



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RELATÓRIO DA 1ª UNIDADE
IMPLEMENTAÇÃO DO ALGORITMO DO MÉTODO K NEAREST
NEIGHBOR (KNN) PARA VERIFICAÇÃO DE ESTADOS EMOCIONAIS

GRUPO Nº 1

MARIA ALICE DE MELO SOUSA: Nº 20200149247

RIJKAARD MELO: Nº 2016018734

SAMUEL CAVALCANTI: Nº 20200149318

TIAGO FELIPE DE SOUZA: Nº 20190153105

Natal-RN
2022

MARIA ALICE DE MELO SOUSA: Nº 20200149247

RIJKAARD MELO: Nº 2016018734

SAMUEL CAVALCANTI: Nº 20200149318

TIAGO FELIPE DE SOUZA: Nº 20190153105

IMPLEMENTAÇÃO DO ALGORITMO DO MÉTODO K NEAREST NEIGHBOR (KNN) PARA VERIFICAÇÃO DE ESTADOS EMOCIONAIS

Primeiro Relatório de apresentação à disciplina de Sistemas Digitais, que corresponde parcialmente a avaliação da 1ª unidade do semestre 2021.2 do curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte, sob orientação do **Prof. Dr. Marcelo Augusto Costa Fernandes**.

Professor: Dr. Marcelo Augusto Costa Fernandes.

Natal-RN
2022

RESUMO

O seguinte relatório apresenta os detalhes de produção do 1º projeto da disciplina de Sistemas Digitais (DCA 0119) na Universidade Federal do Rio Grande do Norte (UFRN), ministrada pelo Professor Dr. Marcelo Augusto Costa Fernandes.

O objetivo do projeto é implementar o algoritmo do método K nearest neighbours (KNN), e que será utilizado para definir os estados emocionais de uma pessoa, sendo: divertido, entendido, relaxado e assustado. Ao longo deste, foi discutido a importância da preparação do dataset e codificação dos rótulos para obtenção dos resultados.

Por fim, o projeto foi útil para nós, discentes, entendermos o processo prático de criação de um modelo baseado em Inteligência Artificial, e que serviu para embasar a teoria vista durante as aulas e aplicar com um objetivo profissional.

Palavras-chave: Sistemas Digitais; K Nearest Neighbours; estados emocionais; dataset.

Lista de Figuras

1	Montagem do circuito	9
2	Parte do código que está sendo medido o tempo	11

Sumário

1	INTRODUÇÃO	6
2	REFERENCIAL TEÓRICO	7
2.1	ATMEGA328P	7
2.2	KNN	8
3	METODOLOGIA	9
4	RESULTADOS	11
5	CONCLUSÃO	13

1 INTRODUÇÃO

Pesquisadores da saúde pública e da comunidade científica tecnológica, já estão fazendo uso de técnicas de Aprendizado de Máquina e Redes Neurais Artificiais agregadas aos sistemas computacionais de diagnósticos, com o intuito de melhorar o desempenho e as atividades desenvolvidas pelos profissionais de saúde, para o diagnóstico dos estados emocionais do ser humano, e propõe o desenvolvimento de um sistema inteligente pra isso. Dentre estas técnicas, a classificação é uma das mais importantes e essenciais tarefas.

O objetivo deste trabalho é construir um algoritmo de Aprendizagem Supervisionada chamado método de K Nearest Neighbor para ajudar os profissionais da saúde dar celeridade a diagnósticos e desenvolver o seu trabalho em classificar estados emocionais baseado em dados, informações de sensores de monitoramento de batimentos cardíacos, volume de sangue, atividade eletrodérmica e respiração.

2 REFERENCIAL TEÓRICO

2.1 ATMEGA328P

Um microcontrolador é um pequeno computador em um único chip de circuito integrado (IC) de óxido de metal semicondutor (MOS). Um microcontrolador contém uma ou mais CPUs (núcleos de processador) juntamente com memória e periféricos de entrada/saída programáveis.

O ATmega328P é um microcontrolador de chip único criado pela Atmel na família megaAVR (mais tarde a Microchip Technology adquiriu a Atmel em 2016). Possui um núcleo de processador RISC (Reduced Instruction Set Computer) que é um computador projetado para simplificar as instruções individuais dadas ao computador para realizar uma tarefa, de 8 bits e arquitetura Harvard modificado. Esta tecnologia combina memória flash ISP de 32 KB com recursos de leitura durante a gravação, EEPROM de 1 KB (Electrically Erasable Programmable Read-Only Memory), é um tipo de memória não volátil usada em computadores, integrada em microcontroladores para cartões inteligentes e sistemas remotos sem chave e outros dispositivos eletrônicos para armazenar quantidades relativamente pequenas de dados, permitindo que bytes individuais sejam apagados e reprogramados, SRAM de 2 KB, 23 linhas de E/S de uso geral, 32 registradores de trabalho de uso geral, 3 temporizadores/ contadores com modos de comparação, interrupções internas e externas, uma interface serial de 2 fios orientada a Byte, porta serial SPI, conversor A/D de 6 canais e 10 bits (8 canais em pacotes TQFP e QFN / MLF), programável temporizador watchdog, temporizador eletrônico ou de software que é usado para detectar e se recuperar de problemas de funcionamento. Os temporizadores watchdog são amplamente utilizados em computadores para facilitar a correção automática de falhas temporárias de hardware, com oscilador interno e 5 modos de economia de energia selecionáveis por software. Opera entre 1,8 e 5,5 volts e atinge uma taxa de transferência de aproximadamente 1 MIPS/MHz (Milhões de Instruções Por Segundo/MegaHertz).

