adicionada no pacote para envio. Na recepção o segmento de bits

recebidos são novamente calculados e o valor aritmético é usado para obter a soma, se a soma do valor aritmético com o valor invertido for igual a zero, os dados recebidos são aceitos; caso contrário, é descartado.

Na Figura 5 é apresentado o modelo para codificação da soma de verificação, onde se acrescenta uma informação (checksum) que é enviada junto da mensagem de dados.

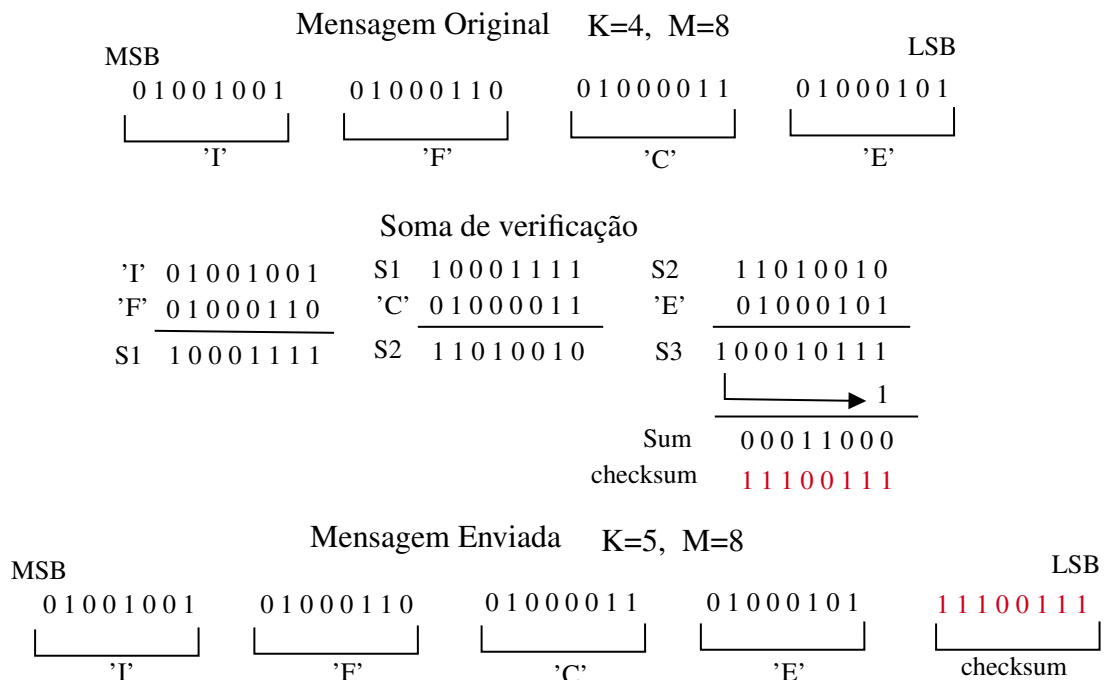


Figura 5: Exemplo de codificação com a técnica CHECKSUM, preparação e cálculo para envio da mensagem, com o valor acrescido do checksum (cor vermelho).

Na Figura 6 é apresentado o modelo inverso da codificação da soma de verificação, que consiste em realizar as mesmas etapas, somas aritméticas dos bits das palavras. Onde é somado a soma de verificação com o checksum e se o complemento desta soma for igual a zero, significa que a transmissão está correta.

2.1.3 Cyclic Redundancy Check (CRC)

O CRC é um dos métodos mais conhecidos para detectar erros durante a transmissão de dados. Diferente do CHECKSUM que é baseado na adição das palavras binárias, o CRC é baseado na divisão binária. A técnica CRC consiste em anexar no final de um pacote de dados, uma palavra binária para que a resultante se torne divisível por um segundo número binário predeterminado. De modo que o receptor faça os mesmos cálculos nos dados recebidos e então se nesta etapa não houver erros, o pacote de dados é considerado correto e portanto, aceito (HUO *et al.*, 2015).

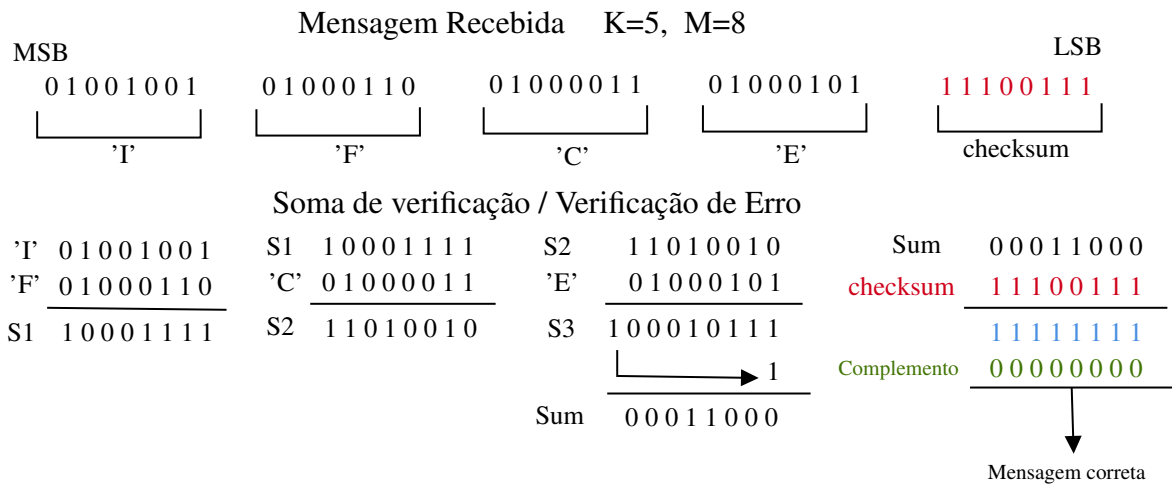


Figura 6: Exemplo de decodificação com a técnica CHECKSUM, processo de cálculo aritmético e detecção de erro (cor verde).

O código CRC é identificado por um polinômio gerador $G(x)$, na Tabela 1 são apresentado alguns dos padrões mais adotados, como o padrão Ethernet IEEE 802.3, que possui um valor de CRC de 32 bits (PEDRONI, 2010).

Tabela 1: Polinômios de CRC mais comuns.

CRC-8-CCITT (8 bits)	$g(x) = x^8 + x^2 + x + 1$
CRC-16-ANSI (16 bits)	$g(x) = x^{16} + x^{15} + x^2 + 1$
CRC-16-CCITT (16 bits)	$g(x) = x^{16} + x^{12} + x^5 + 1$
CRC-32-IEEE (32 bits)	$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Para entender o fluxo de cálculo para enviar informações utilizando o CRC é preciso visualizar a ilustração apresentada na Figura 7 e entender a sua formação seguindo os conceitos a seguir:

1. Temos a mensagem dados $M(x)$, o polinômio gerador $G(x)$, o valor CRC ou resto da divisão $R(x)$, o numerador da divisão $D(x)$ e o quociente da divisão $Q(x)$;
2. Uma informação em bits, pode ser melhor representado por um polinômio para explicação, exemplo $M(x) = 10101_b$ é o equivalente a $M(x) = x^4 + x^2 + 1$ em polinômio e com grau $n = 4$, o grau é o maior expoente;
3. O numerador da divisão $D(x)$ é formado pela mensagem dados $M(x)$ mais o grau n do polinômio gerador $G(x)$ como em:

$$M(x) = 1101011011 = X^9 + X^8 + X^6 + X^4 + X^3 + X + 1$$

$$G(x) = 10011 = X^4 + X + 1 \text{ (grau } n = 4\text{)}$$

$$D(x) = M(x) \cdot X^4 = X^{13} + X^{12} + X^{10} + X^8 + X^7 + X^5 + X^4$$

$D(x)$	$X^{13} + X^{12} + X^{10} + X^8 + X^7 + X^5 + X^4$ $\underline{X^{13} + X^{10} + X^9}$ $X^{12} + X^9 + X^8 + X^7 + X^5 + X^4$ $\underline{X^{12} + X^9 + X^8}$ $X^7 + X^5 + X^4$ $\underline{X^7 + X^4 + X^3}$ $X^5 + X^3$ $\underline{X^5 + X^2 + X}$ $R(x) \quad X^3 + X^2 + X \text{ (1110)}$	$\left \begin{array}{c} X^4 + X + 1 \\ \hline X^9 + X^8 + X^3 + X \end{array} \right. \begin{array}{l} G(x) \\ Q(x) \end{array}$
--------	--	---

XOR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

$$T(x) = 1101011011 \text{ 1110 (mensagem transmitida)}$$

Figura 7: Representação do cálculo de verificação do código polinomial para o exemplo de codificação usando a técnica CRC.

$$D(x) = M(x) \cdot x^n, \quad (2.3)$$

ou seja, os dados binários são aumentados com base no tipo de polinômio de CRC escolhido na implementação, com base no grau n do polinômio gerador, seguindo a regra prática de acrescentar ao $M(x)$ o valor de bits com base no grau n , onde são preenchidos com '0's;

4. O cálculo do fluxo de dados $D(x)$ é dividido pelo polinômio gerador $G(x)$, formando como resultados o quociente $Q(x)$ e o resto $R(x)$;

5. A mensagem enviada $T(X)$ é formada pelos bits de $M(x) + R(x)$.

Os códigos de CRC são considerados eficientes e têm alta implementação em padrões e protocolos da camada de enlace. Nos frames com padrão IEEE, o polinômio gerador $G(x)$ foi estabelecido de acordo com os organismos de padronização. No Ethernet 802.3 é seguido de um CRC de 32 bits (ROCHOL; ABREVIATURAS, 2012).

2.2 Classificadores de Padrões

Nesta seção são descritos todos os classificadores de padrões utilizados neste trabalho. São descritos três classificadores de padrão baseados em redes neurais e um classificador baseado em estatística.

2.2.1 Breve História sobre Inteligência Artificial

Os primeiros trabalhos sobre inteligência artificial começaram com as pesquisas do psiquiatra Warren McCulloch e do matemático Walter Pitts que nos anos de 1940 a 1950 introduziram os processos eletrônicos poderiam simular o comportamento de células neurais humanas. Assim nasceu o neurônio artificial. Nos anos seguintes vários pesquisadores estavam propondo modelos de redes neurais contendo modificações no modelo original de McCulloch e Pitts. Um dos trabalhos relevantes foi do cientista da computação Frank Rosenblatt, com a Regra de Aprendizado do Perceptron, onde poderia resolver problemas de reconhecimento de padrões lineares, assim nasceu o Perceptron Simples ou rede neural de uma única camada, introduzido entre 1956 a 1962. Este fato foi seguido por um longo período de acumulação de conhecimento. Porém, nas décadas de 70 a 80 que foi conhecido como o “inverno da inteligência artificial”, alguns problemas não-lineares não poderiam ser resolvidos, o que fez muitos pesquisadores abandonarem os estudos de IA.

Todavia, em 1986, um grupo de matemáticos conseguiram resolver problemas que até então não tinham solução. Com a limitação das redes neurais de uma única camada de não resolver problemas não-linearmente separáveis iniciou-se uma pesquisa de adicionar uma ou mais camadas intermediárias. Uma rede com um único neurônio na camada intermediária pode implementar qualquer função contínua, enquanto a utilização de dois neurônios da camada intermediária permite a aproximação de qualquer função (CYBENKO, 1989).

2.2.2 Multilayer Perceptron (MLP)

As redes Perceptron Multicamadas são compostas por três conjuntos de camadas: uma camada de entrada; uma ou mais camadas intermediárias (ocultas) e uma camada de saída. Na rede MLP a camada de entrada recebe os atributos do padrão. Na camada oculta os neurônios utilizam funções não-lineares e fazem interconexões entre si. Na camada de saída, também é composta por neurônios que podem ser lineares (ou não), esta camada devolve o(s) valor(es) que se espera encontrar (FATH; MADANIFAR; ABBASI, 2020) como é possível ver na ilustração

da Figura 8.

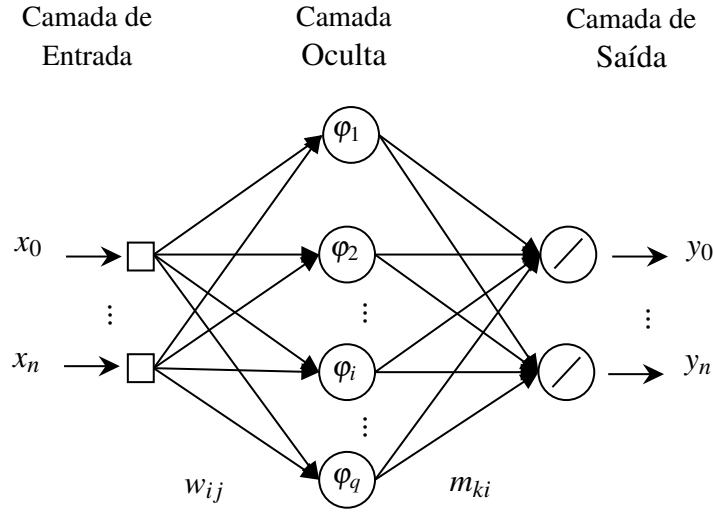


Figura 8: Arquitetura de uma rede neural MLP.

No processo de treinamento do MLP é utilizado o algoritmo de retropropagação do erro (*backpropagation*) que teve como autor principal o Rumelhart (RUMELHART; HINTON; WILLIAMS, 1986). O treinamento se dá em dois sentidos: um direto e outro reverso. O direto que consiste no cálculo das ativações dos neurônios da camada oculta que percorre a rede da camada de entrada a saída. No reverso que percorre a rede da camada de saída a camada oculta.

Dado um vetor de atributos $\mathbf{x}$, na iteração t , e $\mathbf{w}$ os pesos, é calculado o somatório das ativações dos neurônios da camada oculta

$$u_i(t) = \sum_{j=0}^n w_{ij}(t)x_j(t) = \mathbf{w}_i^T \mathbf{x}(t), \quad i = 1, \dots, q \quad (2.4)$$

em que T representa a operação de transposta dos vetores e q indica o número de neurônios da camada intermediária.

Geralmente são utilizadas as funções de ativação ϕ do tipo sigmoide (2.5) e do tipo tangente hiperbólica (2.6) por serem contínuas e diferenciáveis.

$$\phi(u_i(t)) = \frac{1}{1 + \exp[-u_i(t)]} \quad (2.5)$$

$$\phi(t) = \frac{1 - \exp[-u_i(t)]}{1 + \exp[-u_i(t)]} \quad (2.6)$$

O passo seguinte é calcular os neurônios da camada de saída (2.7), no qual o $m \geq 1$

representa o número de neurônios da camada de saída. E após o cálculo da ativação, calcula a saída utilizando uma das formas (2.5) ou (2.6).

$$u_k(t) = \sum_{i=0}^q m_{ij}(t)u_j(t), \quad k = 1, \dots, m \quad (2.7)$$

No sentido reverso, o MLP calcula os gradientes locais $\delta_k(t)$ dos neurônios da camada de saída (2.8), onde $e_k(t)$ é o erro entre a saída desejada e a saída gerada pelo modelo, e $\varphi'[u_k(t)]$ é a derivada da função de ativação.

$$\delta_k(t) = e_k(t)\varphi'[u_k(t)], \quad k = 1, \dots, m \quad (2.8)$$

O mesmo principio é feito para calcular os gradientes locais $\delta_i(t)$ dos neurônios da camada oculta (2.9),

$$\delta_i(t) = \varphi'[u_i(t)] \sum_{k=1}^m m_{ki}(t)\delta_k(t), \quad i = 1, \dots, q \quad (2.9)$$

O passo final é calcular a atualização dos pesos da rede MLP, sendo que (2.10) atualiza os pesos da camada oculta e (2.11) atualiza os pesos da camada de saída. Onde η representa a taxa de aprendizagem.

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i(t)x_j(t) \quad (2.10)$$

$$m_{ki}(t+1) = m_{ki}(t) + \eta \delta_k(t)y_j(t) \quad (2.11)$$

2.2.3 Radial Base Function (RBF)

A rede de base radial (RBF) é caracterizada por uma arquitetura feedforward e consiste basicamente em uma camada de entrada, uma única camada oculta e uma camada de saída (BROOMHEAD; LOWE, 1988). Formalmente, para uma determinada entrada x , a saída de rede y pode ser escrita como (BIZON *et al.*, 2014)

$$y = \sum_{i=1}^N \omega_i R_i(x) + \omega_o, \quad (2.12)$$

onde ω_i são pesos, ω_o é um termo de bias, N representa o número de neurônios na camada oculta, enquanto R_i são as funções de ativação, dadas por

$$R_i(x) = \varphi \left(\left\| x - c_i \right\| \right). \quad (2.13)$$

Na Equação 2.13, φ é a função radial que fornece a característica não linear do modelo e c_i são os chamados centros RBF. O RBF mais popular é dado pela função Gauss

$$\varphi(r) = \exp \left(\frac{-r^2}{\sigma^2} \right), \quad (2.14)$$

com r indicando a distância euclidiana entre o vetor de entrada x e o centro c_i , e σ sendo o chamado parâmetro de dispersão a ser determinado. Neste trabalho, a influência de outros tipos de RBF na precisão da rede é analisada, particularmente multiquadrática inversa e cúbica, dada respectivamente por:

$$\varphi(r) = \exp(r^2 + \sigma^2)^{-1/2} \quad e \quad \varphi(r) = r^3. \quad (2.15)$$

Uma das vantagens da função de ativação da função de base radial, é possuir uma classificação robusta e velocidade de aprendizado rápida. Sua arquitetura mais comum tem somente uma camada intermediária, porém versões com mais de uma camada já foram propostas (HE; LAPEDDES, 1994).

As redes RBF são aproximadores universais de funções assim como as redes MLP, mas que apresentam um tempo de treinamento geralmente mais rápido. A rede RBF constrói aproximações locais para o mapeamento entrada e saída, enquanto o MLP constrói aproximações globais (BRAGA; LUDERMIR; BERNARDA, 2011).

2.2.4 Extreme Learning Machine (ELM)

Esta rede foi desenvolvida para fornecer uma alternativa às redes neurais convencionais em termos de treinamento. Entre os principais detalhes do ELM, o algoritmo permite treinar uma rede neural que possui uma camada de entrada e uma camada oculta, sem o processo iterativo de retropropagação, porque os pesos são inicializados aleatoriamente na camada oculta. Desse modo, os pesos da camada de saída são estabelecidos analiticamente sem o processo de configuração de parâmetros (HUANG; ZHU; SIEW, 2006). O algoritmo oferece vantagens como velocidade de aprendizado rápida, fácil implementação e menos recursos (SHUKLA;

RAGHUWANSI, 2019).

Dado L o número de neurônios da camada oculta, N o número de padrões, β_i pesos da camada de saída e g_i a função de ativação, em (2.16) representa a modelagem matemática do ELM, portanto $j = 1, \dots, N$.

$$t_j = \sum_{i=1}^L \beta_i g_i(x_j) = \sum_{i=1}^L \beta_i g_i(w_i \cdot x_j + b_i) \quad (2.16)$$

Aqui, t_j é o vetor de saída, para um padrão de entrada x_i ; portanto, w_i e b_i vetores de peso definidos aleatoriamente. Em forma de matriz, pode ser representado da seguinte forma

$$H\beta = T. \quad (2.17)$$

Sendo,

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_L \cdot x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_L \cdot x_N + b_L) \end{bmatrix}_{N \times L},$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}$$

Se o número de padrões N é igual ao número de neurônios em cada L , $N = L$ oculto, a matriz de saída da camada oculta H é quadrada e invertível, portanto $\|H\beta - T\| = 0$ para qualquer w_i e b_i selecionado aleatoriamente. No entanto, na maioria dos treinamentos, o número de padrões é maior que o número de neurônios da camada oculta, $N \gg L$, tornando H não quadrado. Assim, a solução de mínimos quadrados com a menor norma é apresentada da seguinte forma

$$\beta = H^\dagger T \quad (2.18)$$

onde $H^\dagger$ pode ser calculado por uma transformação matemática com Moore-Penrose inversa generalizada H da seguinte forma:

$$H^\dagger = (H^T H)^{-1} H \quad \text{ou} \quad H^\dagger = H^T (H H^T)^{-1}. \quad (2.19)$$

O algoritmo de treinamento ELM pode ser apresentado da seguinte forma (LIU *et al.*, 2014):

Entrada: Dado o padrão $S = \{(x_i, t_i) | x_i \in R_n, t_i \in R_m\}$ e o número de neurônios L .

Saída: Pesos da camada de entrada e de saída.

Etapa 1: Selecione aleatoriamente valores para os pesos w_i e bias b_i , onde $i = 1, \dots, N$;

Etapa 2: Calcula a matriz H da camada oculta;

Etapa 3: Calcula pesos da camada de saída $\beta = H^\dagger T$;

A Tabela 2 apresenta as funções de ativação mais usadas no ELM, onde $\bar{\omega}$ representa os pesos, b o bias e $\mathbf{x}$ o padrão (ZHANG *et al.*, 2017).

Tabela 2: Função de ativação comumente usada no ELM.

Função Sigmoid	$g(\bar{\omega}, b, \mathbf{x}) = \frac{1}{1 + \exp[-(\bar{\omega} \cdot \mathbf{x} + b)]}$
Função Gaussiana	$g(\bar{\omega}, b, \mathbf{x}) = \exp(-b \ \mathbf{x} - \bar{\omega}\)$
Função seno	$g(\bar{\omega}, b, \mathbf{x}) = \sin(\bar{\omega} \cdot \mathbf{x} + b)$
Função cosseno	$g(\bar{\omega}, b, \mathbf{x}) = \cos(\bar{\omega} \cdot \mathbf{x} + b)$
Função de limite rígido	$g(\bar{\omega}, b, \mathbf{x}) = \begin{cases} 1 & \text{if } \bar{\omega} \cdot \mathbf{x} + b \leq 0 \\ 0 & \text{caso contrário} \end{cases}$

2.2.5 Naive Bayes classifier (Bayes)

O classificador de Naive Bayes é um classificador probabilístico baseado no teorema de Bayes (MURPHY *et al.*, 2006). O teorema de Bayes descreve a relação entre probabilidades condicionais de uma hipótese e observações. Conforme apresentado na Equação 2.20, sendo a hipótese a classe desconhecida $c \in \{C_1, C_2, \dots, C_K\}$ onde C_K denota o k -ésimo valor de classe. E $\mathbf{x}$ representa a observação feita de tamanho n em que $\mathbf{x} = (x_1, \dots, x_n)$, como mostrado em,

$$P(c|\mathbf{x}) = \frac{P(\mathbf{x}|c)P(c)}{P(\mathbf{x})}, \quad (2.20)$$

ao expandir $\mathbf{x}$ usando a regra da cadeia, temos,

$$P(c|x_1, \dots, x_n) = \frac{P(x_1|c)P(x_2|c) \dots P(x_n|c)P(c)}{P(x_1, \dots, x_n)}, \quad (2.21)$$

que pode ser expresso como,

$$P(c|x_1, \dots, x_n) = \frac{P(c) \prod_{i=1}^n P(x_i|c)}{\sum_{j=1}^K P(x|C_j)P(C_j)}. \quad (2.22)$$

O classificador Bayesiano gera uma função que atribui uma classe $\hat{y}$ à amostra nova de $\mathbf{x}$ com base na regra de classificação apresentada na Equação 2.22 conhecida como máximo a

posteriori *maximum a posteriori* ou MAP.

$$\hat{y} = \operatorname{argmax} P(c) \prod_{j=1}^n P(x_j|c) \quad (2.23)$$

A função densidade de probabilidade mais comumente adotadas é a Gaussiana. A generalização multivariada de uma distribuição Gaussiana no espaço n-dimensional para a classe c é mostrado em,

$$P(\mathbf{x}|c) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \quad (2.24)$$

onde $\boldsymbol{\mu}_k$ é o vetor de médias e Σ_k a matriz de covariância. E sendo $|\Sigma_k|$ e Σ_k^{-1} são respectivamente o determinante e a inversa da matriz de covariância.

2.3 Classificação com Opção de Rejeição

Classificação com opção de rejeição é uma estratégia que consiste em categorizar os padrões que possuem alta possibilidade de serem classificados incorretamente. Esses padrões que são considerados duvidosos podem ser rejeitados e posteriormente avaliados por algum método ou por um especialista humano que pode classificá-los com mais precisão. Dessa forma, aumenta a chance dos padrões não-rejeitados serem classificados corretamente.

Quando o conjunto de classes é constituído por somente duas saídas (-1) e (+1), dizemos que o problema é do tipo binário. Um tarefa de classificação binária é definir uma saída como sendo $y = -1$ ou $y = +1$, ou considerando o estudo de detecção de erros em sequências de números binários, $y = -1$ representando 0 e $y = +1$ representando 1, isso com base em estimativas de probabilidades a posteriori $p(C_i|\mathbf{x})$, para um dado vetor de entrada $\mathbf{x}_i$ do conjunto de padrões.

Em ordem de importância o conteúdo repassado neste assunto de opção de rejeição são derivado dos trabalhos relacionados em (CHOW, 1970), (CHOW, 1957), (ROCHA NETO, 2011), (SOUSA; MORA; CARDOSO, 2009), (FUMERA; ROLI, 2002), (BOUNSIAR; GRALL; BEAUSEROY, 2006) e (OLIVEIRA, 2016).

2.3.1 Regra da Decisão Ótima

Chow possui várias contribuições relacionadas a este tema. Em (CHOW, 1970) descreve que uma regra de decisão é ideal se dado uma taxa de erro (ou probabilidade de erro) minimiza a taxa de rejeição (ou probabilidade de rejeição). Além disto em (CHOW, 1957), a regra de decisão ótima deve rejeitar um determinado padrão, se o maior valor das probabilidades a posteriori for menor do que algum limiar de rejeição.

Mais explicitamente, a regra de decisão ótima δ é dada como

$$(a) \quad \delta(d_k|\mathbf{x}) = 1 \quad (k \neq 0), \quad (2.25)$$

em que d_k representa a decisão de aceitar o padrão $\mathbf{x}$ e atribuí-lo à classe $k = 1, \dots, c$, tal que c representa o número de classes do problema. A Equação (2.25) considera que se aceite o padrão $\mathbf{x}$ para fins de reconhecimento e identifique-o como pertencente à k -ésima classe, se

$$p_k F(\mathbf{x}|C_k) \geq p_j F(\mathbf{x}|C_j) \quad \text{para todo } j = 1, 2, \dots, c, \quad (2.26)$$

e

$$p_k F(\mathbf{x}|C_k) \geq (1-t) \sum_{i=1}^c p_i F(\mathbf{x}|C_i). \quad (2.27)$$

Em,

$$(b) \quad \delta(d_o|\mathbf{x}) = 1 \quad (k = 0), \quad (2.28)$$

d_o representa a decisão de rejeitar o padrão $\mathbf{x}$ sempre que

$$\max_i \{p_i F(\mathbf{x}|C_i)\} < (1-t) \sum_{i=1}^c p_i F(\mathbf{x}|C_i). \quad (2.29)$$

onde $\mathbf{x}$ é o vetor padrão, c é o número de classes, $(P_1, P_2, \dots, P_c)$ é a distribuição de probabilidade a priori das classes, $F(\mathbf{x}|i)$ é a densidade de probabilidade condicional para $\mathbf{x}$, dada a i -ésima classe, $d_i (i \neq 0)$ é a decisão de que $\mathbf{x}$ é identificado como da i -ésima classe enquanto d_0 é a decisão de rejeitar, e t é uma constante entre 0 e 1 ($0 \leq t \leq 1$). A probabilidade de erro, ou taxa de erro, é

$$E(t) = \int_x \sum_{i=1}^c \sum_{j \neq i}^c \delta(d_i|\mathbf{x}) p_i F(\mathbf{x}|C_i) dx \quad (2.30)$$

e a probabilidade de taxa de rejeição ou rejeição é

$$R(t) = \int_x \delta(d_o|\mathbf{x}) \sum_{i=1}^c p_i F(\mathbf{x}|C_i) dx \quad (2.31)$$

ambas as taxas de erro são funções implícitas do parâmetro t . A probabilidade de

reconhecimento correto é

$$C(t) = \int_x \sum_{i=1}^c \delta(d_i|\mathbf{x}) p_i F(\mathbf{x}|C_i) dx = 1 - E(t) - R(t) \quad (2.32)$$

e a probabilidade de aceitação (ou taxa de aceitação) é definido como

$$A(t) = C(t) + E(t). \quad (2.33)$$

Definindo $m(x)$, tal que

$$m(\mathbf{x}) = \frac{p_i F(\mathbf{x}|C_i)}{\sum_{i=1}^c p_i F(\mathbf{x}|C_i)} \quad (2.34)$$

então, a regra ótima de reconhecimento pode ser simplificada e entendida como

1. aceitar o padrão $\mathbf{x}$, se

$$m(\mathbf{x}) \geq (1 - t) \quad \text{ou} \quad (2.35)$$

2. rejeitar o padrão $\mathbf{x}$, se

$$m(\mathbf{x}) < (1 - t). \quad (2.36)$$

A regra ideal é rejeitar o padrão $\mathbf{x}$ sempre que seu máximo de probabilidades a posteriori $m(\mathbf{x})$ for pequeno (menor que $1 - t$). A otimização pode ser analisada observando que $m(\mathbf{x})$ é a probabilidade condicional de reconhecer corretamente um determinado padrão $\mathbf{x}$. Uma prova detalhada é dada em (CHOW, 1957).

O parâmetro t na regra de decisão é chamado de limiar de rejeição. Para qualquer valor fixo de t ($0 \leq t \leq 1$), a regra de decisão δ particiona o espaço padrão V em dois conjuntos (ou regiões) disjuntos $V_A(t)$ e $V_R(t)$ em que (2.27) e (2.29), respectivamente, correspondem a

$$V_A(t) = \{\mathbf{x} | m(\mathbf{x}) \geq (1 - t)\} \quad (2.37)$$

e

$$V_R(t) = \{\mathbf{x} | m(\mathbf{x}) < (1 - t)\} \quad (2.38)$$

Sem perda de generalidade, será assumido que $F(\mathbf{x})$ é diferente de zero em todo o espaço V , caso contrário, o conjunto sobre o qual $F(\mathbf{x})$ é zero é excluído primeiro. V_A e V_R são chamados, respectivamente, de região de aceitação e região de rejeição da regra de decisão. A Figura 9 ilustra as regiões descritas pelos conjuntos disjuntos V_A e V_R , onde a região sombreada é V_R e a região não sombreada é V_A .

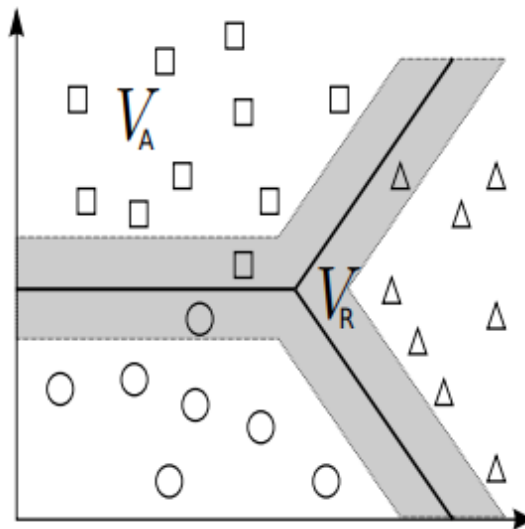


Figura 9: Regiões de rejeição e aceitação da regra ótima de reconhecimento. Fonte: (ROCHA NETO, 2011).

Uma descrição completa do desempenho dos sistemas de reconhecimento é dada pela relação de rejeição e erros, isto é, a relação funcional de erro (E) e rejeição (R) em todos os níveis. Uma curva típica para tal relação é apresentada na Figura 10. Pode-se observar que à medida que a rejeição aumenta a taxa de erro diminui, e vice-versa. Como E e R dos sistemas de reconhecimento ótimos são funções monotônicas do limiar de rejeição t , é possível calcular a taxa de erro $E(t)$ e a taxa de rejeição $R(t)$ para um determinado limiar.

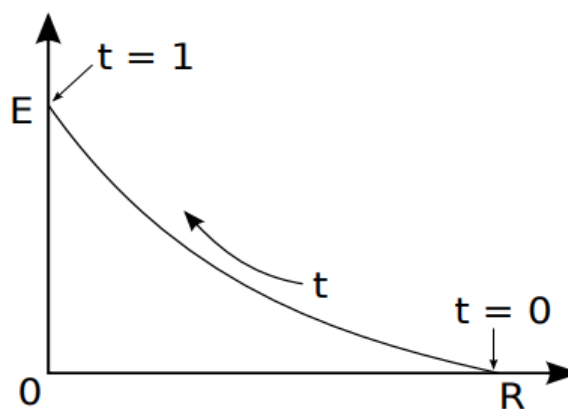


Figura 10: Ilustração da curva de compromisso entre o erro e a rejeição. Fonte: (ROCHA NETO, 2011).

Chow (1970) mostra a taxa de rejeição em função do limiar de rejeição, onde descreve o desempenho do sistema de reconhecimento. Na Figura 11, é mostrado um caso típico dessa curva. As regiões representam a taxa de erro quando nenhuma rejeição é permitida (a soma das duas áreas hachuradas), ou então a região que representa a taxa de erro quando são permitidas rejeições (região hachurada superior).

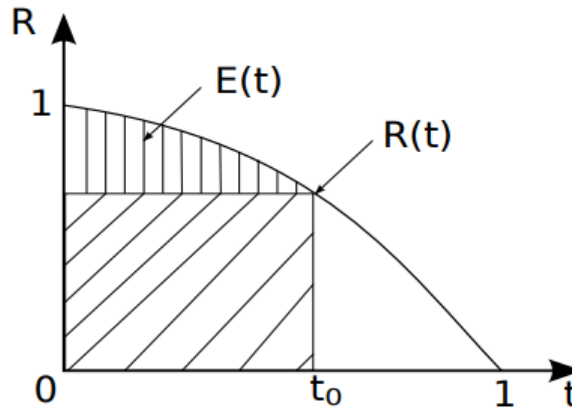


Figura 11: Ilustração da curva hipotética de compromisso entre o erro e a rejeição. Fonte: (ROCHA NETO, 2011).

Sabe-se em (CHOW, 1957) que a regra de decisão ótima dada em (2.25) a (2.28) também é uma regra de risco mínimo se a função de custo for uniforme dentro de cada classe de decisões, ou seja, se nenhuma distinção for feita entre os erros, entre os rejeitados e o reconhecimento correto. O limite de rejeição está relacionado aos custos da seguinte maneira

$$t = \frac{W_r - W_c}{W_e - W_c} \quad (2.39)$$

em que W_r , W_e e W_c são os custos de rejeitar, de errar e de acertar, respectivamente. A regra de decisão ótima apresentada nas Equações (2.25) a (2.29) é também uma regra de risco mínimo se os custos associados aos erros, aos acertos e às rejeições são iguais. Usualmente, tem-se que $W_c < W_r < W_e$. Considerando $W_c = 0$ e $W_e = 1$, pode-se inferir que o custo de rejeição W_r é igual ao limiar de rejeição t ($W_r = t$). Assim, o risco mínimo para um determinado limiar é dado por

$$r(t) = E(t) + tR(t) = \int_0^{t_0} R(t)dt \quad (2.40)$$

o que corresponde à área hachurada na Figura 11. Enquanto o risco empírico, para uma função de decisão e um determinado conjunto de treinamento, é definida como

$$r_{emp}(\alpha) = E(\alpha) + W_r R(\alpha) \quad (2.41)$$

em que α representa o vetor de parâmetros do modelo, dentre eles o limiar de rejeição.

Em um problema de classificação binária, a condição para rejeitar, descrita na Equação 2.29, pode não ser satisfeita quando $t > 0,5$, uma vez que a taxa de rejeição é sempre zero se o limiar de rejeição t exceder $1/2$. Por esta razão o valor de W_r está sempre no intervalo $[0;0,5]$.

2.3.2 Abordagens para Classificação Binária com Opção de Rejeição

Na classificação binária convencional, o espaço das amostras é dividido em duas regiões pelo classificador desenvolvido, como pertencentes a uma das classes $C1$ ou $C2$. Na classificação binária com opção de rejeição, o espaço das amostras são classificados como pertencentes a uma das classes $C1$, $C2$ ou em uma classe de rejeição, onde devem estar os objetos considerados de difícil classificação, através de algum valor de limiar de decisão. Na Figura 12 temos a representação geométrica de uma classificação binária convencional (a) e (b) com opção de rejeição em um espaço $\mathbb{R}^2$.

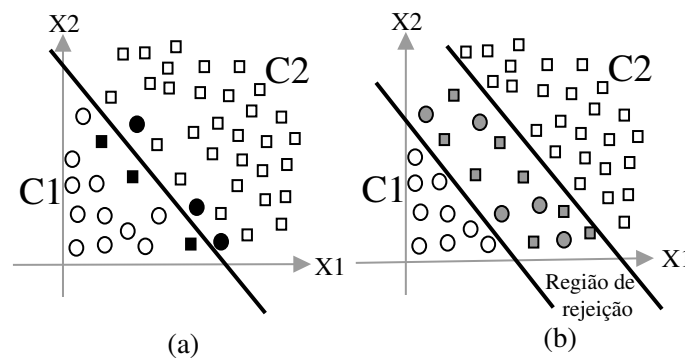


Figura 12: (a) Classificação binária convencional que divide o espaço em duas regiões: $C1$ ou $C2$; (b) classificação binária com opção de rejeição que divide o espaço em três regiões: $C1$, $C2$ ou Rejeição.

Ainda considerando a Figura 12, para padrões muito próximos do valor de decisão, há uma grande probabilidade de ocorrência de erros de classificação, note os padrões em preto, no qual o padrão pode ser atribuído incorretamente a uma classe. Com isso, a área de rejeição reserva um espaço para aumentar a confiabilidade da classificação.

O classificador binário com opção de rejeição pode ser implementado através de três diferentes abordagens, a saber: (i) usando apenas um classificador; (ii) usando dois classificadores independentes; e (iii) usando um classificador com opção de rejeição incorporada na regra de aprendizado.

(i) **Um único classificador binário padrão:** Um padrão é rejeitado se o máximo das duas probabilidades a posteriori, ou seja, $\max\{p(\mathbf{x}|C_{-1}), p(\mathbf{x}|C_{+1})\}$, for menor que um determinado limiar. Se o classificador não fornecer saídas probabilísticas utiliza-se um limiar de rejeição. A região de rejeição é determinada após o treinamento do classificador, definindo valores limite adequados sobre a saída do classificador (ALMEIDA, 2010), (ROCHA NETO, 2011) e (OLIVEIRA, 2016). Esta abordagem leva a uma situação em que não há interseção entre as superfícies de decisão, conforme mostrado na Figura 13.

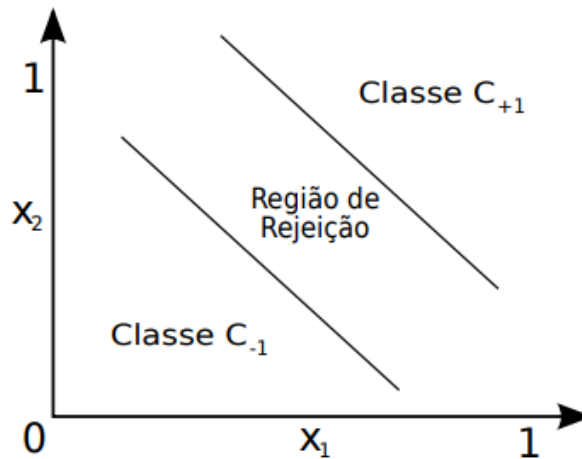


Figura 13: Regiões de decisão obtidas pela utilização de um único classificador para um problema binário. Fonte: (SOUSA; MORA; CARDOSO, 2009).

(ii) **Dois classificadores independentes:** Nessa abordagem são utilizados dois classificadores. O primeiro classificador é treinado para classe C_1 , somente quando a probabilidade de C_{-1} é elevada, o segundo classificador é treinado para saída C_2 somente quando a probabilidade C_{+1} é elevada. A simplicidade desta estratégia é rejeitar sempre que os classificadores discordarem (ALMEIDA, 2010), (ROCHA NETO, 2011).

A regra de aceitação ou rejeição de um padrão para a estratégia baseada em dois classificadores é dada por (BOUNSIAR; GRALL; BEAUSEROY, 2006):

$$\text{Se } P_1(C_{+1}|\mathbf{x}) \geq 0,5 \text{ e } P_2(C_{-1}|\mathbf{x}) < 0,5 \text{ então classe } C_{+1}; \quad (2.42)$$

$$\text{Se } P_1(C_{+1}|\mathbf{x}) < 0,5 \text{ e } P_2(C_{-1}|\mathbf{x}) \geq 0,5 \text{ então classe } C_{-1}; \quad (2.43)$$

$$\text{Se } P_1(C_{+1}|\mathbf{x}) \geq 0,5 \text{ e } P_2(C_{-1}|\mathbf{x}) \geq 0,5 \text{ então rejeitar}; \quad (2.44)$$

$$\text{Se } P_1(C_{+1}|\mathbf{x}) < 0,5 \text{ e } P_2(C_{-1}|\mathbf{x}) < 0,5 \text{ então rejeitar}. \quad (2.45)$$

Em que $P_1(C_{+1}|\mathbf{x})$ e $P_2(C_{-1}|\mathbf{x})$ são as probabilidades a posteriori dos classificadores C_1 e C_2 , respectivamente. Esta abordagem pode fazer com que ocorra uma interseção entre as superfícies de decisão das classes, levando à ocorrência de regiões em que a confiança na tomada de decisão é bastante fraca (SOUSA; MORA; CARDOSO, 2009). Na Figura 14 apresenta esta situação ilustrada.

(iii) **Um único classificador com opção de rejeição incorporado:** Esta abordagem tem a vantagem de determinar a região de rejeição durante a fase de treinamento, mas necessita que o algoritmo de aprendizagem tenha suporte a essa abordagem. Consiste em projetar classificadores que trazem embutido no processo de otimização de suas funções-custo a

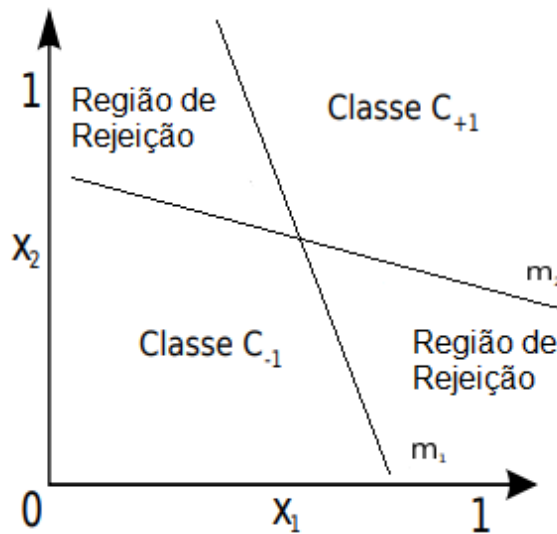


Figura 14: Regiões de decisão obtidas pela utilização de dois classificadores independentes para um problema binário. Fonte: (SOUSA; MORA; CARDOSO, 2009) (modificado).

capacidade de classificação com opção de rejeição. Na abordagem com opção de rejeição embutida, o modelo aprende intrinsecamente a decidir entre as três saídas possíveis, C_{-1} , rejeição e C_{+1} . O resumo dessa abordagem é apresentado nos trabalhos de (SOUSA; MORA; CARDOSO, 2009), (FUMERA; ROLI, 2002) e (BOUNSIAR; GRALL; BEAUSEROY, 2006).

O presente trabalho adota a primeira abordagem para a implementação de um classificador com opção de rejeição para detectar erros em transmissão de bits de dados.

2.3.3 Abordagem com um Único Classificador Padrão

Exemplificando o que foi apresentado no tópico anterior, nesta abordagem um padrão não é rejeitado se o máximo das duas probabilidades posteriori for superior que um determinado limiar $(1 - t)$, ou seja, $\max\{p(\mathbf{x}|C_{-1}), p(\mathbf{x}|C_{+1})\} > (1 - t)$. O limiar de rejeição, que gera consequentemente a região de rejeição, pode ser definido usando uma medida de rejeição calculada após a fase treinamento do classificador, independente do classificador fornecer ou não saídas probabilísticas. Na Figura 15 apresenta o tamanho da região de rejeição, que é proporcional ao valor do limiar, para o limiar igual a zero não existe região de rejeição, ou seja, quanto maior for o limiar então maior é a região de rejeição.

Um exemplo indutivo dessa abordagem é apresentado no trabalho de (OLIVEIRA, 2016). Onde apresenta ilustradamente o processo que o classificador faz para encontrar o limiar. Após terminar todas as n execuções do classificador vão existir n taxas de rejeição e n taxas de acerto para cada valor de limiar. A média de acerto e rejeição de cada limiar correspondente são

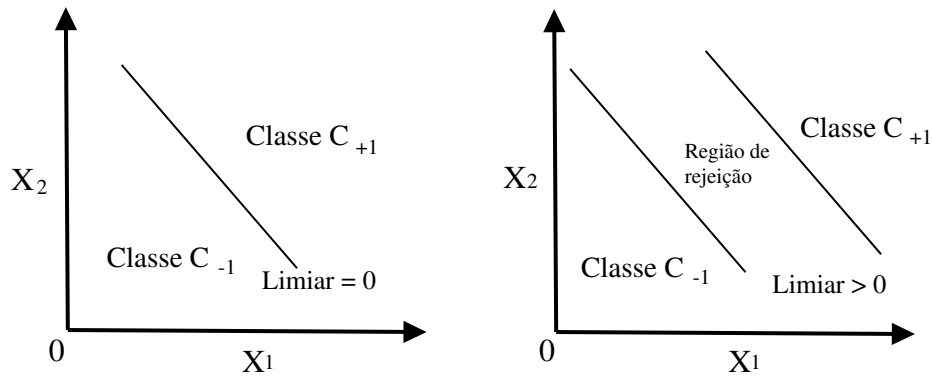


Figura 15: Variação do tamanho da região de rejeição para o limiar igual a zero (Classificação convencional) e limiar diferente de zero (Opção de rejeição).

calculadas. A Figura 16 exemplifica esse processo de classificação com opção de rejeição para valores de limiar iguais a 0; 0,25 e 0,5.

A próxima etapa é escolher o limiar ótimo com base nos resultados obtidos, de tal forma, que permita escolher o melhor classificador com base no desempenho desejável ou mesmo em uma taxa de rejeição aceitável, sendo esta escolha dependente do tipo de problema a ser abordado.

Para isso é preciso analisar o desempenho de predição para diferentes custos de rejeição W_r , buscando minimizar o risco empírico, onde T_E representa a taxa de erro e T_R sendo a taxa de rejeição (CHOW, 1957).

$$\min\{T_E(\text{limiar}) + W_r T_R(\text{limiar})\}. \quad (2.46)$$

O valor de W_r é inversamente proporcional à taxa de rejeição, ou seja, aumenta-se o valor de W_r quando se deseja uma menor quantidade de padrões rejeitados. Conforme a Figura 17 exemplifica para W_r igual a 0.24, o melhor resultado é quando a taxa de acerto é igual a 89,76% e a taxa de rejeição é igual a 12,45% com base nos resultados da Figura 16.

2.4 Conclusão

A seção 2.1 apresentou algoritmos de detecção de erro. Entre os quais, a saber, SPC(Single Parity Check), CRC (Cyclic Redundancy Check) e CHECKSUM. Foi apresentado que os algoritmos são eficientes em detectar erros em transmissões de bits, principalmente o CRC e CHECKSUM. Porém o SPC possui uma alta possibilidade de errar na verificação se o número de erros for diferente da sua configuração par ou ímpar.

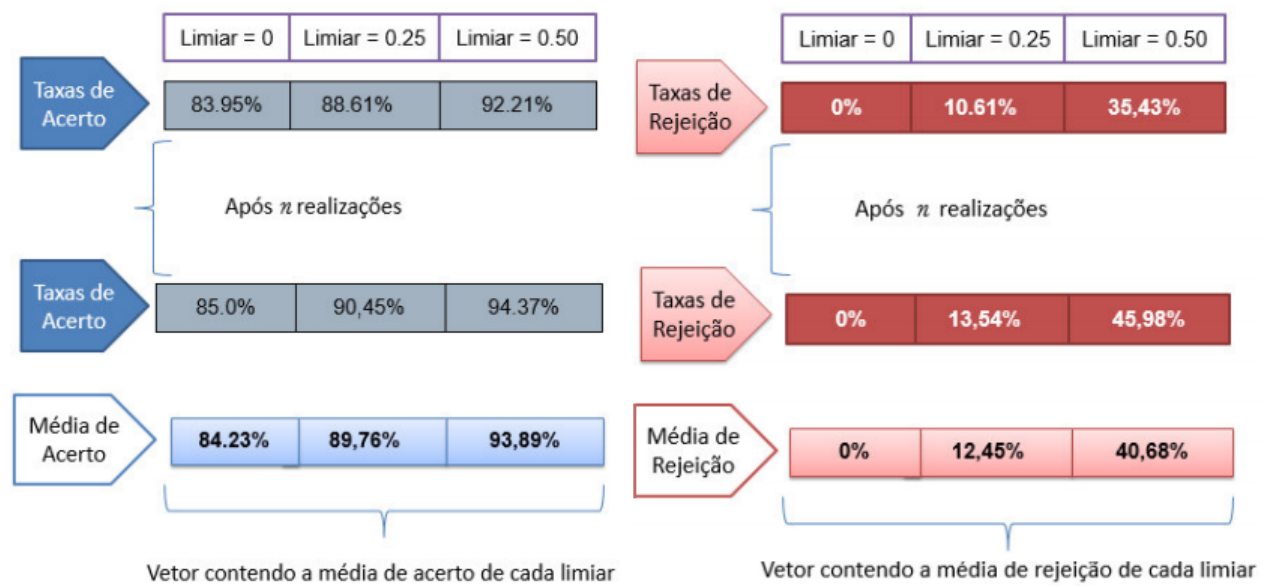


Figura 16: Taxa de acerto e taxa de rejeição para um certo limiar e n realizações. Fonte: (OLIVEIRA, 2016) (modificado).

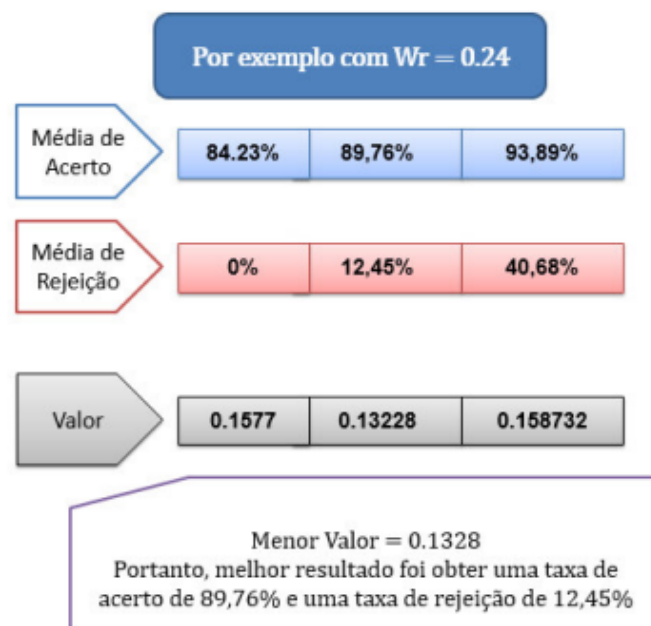


Figura 17: Exemplo de escolha de melhor desempenho quando $W_r = 0.24$. Fonte: (OLIVEIRA, 2016).

Na seção 2.2 foi apresentado a área de pesquisa de classificação de padrões, onde são utilizados três classificadores baseados em redes neurais e um baseado em processos estatísticos. O funcionamento destes algoritmos é de grande importância para esta dissertação. Algoritmos tradicionais de classificação de padrões abordado nesta dissertação, a saber, Multilayer Perceptron (MLP), Extreme Learning Machine (ELM), Radial Base Function (RBF) e Naive Bayes.

A seção 2.3 mostrou a estratégia de classificação com opção de rejeição usando a abordagem com um único classificador binário padrão. A técnica de opção de rejeição é bastante atraente no sentido de aumentar as medidas de acurácia para um determinado custo de rejeição. Também é descrito o conceito de regra de decisão ótima em um processo de classificação com rejeição. O modelo de classificação com opção de rejeição protege o sistema contra eventuais decisões erradas em ocasiões em que existe alta confiabilidade, com os quais podem ser configurados em conformidade com o conhecimento do especialista humano.

MÉTODO PROPOSTO

Neste capítulo é explicado a metodologia para detecção de erros com base em algoritmos de classificação de padrões que são modelados como um equalizador de sinais binário. É aplicado a opção de rejeição para automatizar o processo de detecção de erro. O capítulo inicia com a explicação das etapas empregadas na abordagem da proposta, as quais consistem de forma geral na apresentação de diagramas, algoritmos e técnicas.

O capítulo é essencialmente dedicado a apresentar métodos que são implementados com a opção de rejeição. Apresentamos dois métodos novos para a detecção de erros em transmissões de bits de dados.

O presente capítulo está estruturado em seis seções. A seção 3.1 define teoricamente o conceito de INED e CNED. Na seção 3.2 é apresentado o processo de desenvolvimento do modelo de detecção de erro. Na seção 3.3 é apresentado o diagrama geral do modelo. Nas seções 3.4 e 3.5 são apresentados respectivamente, exemplos práticos do INED e CNED. As principais conclusões deste capítulo são apresentadas na seção 3.6.

3.1 Definição do INED e CNED

O método proposto, Detecção de Erro Neural Independente ou (*Independent Neural Error Detection*, em inglês) (**INED**) caracteriza-se por utilizar a classificação com opção de rejeição como técnica de detecção de erros independente de algoritmos de detecção de erro. Isso significa que o processo é automatizado para fazer o treinamento, predição e dar confiabilidade na entrega de informações. É nossa principal contribuição.

O método proposto, Detecção de Erro Neural Combinado ou (*Combined Neural Error Detection*, em inglês) (**CNED**) é uma variante do método INED, que realiza todos os procedimentos citados anteriormente, porém também se junta a algoritmos de detecção de erro. A ideia de desenvolver um método variante ao INED foi de verificar se conseguimos melhorar a desenvoltura dos atuais detectores de erro.

3.2 Desenvolvimento do Modelo de Detecção de Erro usado no INED e CNED

Em teoria, qualquer algoritmo de classificação de padrão pode ser aplicado para treinamento e inferência com a opção de rejeição. Porém, implementamos com Extreme Learning Machine (ELM), pois o ELM com opção de rejeição provou ser o mais eficiente em precisão e velocidade.

Os padrões gerados são submetidos a treinamento e teste para encontrar todos os hiperparâmetros, semelhante a qualquer processo de classificação. A diferença em nosso modelo está no treinamento e na predição. No conjunto de treinamento, o modelo procura os melhores limites de decisão para cada função de custo. Na etapa de predição, a classificação com opção de rejeição retorna uma das três saídas possíveis: 0; 1 ou R (rejeição). O algoritmo de verificação de erros informa se o pacote ou bloco de informações está correto. Este processo de saída torna nosso método de detecção de erros fácil de implementar e avaliar.

3.2.1 Etapa de Treinamento do Limiar de Decisão

A função de procura do limiar de decisão é uma estratégia para encontrar o melhor limite para uma determinada função de custo α . A função recebe os dados de entrada X_i do conjunto de dados, onde $i = \{1, 2, 3, \dots, N\}$, e N é o número de padrões, w , b_i e β , são os pesos fornecidos pela função de treinamento, também α que é o custo a ser aplicado ao método, começando em $\alpha = 0,04$ e chegando até 0,48, com passo 0.04 como em:

$$\alpha = [0.04, \dots, 0.48]. \quad (3.1)$$

A função define como parâmetros iniciais dos valores de limiares, começando em $L = 0,00$ e chegando a 0,50, com passo 0.0025 como em:

$$L = [0.00, \dots, 0.50]; \quad (3.2)$$

Aqui, para cada limiar, estão usando a função de predição para encontrar as classes dos padrões de teste e verificar o número de classes classificadas e rejeitadas. As funções calculam as taxas de erro e rejeição, que são aplicadas na regra de decisão ótima. A maneira mais adequada de encontrar a taxa de rejeição é usando as seguintes fórmulas:

$$R = \frac{\text{Número de padrões rejeitados}}{\text{Número total de padrões}} \quad (3.3)$$

e a taxa de erro como em:

$$E = \frac{\text{Número de padrões mal classificados}}{\text{Número total de padrões} - \text{Número de padrões rejeitados}} \quad (3.4)$$

esse processo é repetido para diferentes valores de α e L e escolhe aquele que minimiza o risco empírico:

$$r_{emp}(\alpha, L) = E(L) + \alpha R(L) \quad (3.5)$$

O Algoritmo. 1 mostra as etapas para treinar o modelo e encontrar o limite de decisão.

Algoritmo 1 Treinamento do limiar de decisão

- 1: **Entrada** Conjunto de treinamento.
 - 2: **Saída** Limiar de decisão L , hiperparâmetros.
 - 3: :{Etapa 1: Calcula todos os pesos w_i , b_i e β }
 - 4: Calcula β de acordo com a Equação (2.16), Equação(2.17) e Equação(2.18).
 - 5: :{Etapa 2: Procura a melhor configuração de custo e limite.}
 - 6: Inserir α (custo) e L (limite) de acordo com a Equação (3.1) e Equação (3.2).
 - 7: :{Etapa 3: Calcula a minimização do risco empírico}
 - 8: **Repetir para** $R(\alpha, L)$ de acordo com a Equação (3.5).
 - 9: **Retorna** Limiar de decisão L , hiperparâmetros.
-

3.2.2 Etapa de Predição

O objetivo da predição é retornar valores de classe para um determinado padrão X_i . A função recebe as entradas de todos os hiperparâmetros que são os pesos fornecidos pela função de treinamento, L o valor do limiar e X_i o conjunto de dados, sendo que $i = \{1, 2, 3, \dots, N\}$ e N o número de padrões. Considerando as funções sigmóides, a entrada e a saída da rede são calculadas como em:

$$h_i = \frac{1}{1 + e^{-(x_i w^T)}} \quad (3.6)$$

$$y_i = \frac{1}{1 + e^{-(h_i \beta^T)}} \quad (3.7)$$

Considerando a classificação binária que possui duas saídas, o próximo passo é encontrar o maior valor de $y_i = [y_1, y_2]$, consequentemente a classe de saída idx , se y_1 é maior que y_2 , então a classe é 0, caso contrário, 1 como em:

$$idx = \begin{cases} 0 & \text{se } y_1 > y_2 \\ 1 & \text{caso contrário.} \end{cases} \quad (3.8)$$

Com o valor da classe e o limiar de decisão, o mecanismo de aceitação ou rejeição do padrão é calculado da seguinte forma:

$$C = \begin{cases} idx & \text{se } \max(y_i) \geq 1 - L \\ R & \text{caso contrário} \end{cases} \quad (3.9)$$

Neste caso, C pode assumir os valores de $C = 0$ se o padrão tem a estimativa mais alta para ser da classe C_{-1} , $C = 1$ se o padrão tem a estimativa mais alta para ser da classe C_{+1} e $C = R$ se o padrão não tiver estimativa suficiente para ser classificado como classe C_{-1} nem com classe C_{+1} .

A função de predição em Algoritmo. 2 exibe a saída do conjunto de padrões. A função recebe os dados do conjunto de dados, limite e pesos. Calcula as saídas de padrão usando fórmulas (3.6) e (3.7). Em (3.8) e (3.9) mostra os conceitos da opção de rejeição e retorna os dados com os valores "0", "1" ou "R". Onde "R" representa que o sistema rejeitou o bloco.

Algoritmo 2 Algoritmo de predição

- 1: **Entrada** Conjunto de teste, limiar de decisão L , hiperparâmetros.
 - 2: **Saída** Classe binária C
 - 3: :{Etapa 1: Calcula a função de ativação}
 - 4: Calcula H e Y de acordo com Equação (3.6) e Equação (3.7).
 - 5: :{Etapa 2: Procura o valor máximo do índice}
 - 6: Computa o idx de acordo com a Equação (3.8).
 - 7: :{Etapa 3: Decide se o máximo Y é maior que $1 - L$ }
 - 8: Prediz C de acordo com a Equação (3.9).
 - 9: **retorna** C
-

3.2.3 Etapa de Detecção do Erro

A etapa de verificação de erros é normal para todos os detectores de erros. O nosso é de fácil implementação, o seu funcionamento simples implica em uma verificação rápida e eficiente porque apenas soma o número de bits rejeitados por cada pacote ou bloco. No Algoritmo. 3, o processo de detecção de erros é apresentado.

Algoritmo 3 Algoritmo de detecção de erro

- 1: **Entrada** Bloco/pacote de bits C.
- 2: **Saída** Status do bloco de bits S.
- 3: :{Etapa 1: Verifica o bloco/pacote}
- 4: Calcula o número de erros $cont++$.
- 5: :{Etapa 2: Apresenta o status}
- 6: **Se** $cont > 0$ **então** : S é rejeitado;
- 7: **Senão** : S é aprovado.
- 8: **Retorna** S

3.3 Diagrama do Modelo Geral

Todas as etapas do método de detecção proposto são apresentadas no diagrama esquemático da Figura. 18. Os dados passam por uma etapa de treinamento e cálculo dos custos do limiar de decisão. Na etapa de inferência dos dados, o preditor retorna com a classificação e o algoritmo de detecção de erro informa se o bloco de dados está correto ou não.

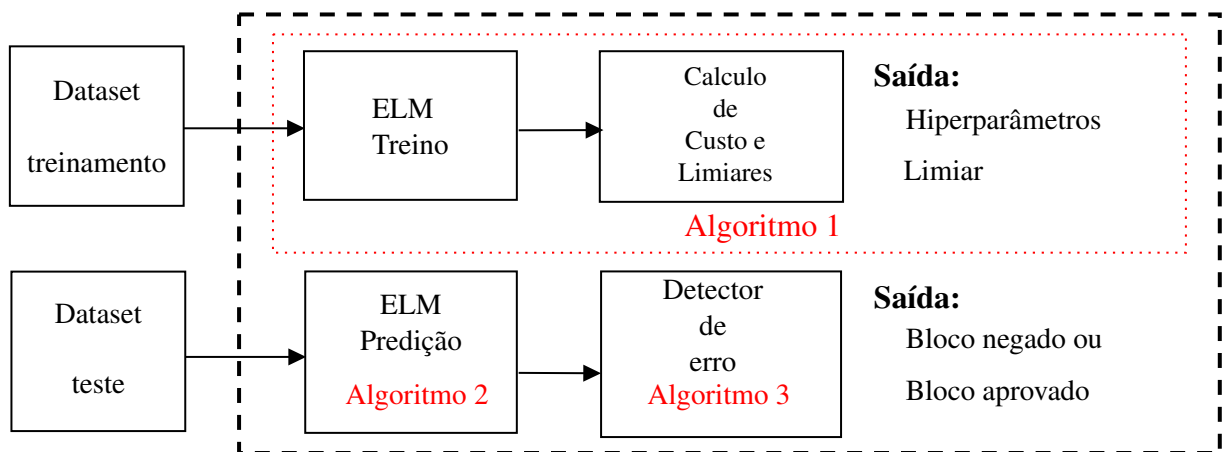


Figura 18: Diagrama que representa o processo de treinamento, predição e detecção de erros com o método INED.

3.4 Exemplo da Detecção de Erro Neural Independente (INED)

O método INED é apresentado no diagrama esquemático da Figura 19. O sistema que verifica o erro com base na rejeição R é o mesmo mostrado no Algoritmo. 3. No esquema apresentado no **Primeiro exemplo**, o classificador se pergunta qual o valor do quarto bit do bloco de código binário, **0 ou 1**?, como o sistema não tem certeza, atribui o valor R pois o custo de rejeitar é menor que o custo de aceitar o bloco, quando o verificador INED detecta a rejeição,

pode ser solicitada uma retransmissão.

No **Segundo exemplo**, nenhum bit é rejeitado pelo sistema, logo a mensagem é aprovada para o destinatário final, pois pode retornar os bits de maneira confiável. Vale informar que o sistema não aceita uma quantidade infinita de retransmissões, porque isso pode causar um atraso na transmissão de todos os blocos. Portanto, se ocorrerem muitas solicitações de retransmissão, o sistema pode optar por transferir a mensagem para avaliação por um especialista humano. O INED não precisa verificar bits redundantes para detectar o erro.

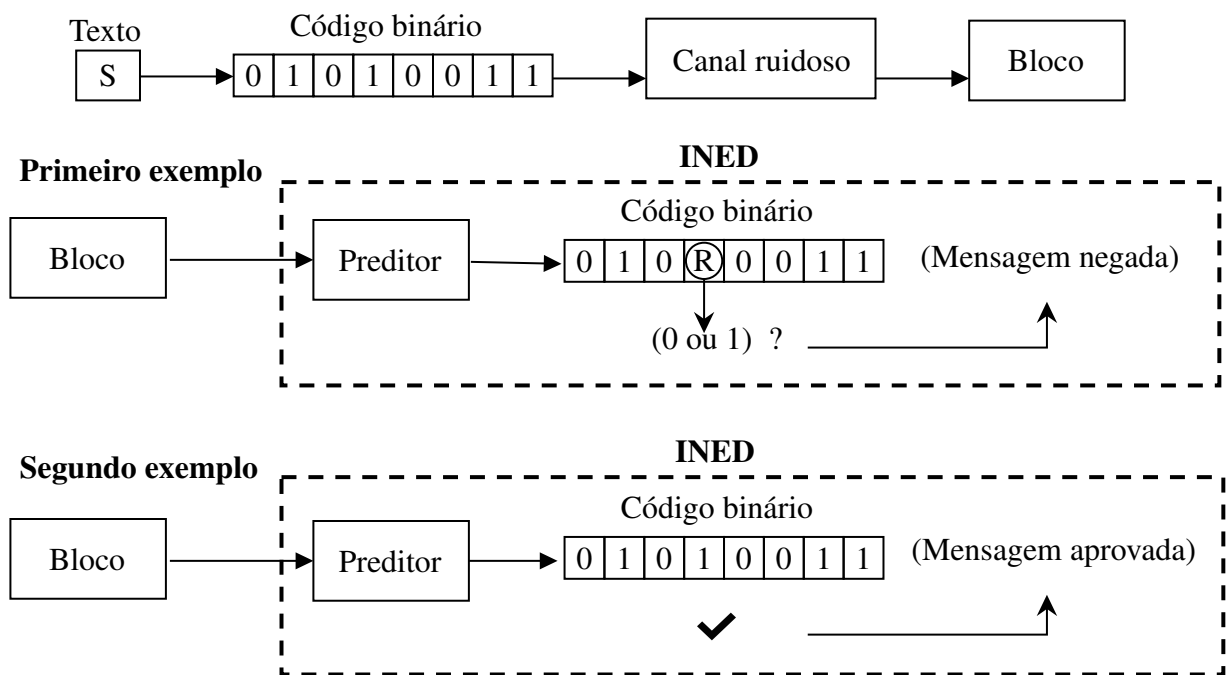


Figura 19: Diagrama representando a detecção de erros pelo método INED na transmissão de mensagens de dados.

3.5 Exemplo do Detector de Erro Neural Combinado (CNED)

O método CNED tem a função de encontrar possíveis erros nas sequências de blocos e se não for o caso, ou seja, não encontrar erros, transferir os dados para uma segunda avaliação usando as abordagens existentes. Enquanto o INED possui um verificador que depende apenas de si mesmo, o CNED é uma combinação com outro método existente. O CNED é apresentado na Figura. 20, onde no **Primeiro exemplo** o algoritmo verifica se há bits duvidosos, no caso afirmativo, a mensagem é negada. No **Segundo exemplo** o algoritmo verifica se ocorreu erro, em caso negativo, a mensagem é enviada para que um algoritmo de detecção convencional que

possa também verificar o bloco de bits. Ou seja, o CNED avalia a ocorrência de erros, porém também envia o bloco para avaliação de outro método de forma a ter uma segunda camada de verificação. O CNED utiliza o Algoritmo. 4 que possui uma pequena diferença em relação ao Algoritmo. 3.

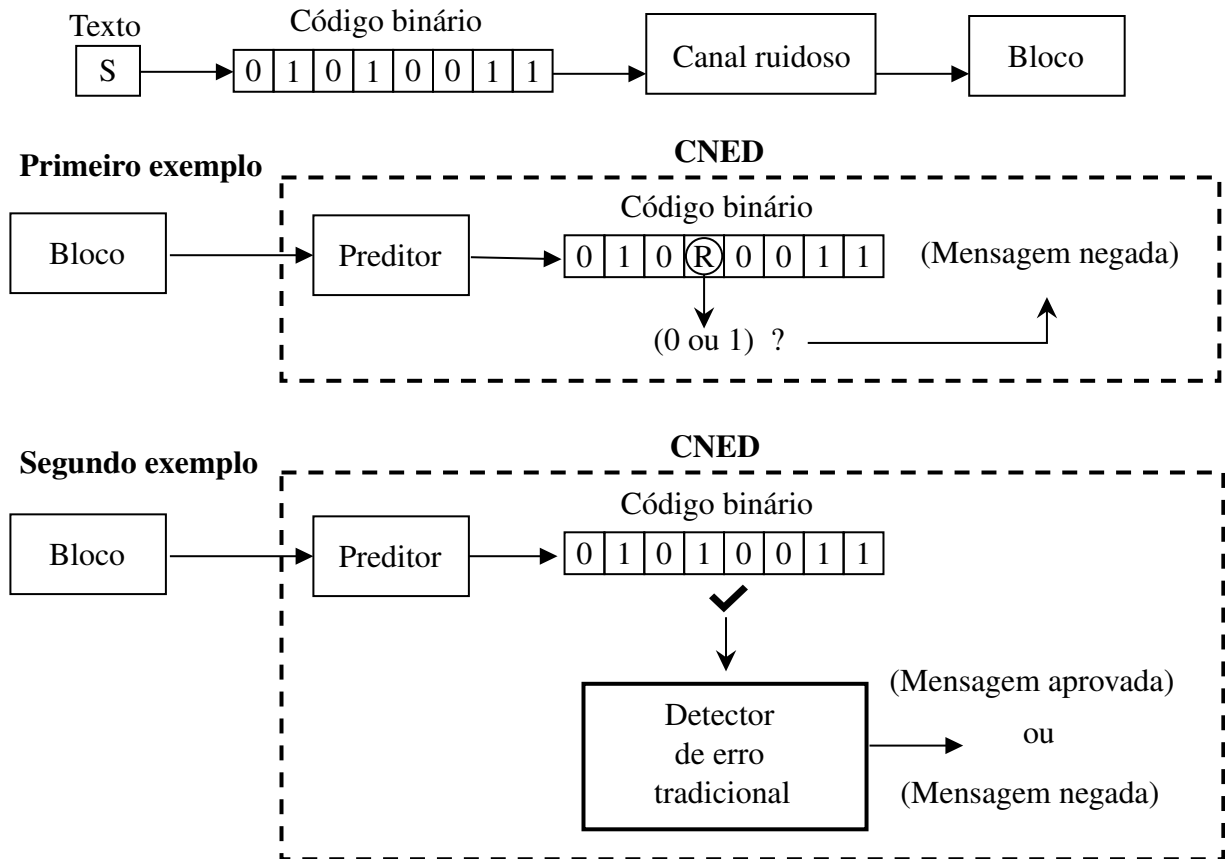


Figura 20: Diagrama representando a detecção de erros pelo método CNED na transmissão de mensagens de dados.

Algoritmo 4 Modelo de Detector de Erro Neural Combinado

- 1: **Entrada** Bloco/pacote de bits C.
 - 2: **Saída** Status do bloco de bits S.
 - 3: :{Etapa 1: Verifica a sequência de erros}
 - 4: Calcula o número de erros $cont++$.
 - 5: :{Etapa 2: Apresenta status}
 - 6: **Se** $cont > 0$ **então** : S é "rejeitado";
 - 7: **Senão** : S é "aprovado" ou "negado" por outro detector;
 - 8: **retorna** S
-

3.6 Conclusão

O capítulo 3 introduz que a detecção de erro pode ser automatizada através da classificação com opção de rejeição. Com o uso matemático e computacional apresentamos através de diagramas, algoritmos e conceitos o funcionamento do método proposto.

A seção 3.1 apresentou a definição formal do INED e CNED e os seus respectivos propósitos. INED é uma maneira que classifica padrões com confiabilidade. CNED é uma maneira que além de classificar padrões com confiabilidade se une com outros algoritmos de detecção para melhorar as irregularidades dos mesmos.

A seção 3.2 apresenta todo o processo para desenvolvimento e formulação da ideia, passando por treinamento, predição e verificação do erro. Na seção 3.3 é apresentado o diagrama do modelo geral, que abrange todos os conceitos mencionados nas seções anteriores.

Nas seções 3.4 e 3.5 são mostrados os respectivos exemplos dos métodos INED e CNED. O método INED aprova ou reprova um bloco de informação de forma independente, enquanto que o CNED após aprovação do bloco, delega a responsabilidade para outro algoritmo de detecção de forma a buscar melhoramento deste.

EXPERIMENTOS E RESULTADOS

Neste capítulo são mostrados os experimentos e resultados alcançados durante o período do mestrado.

4.1 Experimentos

Nesta seção de experimentos são mostrados os processos para gerar a base de dados, bem como foram projetados a simulação computacional e configuração dos parâmetros da classificação, além da simulação computacional em hardware.

4.1.1 Base de Dados dos Experimentos

O desempenho foi avaliado por três conjuntos de dados gerados a partir de palavras aleatórias do mundo real e convertidos em números binários, onde os valores podem assumir os valores 0 ou 1. Na tabela 3, o dataset (Data1) é usado para treinar e testar o modelo desenvolvido. Enquanto (Data2) é usado para avaliação com detectores SPC e CRC. (Data3) usado para avaliar com o detector CHECKSUM.

Tabela 3: Descrição dos datasets.

	Bytes	Bits	Padrões	Blocos
Data1	1250	10000	10000	N/A
Data2	100	800	800	100
Data3	200	1600	1600	100

Uma sequência binária é formada por valores de bits 0 ou 1. Cada valor binário pode ser representado por um vetor $\mathbf{X} = \{x^1, x^2, x^3, \dots, x^i\}$ onde cada valor x^m representa uma amostra de tempo fixo do sinal obtido do aparelho receptor. i representa o último termo do canal, portanto, os padrões podem ser obtidos com base no canal, como em:

$$y(n) = s(n) - 0.5s(n-1) + s(n-2) + 0.5s(n-3) - 0.125s(n-4), \quad (4.1)$$

Aqui, ele produz um valor estimado de saída no sistema receptor, considerando a influência dos valores anteriormente transmitidos no canal, o que caracteriza a interferência intersimbólica. Essa padronização foi adotada para se adequar à saída do classificador. Portanto, dada como exemplo a sequência binária $S = \{10100101\}$ e o exemplo de um canal (4.1), temos os padrões representados na Figura 21.

	X	Y	D
Dataset	[1.0000 0.0000 0.0000 0.0000 0.0000] =	1,0000	1
	[0.0000 - 0.5000 0.0000 0.0000 0.0000] =	- 0,5000	0
	[1.0000 0.0000 1.0000 0.0000 0.0000] =	2,0000	1
	[0.0000 - 0.5000 0.0000 0.5000 0.0000] =	0,0000	0
	[0.0000 0.0000 1.0000 0.0000 - 0.1250] =	0,8750	0
	[1.0000 0.0000 0.0000 0.5000 0.0000] =	1,5000	1
	[0.0000 - 0.5000 0.0000 0.0000 - 0.1250] =	- 0,6250	0
	[1.0000 0.0000 1.0000 0.0000 0.0000] =	2,0000	1

Figura 21: Representação de convolução entre canal e sinal [Y] e formação do conjunto de padrões [X D].

Ainda considerando a Figura 21, X representa o padrão que entra no classificador para ser treinado e avaliado, Y representa o somatório dos atributos do padrão e D representa a variável alvo que é utilizado no processo de aprendizado. Esses são os dados gerados para treinar o modelo.

Os valores binários são submetidos ao mesmo canal, mas com diferentes ambientes ruidosos a partir da relação sinal-ruído (SNR). SNR é uma medida usada nos sistemas de comunicação para comparar a potência do sinal e a potência do ruído como em (4.2), no entanto, os valores do sinal geralmente são expressos usando a escala de decibéis logarítmicos como em (4.3):

$$SNR = \frac{P_{sinal}}{P_{ruído}}. \quad (4.2)$$

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{sinal}}{P_{ruído}} \right). \quad (4.3)$$

Assim, a amplitude do ruído normalizada para gerar um determinado SNR é calculada como em (4.4):

$$A = \sqrt{10^{-SNR_{dB}/10}} \quad (4.4)$$

Figura 22 apresenta a influência do SNR na recepção de mensagens. Observa-se que quanto menor o SNR, maior o grau de influência do ruído no sinal original e, conseqüentemente, quanto maior o SNR, menor a interferência. Vale informar que a Figura 22 é somente uma representação da influência da relação sinal-ruído e o que entra na rede neural é o esquema apresentado na Figura 21.

Bloco original com 8 bits							
0.0000	1.0000	1.0000	0.0000	1.0000	0.0000	0.0000	1.0000
Bloco original com SNR = 50 dB							
-0.0040	1.0047	1.0045	0.0045	1.0021	-0.0038	0.0023	1.0052
Bloco original com SNR = 10 dB							
0.1546	1.3272	1.2299	-0.0960	1.0929	-0.2490	0.2809	0.4973
Bloco original com SNR = 5 dB							
-0.6011	0.5448	-0.6557	0.8089	1.1829	-0.4245	0.7706	0.5375
Bloco original com SNR = 0 dB							
1.0933	2.1093	0.1363	0.0774	-0.2141	1.1135	-0.0068	2.5326

Figura 22: Representação da influência da relação sinal-ruído na mensagem do bloco que é transmitido.

O conjunto de dados são submetidos a uma transmissão que passa por um dos seis ambientes ruidosos com base na relação sinal-ruído, sendo 0, 3, 6, 9, 12, 15 e 18 em dB os SNR escolhidos. Estes seis ambientes vão formar os conjuntos de dados que são testados em experimentos, essa etapa ocorre depois que o sinal passa pelo canal de comunicação.

4.1.2 Testes Realizados

Os métodos INED e CNED são comparados usando o algoritmo de rede neural ELM. Da mesma forma, os detectores SPC, CHECKSUM e CRC foram modelados usando a rede neural ELM para a comparação ser a mais fiel possível. O objetivo das comparações é mostrar que os métodos INED e CNED são melhores ou comparáveis aos outros métodos porque detectam

erros em bits vulneráveis no bloco de dados recebido. Nesta simulação, o programa realiza 50 realizações para calcular a taxa de acerto e a taxa de rejeição.

Os métodos INED e CNED são comparados com o SPC, CRC e CHECKSUM, para verificar com diferentes características a eficiência do método que utiliza a opção de rejeitar, a seguir segue a organização utilizada nas simulações:

- O SPC se baseia na adição de um bit de redundância a cada bloco composto por n bits, para obter uma soma da quantidade par ou ímpar. O número $n = 8$ bits foi escolhido como parâmetro para cada bloco. Assim, na transmissão os blocos de 9 bits são projetados usando os detectores SPC e CNED, com INED somente 8 bits são necessários;
- O CHECKSUM consiste em somar uma certa quantidade de palavras que podem ser transmitidas, as palavras de código são enviadas junto com o resultado invertido dessa soma. Considerando duas palavras com 8 bits para cada palavra, os blocos de 24 bits são transmitidos usando os detectores CHECKSUM e CNED, nos blocos INED são 16 bits;
- O CRC é baseado no uso de lógica polinomial, incluindo um pequeno número de bits redundantes. Ele pode detectar erros múltiplos e simples nas mensagens de dados. Nesta análise, utilizamos o padrão CRC-8-CCITT para uma sequência de 8 bits. São formados blocos com 16 bits transmitidos com CNED e CRC, e 8 bits com o detector INED.

O algoritmo ELM com opção de rejeição é comparado com os classificadores MLP, RBF e Naive Bayes, todos com a opção de rejeição. Essa simulação serve como modelo para mostrar a eficiência dos algoritmos de classificação. Os algoritmos são comparados usando também a SNR com os valores 0, 3, 6, 9, 12, 15 e 18 em dB. O programa realiza 50 realizações para calcular a acurácia e os valores de desvio padrão.

4.1.3 Validação Cruzada e Parâmetros da Classificação

Os padrões são classificados usando o conceito de 80% dos padrões para treinamento e 20% dos padrões para teste. O programa desenvolvido usa a validação cruzada de $k - Folds$ para encontrar os melhores parâmetros dos classificadores. O método separa o conjunto de treinamento em k partições e para cada configuração de parâmetro, um classificador é treinado com uma das combinações possíveis. O erro médio de todas as partes de k são calculados para determinar a melhor configuração de parâmetro. Neste trabalho, para não aumentar o tempo de treinamento do classificador são utilizados 5 *Folds*. Com o objetivo de minimizar a equação empírica da função de risco, são empregados os seguintes valores de custos de rejeição:

$$w_r = \{0.04; 0.08; 0.12; 0.16; 0.20; 0.24; 0.28; 0.32; 0.36; 0.40; 0.44; 0.48\} \quad (4.5)$$

O ambiente de execução de teste foi realizado em um computador com um processador i5, uma frequência de 3,4 GHz e 8GB de memória DDR3. Ubuntu 16.04.6 LTS de 64 bits. Todos os algoritmos foram implementados na linguagem Matlab (versão 2013a - licença 28878-48087-65497-12872-42979-61998-52336-35817-09598).

4.1.4 Simulação Computacional em Hardware

O principal objetivo da realização de testes práticos com equipamentos de transmissão de dados é ver o funcionamento dos métodos de detecção de erro em situações reais. Além disso, analisar a eficiência dos métodos propostos INED e CNED. Para essa análise, foi elaborado um sistema de comunicação sem fio, entre duas antenas, uma para enviar os bits e outra para receber a informação. Como apresentado na ilustração da Figura 23, o processo envolveu a modulação ASK que consiste na modificação do nível de amplitude da onda portadora em função do sinal digital de entrada a ser transmitido. O sinal modulante assume um dos dois níveis discretos da fonte de informação (nível lógico 0 ou 1). Algo muito parecido com o implementado no Matlab para classificação binária.

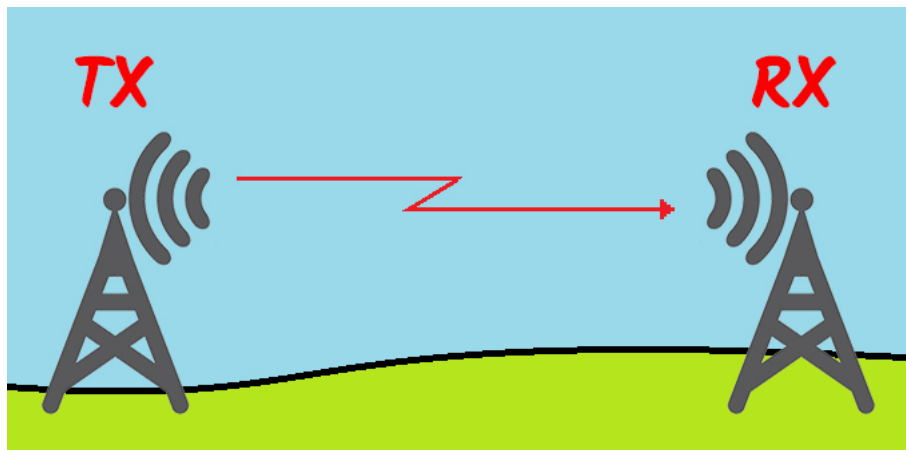


Figura 23: Ilustração representando a simulação de transmissão binária entre Tx e Rx.

A simulação em hardware testou o funcionamento de um sistema de detecção de erro de pacotes usando os detectores INED, CNED, SPC e CHECKSUM. As antenas estavam com distância de variação com 30 metros entre TX e RX em ambiente aberto e foram transmitidos 250 pacotes de dados à recepção para ser avaliado pelo sistema.

Para colocar em prática o sistema de comunicação digital, foram utilizados dois microcontroladores Arduino Uno, duas placas protoboard, dois módulos, um transmissor e

outro receptor como apresentado na Figura 24 operados à uma frequência de 433 MHz, IDE Arduino para programação das placas microcontroladoras e alguns Jump's. A implementação geral dos equipamentos do teste prático é mostrada na Figura 25. A Tabela 4 apresenta as características do módulo RF 433Mhz.

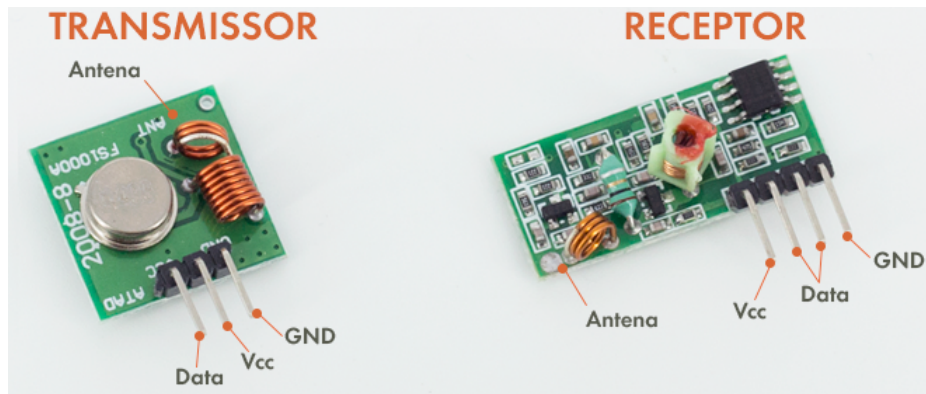


Figura 24: Módulos utilizados na simulação para comunicação sem fio. Fonte:(FILIFELOP,)

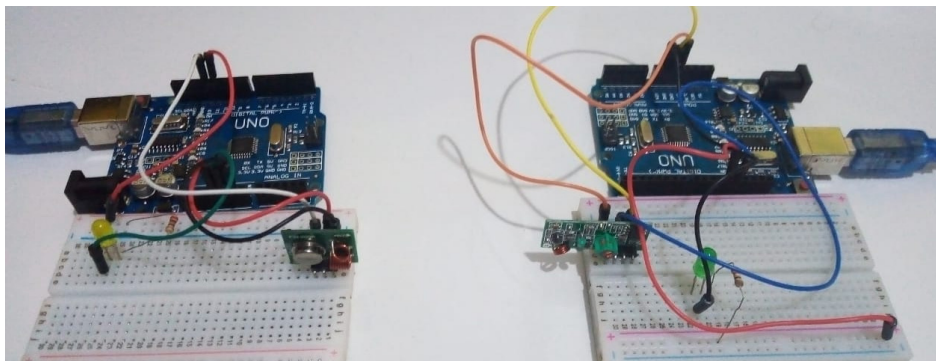


Figura 25: Conjunto transmissor e receptor do sistema de comunicação sem fio.

4.1.5 Constituição do Frame de Comunicação

Na transmissão de bits entre o transmissor e o receptor, é normal ocorrer falta de sincronização e erros de bits. Por isso, em um processo de comunicação é normal formar um enquadramento de bytes ou frame de comunicação. Geralmente o enquadramento é formado por bytes de sincronização (flag) mais a informação útil. Na elaboração do frame, foram utilizados 10 bits para sinalização e 24 bits para informação útil, como representado na Figura 26.

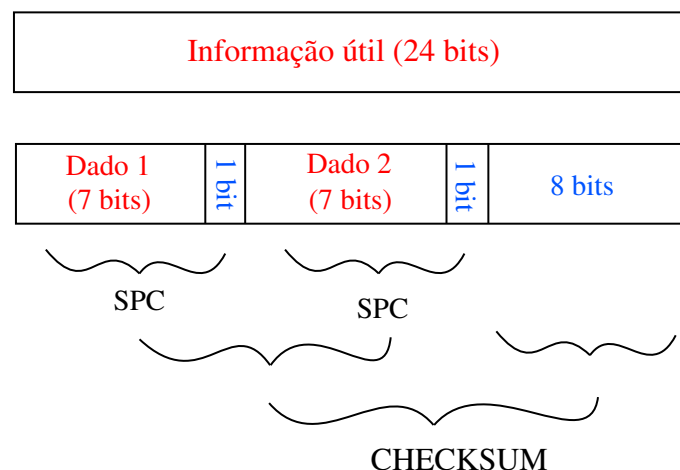
Flag (10 bits)	Informação útil (24 bits)
----------------	---------------------------

Figura 26: Esquema de enquadramento usado na comunicação sem fio.

Tabela 4: Característica do módulo RF 433MHz

Transmissor	Receptor
– Modelo: MX-FS-03V – Alcance: 20-200 metros (conforme voltagem) – Tensão de operação: 3,5-12v – Modo de operação: ASK – Taxa de transferência: 4KB/s – Potência de transmissão: 10mW – Frequência de transmissão: 433MHz – Pinagem: Dados-VCC-GND (Esq.-> Dir.) – Dimensões: 19 x 19mm	– Modelo: MX-05V – Tensão de operação: 5v DC – Corrente de operação: 4mA – Frequência de recepção: 433MHz – Sensibilidade: -105dB – Dimensões: 30 x 14 x 7mm

A informação útil com 24 bits ou 3 bytes de dados foi projetada para ser utilizada com os detectores de erros SPC e o CHECKSUM, além do INED. Como mostrado na Figura. 27. A informação útil tem (Dado 1) e (Dado 2) cada um com 7 bits de informação. A cada 7 bits o SPC acrescenta o bit de paridade, formando um byte com 8 bits. A cada 2 palavras com 8 bits, o CHECKSUM forma uma palavra de código.

**Figura 27:** Projeto de implementação dos detectores de erro na formação da informação útil usado na transmissão.

Os ambientes de simulação computacional em hardware foram implementados em dois computadores: um computador com processador AMD, frequência de 3,0 GHz e 4 GB de memória DDR3; e um notebook com processador i5 de frequência 2.5 GHz e 4GB de memória DDR3. Ambos Windows de 64 bits. Todos os algoritmos foram implementados na linguagem C e executados no IDE Integrated Development. Ambiente para Arduino Uno (versão 2020).

4.2 Resultados

Nesta seção são apresentados resultados gráficos de análise, performance e os resultados para classificadores com opção de rejeição, e os métodos INED e CNED.

4.2.1 Análise Exploratória dos Dados

A Análise Exploratória dos Dados refere-se ao processo crítico de realização de investigações iniciais sobre dados a partir de visualizações dos dados, descobrir padrões, detectar anomalias e testar hipóteses são algumas das tarefas desta área. Verificar suposições com a ajuda de representações gráficas ajudam a entender melhor a base de dados. Com isso, apresentamos uma breve análise dos dados gerados com base no dataset Data1.

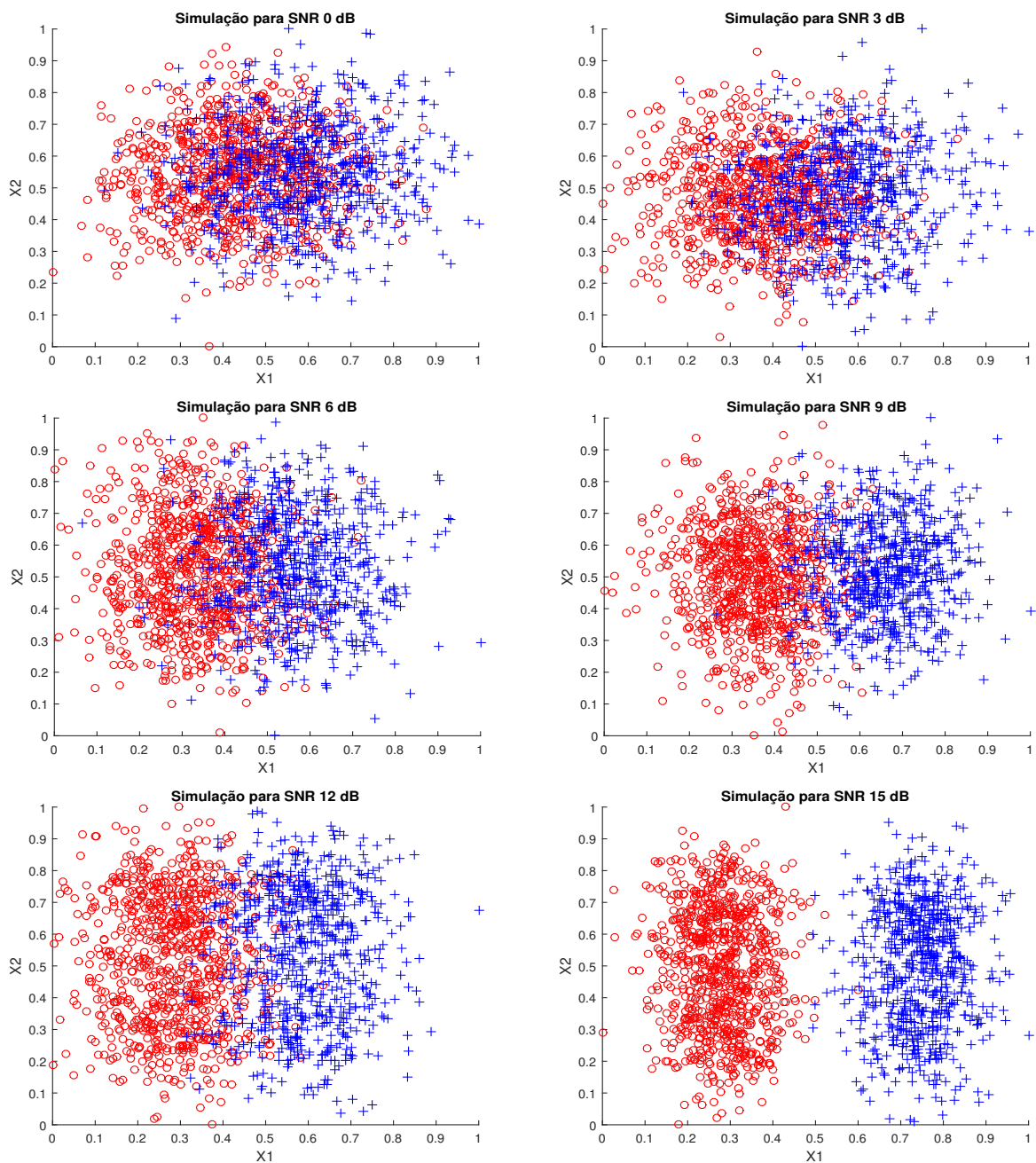
Na Figura 28 é apresentado o gráfico de dispersão dos padrões, variando o SNR de 0 a 15 dB. Os padrões estão normalizados entre 0 a 1. Os padrões em cor vermelha (o) podem ser considerados como sendo padrões de bit (zero) e os padrões de cor azul (+) podem ser considerados como sendo padrões de bit (um). Os eixos X1 e X2 são atributos do dataset Data1. Algumas hipóteses iniciais que podem ser respondidas com a observação do conjunto de dados apresentado na Figura 28 são:

1. Qual(is) SNR(s) mais críticos?, fazendo os padrões ficarem mais misturados e em regiões diferentes?
2. Para qual SNR pode-se afirmar que os padrões de classes diferentes estão completamente separados?
3. Os padrões estão balanceados?

Para responder a primeira pergunta, somente é necessário observar a Figura 28, os padrões ficam mais misturados com SNRs abaixo de 15 dB. Na segunda questão, não é apresentado na Figura 28, mas com SNR de 18 dB em diante foi observado que os dados ficam linearmente separados. Na terceira análise, considerando a Tabela 5, os dados estão balanceados, embora o número de padrões de classe "0" seja um pouco maior que a quantidade de padrões de classe "1", respectivamente 53.27% e 46.73%.

Tabela 5: Balanceamento dos padrões das classes do dataset Data1.

#	Treinamento		Teste		Total
Classe	Quantidade	(%)	Quantidade	(%)	(%)
0	4249	53.12	1078	53.94	53.27
1	3751	46.88	922	46.06	46.73
Total	8000	100.0	2000	100.0	100.0

**Figura 28:** Representação visual do conjunto de padrões binário variando de 0 a 15 dB.

4.2.2 Comparação de Desempenho entre os Classificadores de Padrões com Opção de Rejeição

O objetivo desta análise foi realizar comparações com os algoritmos de inteligência artificial junto da técnica de opção de rejeição. Os algoritmos MLP, RBF, Naive Bayes e ELM são testados para analisar os mais eficientes para classificação binária com opção de rejeição; esses testes geralmente são realizados usando curvas de acurácia-rejeição (A-R). Um método tem melhor desempenho quando atinge maior acurácia para diferentes taxas de rejeição. As curvas (A-R) foram construídas atingindo o limiar de rejeição que minimiza o risco empírico, seguindo os princípios propostos por Chow (CHOW, 1970). Para um determinado custo de rejeição, α , geramos um resultado da curva (A-R). Como consequência dessa estratégia, o limite de decisão não pode ser definido ao construir uma curva (A-R), porém está no intervalo [0;0.5].

Na Figura 29 é apresentado as curvas (A-R) para os algoritmos de classificação que são comparados. Observe em todos os métodos um comportamento que aumenta a acurácia à medida que a taxa de rejeição aumenta. As curvas (A-R) mostram que a eficiência do ELM se destaca considerando ambientes ruidosos mais críticos, ou seja, com SNR menor que 12 dB, o ELM apresentou a melhor taxa de acurácia-rejeição que a mesma comparação com o MLP, RBF e Naive Bayes.

Na Tabela 6 é mostrado resultados quantitativos para acurácia e desvio padrão das simulações. Para SNRs superiores a 12 dB, os classificadores MLP, Naive Bayes e ELM possuem boas métricas de acurácia e desvio padrão na classificação binária, com acurácia superior a 97%. Ainda considerando a Tabela 6 mostra que a eficiência do ELM é maior que os outros algoritmos. Para SNR igual a 15 dB o algoritmo ELM alcançou 100% de acurácia e 0% de desvio padrão com a menor taxa de rejeição.

Na Tabela 7 é apresentado a acurácia e a média de rejeição, também considerando a SNR. Embora a média de rejeição do ELM esteja relativamente um pouco mais alta que os demais classificadores, a sua acurácia supera todos esses classificadores. A Tabela 8 apresenta o desempenho temporal dos algoritmos; nesta análise foi considerado o pior caso, com SNR igual a 0 dB e novamente o ELM apresentou os dados de performance mais otimizados. A execução total do classificador ELM foi três vezes mais rápido que o MLP e RBF, e quase vinte vezes maior que o classificador de Bayes. Considerando os testes dos algoritmos de aprendizado de máquina, vamos considerar para os resultados em diante sempre o uso do modelo com o ELM, que mostrou ser o mais interessante.

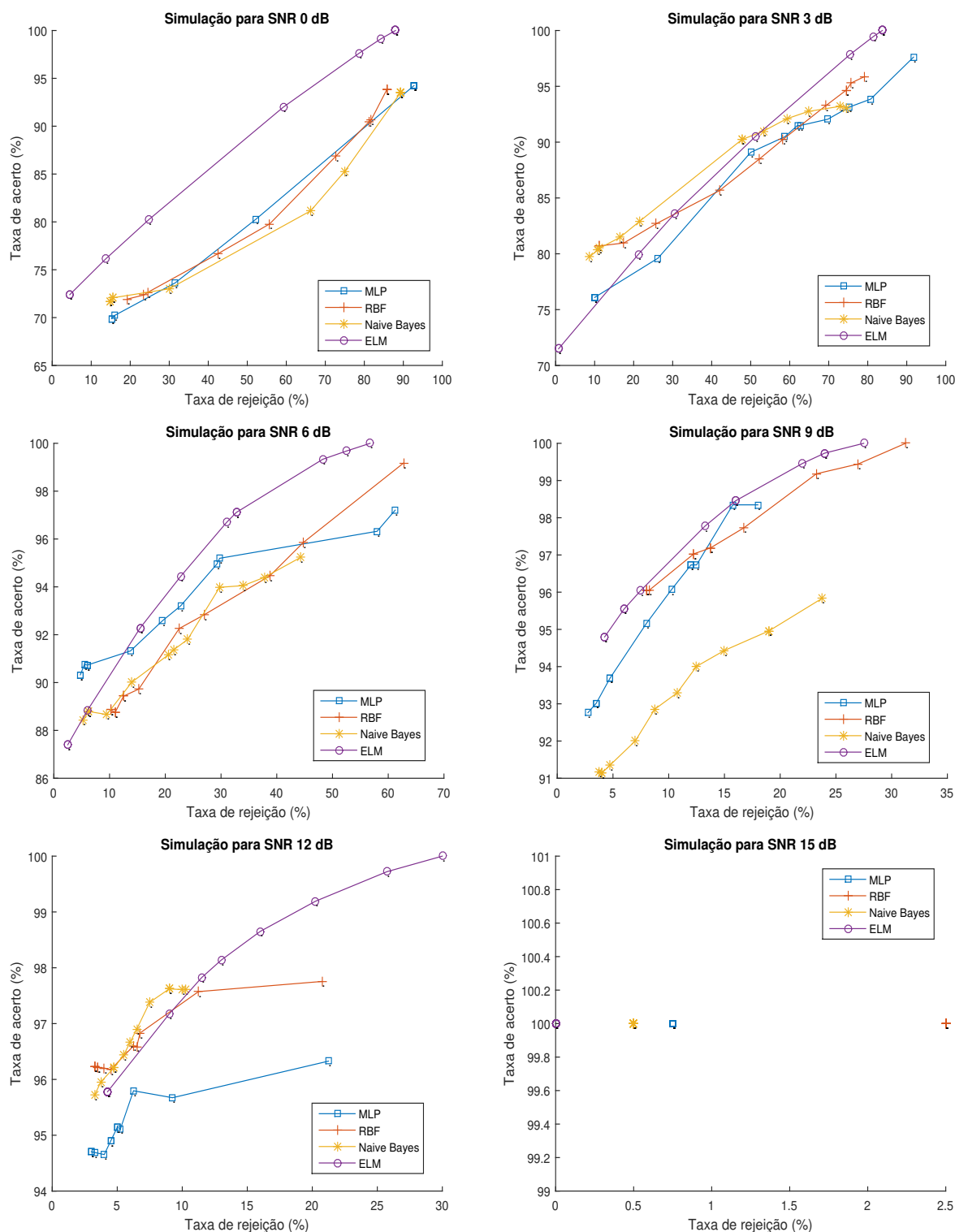


Figura 29: Curvas de desempenho A-R com SNR variando de 0 dB a 15 dB.

4.2.3 Resultados dos Algoritmos de Detecção de Erro com ELM

O objetivo desta análise foi apresentar os resultados para os mecanismos de detecção de erro. Como visto em capítulos anteriores, os algoritmos têm como foco colocar bits redundantes

Tabela 6: Acurácia e desvio padrão de classificadores de padrões (%).

SNR (dB)	MLP	Naive Bayes	RBF	ELM
0	85.49±09.94	82.76±08.20	86.80±02.01	92.56±10.10
3	84.63±09.78	84.95±08.05	81.25±07.78	91.14±10.67
6	89.97±01.36	89.86±03.20	87.70±04.28	92.14±06.78
9	93.28±02.52	95.24±02.19	91.37±03.23	95.43±03.03
12	97.33±00.07	97.88±00.10	93.41±03.15	98.96±00.51
15	99.42±00.24	99.53±00.12	99.41±00.82	100.0±00.00
18	100.0±00.00	100.0±00.00	100.0±00.00	100.0±00.00

Tabela 7: Acurácia e média de rejeição em classificadores de padrões com opção de rejeição (%).

#	MLP		Naive Bayes		RBF		ELM	
SNR (dB)	A	R	A	R	A	R	A	R
0	84.36	58.90	82.77	60.22	85.05	95.71	88.85	49.01
3	83.88	44.77	82.71	42.34	80.62	66.85	84.85	39.42
6	90.43	19.63	91.57	29.85	89.30	42.30	90.60	30.80
9	92.55	20.83	93.72	21.45	91.09	29.87	94.37	26.71
12	97.50	01.28	97.95	02.31	93.44	23.44	98.97	06.80
15	99.73	00.74	99.81	02.47	99.81	00.50	100.0	00.01
18	100.0	00.00	100.0	00.00	100.0	00.00	100.0	00.00

Tabela 8: Tempo em segundos da performance para 1 realização e com SNR em 0 dB.

Etapas	MLP	Naive Bayes	RBF	ELM
Treinamento	08.65	50.85	08.64	01.65
Testes/Avaliações	00.75	06.65	01.15	00.68
Acurácia	00.45	00.85	00.45	00.47
Desvio Padrão	00.20	00.31	00.16	00.17
Total	10.05	58.66	10.40	02.97

nos blocos de informação, ainda na etapa de codificação do transmissor. Para que na recepção os blocos possam passar por uma verificação de integridade. Porém mesmo com a inserção de bits redundantes não é possível evitar o erro porque os próprios bits redundantes dos detectores podem chegar na recepção com erros. Na simulação foi utilizado como classificador binário a rede neural ELM. Utilizando os mesmos parâmetros no classificador, mudando somente a implementação dos detectores.

Na Tabela 9 são apresentados os resultados para o detector SPC. É o detector mais irregular entre os apresentados aqui. Mas tudo isso se deve à forma de implementação. Se o erro for par e sua implementação for ímpar, o SPC erra na detecção. Por causa desse problema

mencionado, para os blocos recebidos de ambientes ruidosos abaixo de SNR 12 dB possuem alta probabilidade de serem erroneamente não detectados. Para o SNR 0 dB, o SPC informou que 45.00% dos blocos estão **certos**, porém somente 6.50% dos blocos estavam **realmente certos**. Ou seja, o SPC erra muito no cálculo de integridade do bloco. Para os ambientes com SNR de 3, 6 e 9 dB a mesma situação aconteceu. Nos testes realizados a convergência do detector somente aconteceu a partir dos ambientes de SNR 15 dB em diante.

Tabela 9: Acurácia e desvio padrão de simulação de transmissão com SPC (%).

SNR (dB)	Bits Certos	Blocos Certos	Blocos Realmente Certos
0	69.17±00.96	45.00±04.73	06.50±02.49
3	77.68±00.77	50.37±05.52	11.13±02.36
6	84.61±00.63	56.25±04.25	21.13±02.98
9	91.19±00.53	66.25±03.09	45.00±02.81
12	97.06±00.25	78.50±02.21	76.00±02.21
15	99.71±00.14	97.37±01.28	97.37±01.28
18	100.0±00.00	100.0±00.00	100.0±00.00

A Tabela 10 apresenta os resultados para o detector CHECKSUM. Os resultados para a detecção com esse modelo são equilibrados. A convergência do detector começa com o SNR em 12 dB, ou seja, nesta simulação os blocos ou frames recebidos pelo sistema de recepção tem grande possibilidade de serem bem detectados. Mesmo com SNRs abaixo de 12 dB, os valores de detecção são bem similares entre os blocos considerados corretos pelo algoritmo de detecção e blocos confirmados como certos.

Tabela 10: Acurácia e desvio padrão de simulação de transmissão com CHECKSUM (%).

SNR (dB)	Bits Certos	Blocos Certos	Blocos Realmente Certos
0	68.71±00.82	00.00±00.00	00.00±00.00
3	76.23±00.73	00.50±00.71	00.00±00.00
6	84.61±00.44	06.00±02.26	01.00±02.02
9	91.82±00.57	16.00±04.16	14.60±03.33
12	97.80±00.26	51.80±06.04	51.80±06.04
15	99.13±00.10	89.20±02.34	89.20±02.34
18	100.0±00.00	100.0±00.00	100.0±00.00

Na Tabela 11 é mostrado os resultados para o detector CRC. A conclusão que se pode fazer é que os resultados para a detecção com esse modelo são equilibrados e parecidos com o CHECKSUM. Embora o CHECKSUM tenha tido resultados melhores e convergência com SNR em 9 dB. O CRC a partir de SNR em 9 dB tem valores de detecção de erro bem próximos

do real testado. A convergência do detector começa com o SNR em 15 dB, ou seja, nesta simulação os blocos recebidos pelo sistema de recepção tem grande possibilidade de serem bem verificados.

Tabela 11: Acurácia e desvio padrão de simulação de transmissão com CRC (%).

SNR (dB)	Bits Certos	Blocos Certos	Blocos Realmente Certos
0	68.37±00.65	00.20±00.69	00.00±00.00
3	76.02±00.78	06.15±01.45	01.10±01.25
6	85.10±00.23	08.80±02.10	07.50±01.43
9	91.78±00.41	37.35±02.92	34.85±03.01
12	97.74±00.25	65.20±03.81	64.20±03.76
15	99.83±00.04	96.35±01.53	96.35±01.53
18	100.0±00.00	100.0±00.00	100.0±00.00

4.2.4 A Eficiência da Classificação com Opção de Rejeição em Detectar Erro de Bit

Nesta simulação computacional foi observado se o sistema com opção de rejeição consegue classificar **bit errado** como **bit que deve ser rejeitado**. Por exemplo, em uma sequência de transmissão de 100 bits, 20 bits são recebidos com erros e a opção de rejeição deve informar que 20 bits ou algo próximo disso devem ser rejeitados. Na simulação computacional foram utilizados também 100 bits de dados.

Na Tabela 12 apresenta os resultados desta simulação. E como resultado fica evidente que a taxa de bits rejeitados é muito próxima da taxa de bits errados. A diferença para SNR em 0 dB alcança -2.21 de diferença. Para 3 dB alcança diferença de +1.53. O que mostra que em certas situações o modelo prefere rejeitar que se arriscar em errar. Para o SNR em 15 dB a taxa de diferença ficou em 0, significando que o modelo nesta configuração consegue rejeitar somente os bits errados.

A Tabela 13 apresenta a mesma simulação, porém com apenas uma realização executada. Notamos que o modelo com SNR em 12 dB conseguiu identificar bits errados como bits que devem ser rejeitados. Consideramos a Tabela 12 e a Tabela 13 as análises mais importantes deste trabalho de mestrado, porque é aqui que demonstramos que o processo de identificação do erro pode ser automatizado com a opção de rejeição.

Na Tabela 14 é apresentado os primeiros resultados para o detector INED. A convergência do detector INED começa com o SNR em 15 dB. Porém, a observação inicial que podemos fazer

Tabela 12: Eficiência do ELM com a técnica de opção de rejeição em detectar erros na transmissão de bits (50 realizações).

SNR (dB)	Bits Certos	Bits Errados	Bits Rejeitados	Diferença
0	67.12	32.87	30.66	-02.21
3	78.69	21.31	22.84	+01.53
6	84.91	15.09	16.48	+01.39
9	90.63	09.38	10.38	+01.00
12	97.31	02.69	02.47	00.22
15	99.69	00.31	00.31	00.00
18	100.0	00.00	00.00	00.00

Tabela 13: Eficiência do ELM com a técnica de opção de rejeição em detectar erros na transmissão de bits (1 realização).

SNR (dB)	Bits Certos	Bits Errados	Bits Rejeitados	Diferença
0	67	33	31	-2
3	78	22	23	+1
6	84	16	17	+1
9	90	10	11	+1
12	97	3	3	0
15	99	1	1	0
18	100	0	0	0

é que os resultados para a detecção com esse modelo são equilibrados, mesmo com valores de SNRs menores que 15 dB. E novamente nota-se um comportamento diferente dos outros detectores que preferem arriscar o erro, ao invés de ser mais moderado como o INED.

4.2.5 Comparação de Detecção de Erro entre SPC, CHECKSUM, CRC e INED

Esta seção tem o objetivo de fazer comparações entre os detectores de erros conhecidos e o INED. Aqui apresentamos resultados considerando os dataset Data2 e Data3 com 100 blocos de informação, além do tempo gasto para verificação de integridade. Os resultados aqui estão considerando 1 realização.

A Tabela 15 apresenta os resultados da comparação entre os detectores SPC e INED. Observe que o detector SPC deixa passar muito bloco com erro para SNRs abaixo de 12 dB, como visto na seção anterior. Por outro lado, o INED obteve resultados eficientes para todos os ambientes ruidosos. Nos experimentos realizados a convergência do detector INED ocorre

Tabela 14: Acurácia e desvio padrão de simulação de transmissão com INED (%).

SNR (dB)	Bits Certos	Blocos Confirmado	Blocos Certos
0	69.91±00.92	02.75±02.49	04.25±02.37
3	77.68±00.77	14.75±03.43	13.25±02.65
6	84.61±00.63	18.75±04.89	19.75±03.43
9	91.84±00.65	47.00±15.54	52.25±04.78
12	97.41±00.34	75.25±10.30	78.75±04.60
15	99.81±00.14	98.50±01.29	98.50±01.29
18	100.0±00.00	100.0±00.00	100.0±00.00

a partir dos ambientes SNR 12 dB, enquanto o SPC somente após SNR 12 dB. O tempo de inferência do INED proposto foi semelhante e em vários casos mais rápido do que o SPC.

Tabela 15: Desempenho dos detectores SPC e INED sob diferentes condições de ruído.

SNR (dB)	Blocos recebidos (100) Blocos errados	Detecção de erro		Tempo (ms)	
		SPC	INED	SPC	INED
0	93	55	96	23	23
3	88	50	86	21	18
6	79	45	81	19	17
9	55	35	53	18	17
12	24	22	24	18	17
15	3	3	3	18	15
18	0	0	0	18	14

A Tabela 16 apresenta os resultados da comparação entre os detectores CHECKSUM e INED. Os resultados da detecção com as duas técnicas de detecção são semelhantes. A convergência do detector CHECKSUM começa com o SNR em 12 dB enquanto o INED após 12 dB. Porém, a eficiência de detecção com o INED é mais convergente com a realidade dos erros. Com 6 dB o CHECKSUM detectou 94 erros, enquanto o INED detectou 100 erros e o sistema recebeu 99 erros. Portanto, em casos específicos, o INED é mais confiável. Novamente, o tempo de processamento de detecção do INED é menor em relação ao CHECKSUM e SPC.

Por fim, a Tabela 17 mostra os resultados da comparação entre o método INED e CRC. Embora o CRC esteja entre as técnicas tradicionais de detecção de erros mais eficientes, é mostrado que o INED foi mais eficiente do que o CRC. A convergência do detector INED começa com o SNR em 12 dB enquanto o CRC após 12 dB. Além disso, o tempo de processamento de detecção do INED é menor que o CRC, pelo motivo de que ele precisa realizar menos cálculos para verificar a veracidade do pacote ou bloco.

Tabela 16: Desempenho para os detectores CHECKSUM e INED sob diferentes condições de ruído.

SNR (dB)	Blocos recebidos (100) Blocos errados	Detecção de erro		Tempo (ms)	
		CHECKSUM	INED	CHECKSUM	INED
0	100	100	100	58	24
3	100	99	100	46	18
6	99	94	100	46	17
9	88	84	89	45	17
12	49	49	50	36	17
15	11	11	11	36	18
18	0	0	0	31	17

Tabela 17: Desempenho para os detectores CRC-8-CCITT e INED sob diferentes condições de ruído.

SNR (dB)	Blocos recebidos (100) Blocos errados	Detecção de erro		Tempo (ms)	
		CRC	INED	CRC	INED
0	100	99	100	81	24
3	99	93	100	56	22
6	92	91	94	47	20
9	66	63	70	43	17
12	36	35	36	34	17
15	4	4	4	30	15
18	0	0	0	25	15

4.2.6 Detecção de Erro com CNED

O método de opção de rejeição pode ser aplicado a uma técnica de detecção já existente utilizando o CNED. Como forma de aumentar uma camada de segurança, foi analisado que os detectores CHECKSUM E CRC são eficientes na verificação e integridade dos dados. Porém, o método SPC mostrou não ser confiável para SNRs abaixo de 15 dB. Logo, foi aplicado o método de rejeição como forma de aumentar a eficiência do SPC, com isso podemos combinar o SPC e CNED (SPC-CNED).

Na Tabela 18 mostra os resultados encontrados com a combinação do método de bit paridade com a opção de rejeição. É possível notar que os resultados estão mais reais em relação aos dados de blocos confirmados com blocos certos. Com SNR de 15 dB o método já está convergente para acertar os blocos errados. A Tabela 19 mostra o resultado para 1 realização, nota-se que o SPC-CNED foi mais eficiente que o SPC sozinho, ou seja, o método CNED ajuda o SPC a ser mais eficiente. O lado desvantajoso em relação a essa combinação é em relação ao tempo que aumenta considerando a união de técnicas.

Tabela 18: Acurácia e desvio padrão de simulação de transmissão com SPC-CNED (%).

SNR (dB)	Bits Certos	Blocos Confirmado	Blocos Certos
0	69.91±00.92	02.75±02.49	04.25±02.37
3	77.68±00.77	14.75±03.43	13.25±02.65
6	84.61±00.63	18.75±04.89	19.75±03.43
9	91.84±00.65	47.00±15.54	52.25±04.78
12	97.41±00.34	75.25±10.30	78.75±04.60
15	99.81±00.14	98.50±01.29	98.50±01.29
18	100.0±00.00	100.0±00.00	100.0±00.00

Tabela 19: Desempenho dos detectores SPC e SPC-CNED sob diferentes condições de ruído.

SNR (dB)	Blocos recebidos (100) Blocos errados	Detecção de erro		Tempo (ms)	
		SPC	SPC-CNED	SPC	SPC-CNED
0	92	55	95	24	32
3	87	51	86	22	31
6	79	47	80	19	29
9	54	36	53	18	29
12	23	20	23	18	27
15	3	3	3	18	25
18	0	0	0	18	24

4.2.7 Resultados da Simulação em Hardware

Como mencionado, o principal objetivo de fazer a simulação em hardware é mostrar que o método de opção de rejeição pode ser aplicado na prática. Na Figura 30 é apresentado o monitor serial do receptor do Arduino. No monitor são apresentados os bits de sinalização (1010111011), a informação transmitida e bits redundantes (negrito) (1011000101111101111010011). Na Figura 30 são apresentados o resultado pelos detectores de erro:

1. Na primeira transmissão de (bloco), é possível notar que o detector CHECKSUM retornou a transmissão errada, porém os dados estavam certos. Isso ocorreu porque o código da soma chegou errado na recepção. SPC e INED constataram isso;
2. Na terceira transmissão o SPC informou que os dois bytes de dados recebidos estavam certos, porém CHECKSUM e INED informaram que estavam errados (rejeitados). Em questão, SPC verificou de forma errada;
3. Na segunda e quarta transmissão os dados estavam corretos, e SPC, CHECKSUM e

INED informaram que estavam certos.

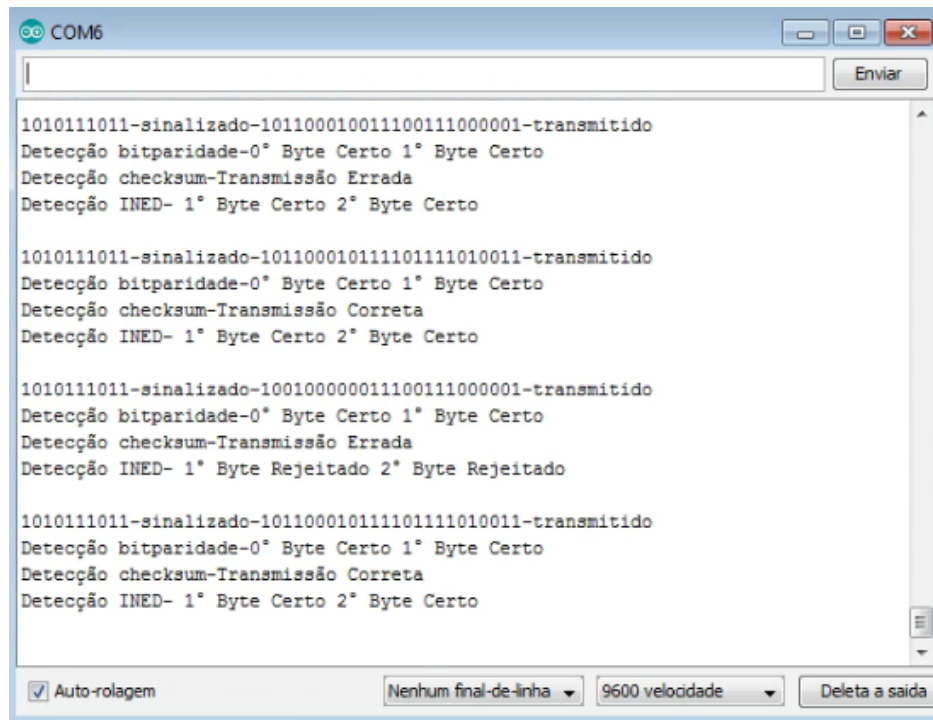


Figura 30: Monitor serial do Arduino: Análise dos resultados de saída do receptor.

Na realização da simulação em hardware foram verificados os blocos recebidos. São considerados 250 recebimentos de blocos para cada execução. Em média a cada execução retornou com 215 blocos com a informação correta e outros 35 com os dados errados considerando os bits redundantes. Foi verificada a eficiência de detecção em equipamento real entre os detectores SPC, CHECKSUM, INED e SPC-INED.

Tabela 20: Simulação de transmissão de 250 blocos no teste prático (média de 215 certos e 35 errados).

Execução	Blocos com erros	SPC	CHECKSUM	INED	SPC-CNED
1 ⁰	36	21	34	37	37
2 ⁰	35	25	35	35	35
3 ⁰	37	24	33	37	37
4 ⁰	34	25	32	35	35
5 ⁰	34	28	34	34	34
6 ⁰	35	21	35	35	35
7 ⁰	36	29	36	36	36
8 ⁰	35	20	34	35	35
9 ⁰	34	23	33	35	35
10 ⁰	34	24	31	34	34

CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho de mestrado, propomos dois métodos baseados no classificador ELM com opção de rejeição para aplicação na detecção de erros de bits. Os métodos propostos chamados INED e CNED, foram testados em seis ambientes ruidosos diferentes, considerando a relação sinal-ruído. Os métodos foram testados considerando as técnicas de detecção de erro presentes nos sistemas de transmissão e recepção de dados. Os classificadores de padrões MLP, RBF, Bayes e ELM são comparados e verificados entre si ao usar a opção rejeição, e é mostrado que o ELM é mais eficiente que os outros classificadores mencionados, em métricas e velocidade.

O uso da classificação com opção de rejeição apresentou resultados promissores para este trabalho, os métodos INED e CNED foram eficientes em quase todos os casos simulados. O INED mostrou que mesmo sem combinar com outras técnicas e sem a inserção de bits redundantes, possui métricas de avaliação comparáveis ou mais eficientes que os detectores SPC, CHECKSUM e CRC testados. O CNED combinado com outros detectores aumenta a eficiência deles em detectar erros. O classificador ELM com opção de rejeição foi melhor que os outros algoritmos de classificação simulados, mostrando o porquê do autor usar o ELM. O detector de erros de bits mostrou que quando o sistema não rejeita um bit, ele tem uma grande chance de obter o resultado final corretamente, alcançando uma acurácia muito alta (acima de 92%) nas situações mais críticas de SNR.

A simulação em hardware dos detectores INED e CNED mostraram em comparações que podem ser usados como detectores de erros de bits, já que a estrutura do detector é fácil de implementar. Os detectores foram aplicados para detecção de erros de bits, mas podem ser aplicados para muitas outras aplicações. Até onde sabemos, é a primeira vez que essa abordagem é desenvolvida com opção de rejeição e com o algoritmo ELM.

Para trabalhos futuros, pretendemos aperfeiçoar os métodos INED e CNED, testar outros algoritmos de aprendizado de máquina e fazer novas comparações, analisando sempre o desempenho e velocidade do método. Além de fazer comparações com outros métodos de detecção de erro. Também pretendemos verificar a usabilidade de usá-los em conjunto com algoritmos de correção de erros e implementar os métodos em soluções de internet das coisas.

Referências Bibliográficas

- AHMADI, Ali; OMATU, Sigeru; FUJINAKA, Toru; KOSAKA, Toshihisa. Improvement of reliability in banknote classification using reject option and local pca. **Information Sciences**, Elsevier, v. 168, n. 1-4, p. 277–293, 2004.
- ALENCAR, Alisson SC; ROCHA NETO, Ajalmar R; GOMES, João Paulo P. A new pruning method for extreme learning machines via genetic algorithms. **Applied Soft Computing**, Elsevier, v. 44, p. 101–107, 2016.
- ALMEIDA, Ezilda Duarte. **Classificação ordinal com opção de rejeição**. 2010.
- ALRSHAH, Mohamed A; OTHMAN, Mohamed; ALI, Borhanuddin; HANAPI, Zurina Mohd. Agile-sd: A linux-based tcp congestion control algorithm for supporting high-speed and short-distance networks. **Journal of Network and Computer Applications**, Elsevier, v. 55, p. 181–190, 2015.
- AYDOS, Gökçe; FEY, Görschwin. Empirical results on parity-based soft error detection with software-based retry. In: IEEE. **2015 Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC)**. [S.l.], 2015. p. 1–4.
- BARBARÁ, Daniel; GOEL, Rajni; JAJODIA, Sushil. Using checksums to detect data corruption. p. 136–149, 2000.
- BARTLETT, Peter L; WEGKAMP, Marten H. Classification with a reject option using a hinge loss. **Journal of Machine Learning Research**, v. 9, n. Aug, p. 1823–1840, 2008.
- BIZON, Katarzyna; CONTINILLO, Gaetano; LOMBARDI, Simone; MANCARUSO, Ezio; VAGLIECO, Bianca M. Ann-based virtual sensor for on-line prediction of in-cylinder pressure in a diesel engine. Elsevier, v. 33, p. 763–768, 2014.
- BOUNSIAR, Abdenour; GRALL, Edith; BEAUSEROY, Pierre. A kernel based rejection method for supervised classification. **International Journal of Computational Intelligence**, v. 3, n. 4, p. 312–321, 2006.
- BRAGA, Antonio de Padua; LUDERMIR, André Ponce de Leon F. De Carvalho; BERNARDA, Teresa. **Redes neurais artificiais: teoria e aplicações 2^a ed.** [S.l.]: Livros Técnicos e Científicos, 2011.
- BROOMHEAD, DS; LOWE, D. Multivariable functional interpolation and adaptive networks, complex systems, vol. 2. 1988.
- CHOW, C. An optimum character recognition system using decision functions. **IRE Transactions on Electronic Computers**, IEEE, n. 4, p. 247–254, 1957.

CHOW, C. On optimum recognition error and reject tradeoff. **IEEE Transactions on information theory**, IEEE, v. 16, n. 1, p. 41–46, 1970.

CONDESSA, Filipe; BIOUCAS-DIAS, José; CASTRO, Carlos A; OZOLEK, John A; KOVAČEVIĆ, Jelena. Classification with reject option using contextual information. p. 1340–1343, 2013.

CYBENKO, George. Approximation by superpositions of a sigmoidal function. **Mathematics of control, signals and systems**, Springer, v. 2, n. 4, p. 303–314, 1989.

DONG, Xiguang; HE, Yongqiang. Crc algorithm for embedded system based on table lookup method. **Microprocessors and Microsystems**, Elsevier, v. 74, p. 103049, 2020.

FATH, Aref Hashemi; MADANIFAR, Farshid; ABBASI, Masood. Implementation of multilayer perceptron (mlp) and radial basis function (rbf) neural networks to predict solution gas-oil ratio of crude oil systems. **Petroleum**, Elsevier, v. 6, n. 1, p. 80–91, 2020.

FILIPEFLOP. **Comunicação Wireless com Módulo RF 433MHz**. Disponível em: <<https://www.filipeflop.com/blog/modulo-rf-transmissor-receptor-433mhz-arduino>>. Acesso em: 02 mar. 2020.

FUMERA, Giorgio; ROLI, Fabio. Support vector machines with embedded reject option. p. 68–82, 2002.

GRAMI, Ali. Chapter 3-signals, systems, and spectral analysis. **Introduction to Digital Communications**, p. 41–150, 2016.

HE, Xiangdong; LAPEDES, Alan. Nonlinear modeling and prediction by successive approximation using radial basis functions. **Physica D: Nonlinear Phenomena**, Elsevier, v. 70, n. 3, p. 289–301, 1994.

HUANG, Guang-Bin; ZHU, Qin-Yu; SIEW, Chee-Kheong. Extreme learning machine: theory and applications. **Neurocomputing**, Elsevier, v. 70, n. 1-3, p. 489–501, 2006.

HUO, Yuanhong; LI, Xiaoyang; WANG, Wei; LIU, Dake. High performance table-based architecture for parallel crc calculation. p. 1–6, 2015.

HUO, Yuanhong; LIU, Dake. High-throughput bit processor for cryptography, error correction, and error detection. **Microprocessors and Microsystems**, Elsevier, v. 61, p. 207–216, 2018.

KWON, Beom; LEE, Sanghoon. Error detection algorithm for lempel-ziv-77 compressed data. **Journal of Communications and Networks**, IEEE, v. 21, n. 2, p. 100–112, 2019.

LIM, JunSeok. Recursive dls solution for extreme learning machine-based channel equalizer. **Neurocomputing**, Elsevier, v. 71, n. 4-6, p. 592–599, 2008.

LIU, Jun; MEI, Kai; ZHANG, Xiaochen; MA, Dongtang; WEI, Jibo. Online extreme learning machine-based channel estimation and equalization for ofdm systems. **IEEE Communications Letters**, IEEE, v. 23, n. 7, p. 1276–1279, 2019.

LIU, Xia; LIN, Shaobo; FANG, Jian; XU, Zongben. Is extreme learning machine feasible? a theoretical assessment (part i). **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 26, n. 1, p. 7–20, 2014.

MESQUITA, Diego PP; ROCHA, Lincoln S; GOMES, João Paulo P; ROCHA NETO, Ajalmar R. Classification with reject option for software defect prediction. **Applied Soft Computing**, Elsevier, v. 49, p. 1085–1093, 2016.

MURPHY, Kevin P *et al.* Naive bayes classifiers. **University of British Columbia**, v. 18, p. 60, 2006.

OLIVEIRA, Adonias Caetano de. **Máquina de aprendizagem mínima com opção de rejeição**. 2016.

PEDRONI, Volnei A. **Eletrônica digital moderna e VHDL**. [S.l.: s.n.], 2010.

PILLAI, Ignazio; FUMERA, Giorgio; ROLI, Fabio. Multi-label classification with a reject option. **Pattern Recognition**, Elsevier, v. 46, n. 8, p. 2256–2266, 2013.

ROCHA NETO, Ajalmar R. **SINPATCO II: Novas Estratégias de Aprendizado de Máquinas para Classificação de Patologias da Coluna Vertebral**. Tese (Doutorado) — Universidade Federal do Ceará, 2011.

ROCHA NETO, Ajalmar R; SOUSA, Ricardo; BARRETO, Guilherme de A; CARDOSO, Jaime S. Diagnostic of pathology on the vertebral column with embedded reject option. p. 588–595, 2011.

ROCHOL, Juergen; ABREVIATURAS, Lista de. **Comunicação de Dados**. [S.l.: s.n.], 2012.

RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

SCHILLER, Frank; MATTES, Tina; WEBER, Uwe; MATTES, Rainer. Undetectable manipulation of crc checksums for communication and data storage. p. 1–9, 2008.

SHUKLA, Sanyam; RAGHUWANSHI, Bhagat Singh. Online sequential class-specific extreme learning machine for binary imbalanced learning. **Neural Networks**, Elsevier, v. 119, p. 235–248, 2019.

SOUSA, Ricardo; MORA, Beatriz; CARDOSO, Jaime S. An ordinal data method for the classification with reject option. p. 746–750, 2009.

SOUSA, Ricardo Gamelas; ROCHA NETO, Ajalmar R; CARDOSO, Jaime S; BARRETO, Guilherme A. Robust classification with reject option using the self-organizing map. **Neural Computing and Applications**, Springer, v. 26, n. 7, p. 1603–1619, 2015.

SU, Te-Jen; CHENG, Jui-Chuan; YU, Chia-Jung. An adaptive channel equalizer using self-adaptation bacterial foraging optimization. **Optics Communications**, Elsevier, v. 283, n. 20, p. 3911–3916, 2010.

YANG, Rong; YANG, Ling; ZHANG, Jiliang; SUN, Chao; CONG, Wensheng; ZHU, Shiqing. Blind equalization of qam signals via extreme learning machine. p. 34–39, 2018.

ZHANG, Yongshan; WU, Jia; ZHOU, Chuan; CAI, Zhihua. Instance cloned extreme learning machine. **Pattern Recognition**, Elsevier, v. 68, p. 52–65, 2017.

ZHAO, Haiquan; ZENG, Xiangping; ZHANG, Jiashu; LIU, Yangguang; WANG, Xiaomin; LI, Tianrui. A novel joint-processing adaptive nonlinear equalizer using a modular recurrent neural network for chaotic communication systems. **Neural networks**, Elsevier, v. 24, n. 1, p. 12–18, 2011.

ZHAO, Haiquan; ZENG, Xiangping; ZHANG, Xiaoqiang; ZHANG, Jiashu; LIU, Yangguang; WEI, Tiao. An adaptive decision feedback equalizer based on the combination of the fir and flnn. **Digital Signal Processing**, Elsevier, v. 21, n. 6, p. 679–689, 2011.

ZHONG, Zonglin; HU, Wengui. Error detection and control of iiot network based on crc algorithm. **Computer Communications**, Elsevier, v. 153, p. 390–396, 2020.