Para a entrada foram implementados 4 sensores, representados por potenciômetros, alimentados por uma diferença de potencial de 5V, do próprio Arduino, com os pinos de entrada analógica de A0 até A3 do Arduino UNO, que correspondem aos pinos PC0 até o PC3 do microcontrolador do ATmega328P, representando a captação dos dados. Para a saída, foram utilizadas 4 saídas digitais com ajuste por frequência, via PWM (Pulse-Width Modulation), com os pinos do Arduino D3, D5, D6 e D11, que representam os pinos PD3, PD5, PD6 e PB3, respectivamente do ATmega328P, que são as saídas que representam os 4 estados emocionais (divertido, entendido, relaxado e assustado) pelos leds, e que dependendo da frequência ficam mais ou menos claros, até totalmente claros ou totalmente escuros.

O ATmega328 é comumente usado em muitos projetos e sistemas autônomos onde é necessário um microcontrolador simples, de baixa potência e baixo custo. Talvez a implementação mais

comum deste chip seja na popular plataforma de desenvolvimento Arduino, ou seja, os modelos Arduino Uno, Arduino Pro Mini e Arduino Nano.

2.2 KNN

O K Nearest Neighbor ou simplesmente, KNN, é um algoritmo que pode ser usado tanto para classificação como regressão. Seu objetivo é determinar a qual grupo uma determinada amostra vai pertencer com base nas amostras vizinhas. Ele é um algoritmo simples e de fácil implementação, os exemplos de treinamento são armazenados e a previsão é feita somente quando um novo registro precisa ser classificado. O algoritmo do KNN é não paramétrico. Isso significa que sua funcionalidade está totalmente atrelada ao dataset, sem ter a necessidade da etapa de treinamento. Por um lado isso é positivo, porque não precisamos realizar etapas custosas de treinamento e o algoritmo vai aperfeiçoando sua precisão a medida que o dataset evolui. Por outro lado, negativo, não podemos destrelar a etapa da inferência da presença do dataset, tendo em vista que o algoritmo precisa sempre estar alimentado com os dados já obtidos.

O algoritmo KNN calcula a distância euclidiana entre o ponto de classificação e os dados do dataset. Com isso, é selecionado um número inteiro K de vizinhos. Esses vizinhos são os pontos onde o valor da distância euclidiana entre eles e o ponto retornou os menores valores entre o dataset. Após essa etapa, a classificação do ponto de inferência se dá pela dominância de uma classe entre seus vizinhos. Dessa forma, se hipoteticamente tivermos as classes A e B presentes como característica dos dados do dataset e o nosso número K for igual a 3, por exemplo, o ponto de inferência será classificado como "classe A" se dois ou três de seus vizinhos também tem "A" como classe. Dessa forma, podemos dizer que a classe do ponto de inferência se dá pela "votação" de seus vizinhos.

O KNN desenvolvido e programado recebe os dados como entrada os valores dos potenciômetros que representam os sensores de Eletrocardiograma (ECG), Volume de Sangue no Pulso (BVP), Condução Elétrica (GSR) e Respiração, para construção do dataset e como saída, temos os leds representando a intensidade dos sentimentos.

3 METODOLOGIA

Esse Trabalho é um teste de desempenho de tempo da KNN em seu trabalho de classificação onde foi simulado um problema onde dado sensores de monitoramento de batimentos cardíacos, volume de sangue, atividade eletrodérmica e respiração o algoritmo KNN diria a probabilidade para cada sentimento o usuário estaria sentindo: eufórico, entediado, relaxado ou com medo. Tanto as medidas de teste quando o conjunto de dados usado foi simulado. Para testar o funcionamento do KNN separado do hardware foi feito testes automatizados e para testar o algoritmo foi simulado um arduino uno no tinkercad. Informações de como executar os testes automatizados do KNN podem ser encontrados no repositório do projeto [1]. O KNN necessita de uma medida de distância para o seu funcionamento, por simplicidade foi assumindo que a união dos quatro sensores gera um espaço euclidiano R^4 e portanto foi utilizado a sua distancia, a distância euclidiana

$$\sqrt{\sum_{i=1}^4 (P_i - Q_i)^2} \quad (1)$$

onde Q e P são vetores do R^4 , formados pelas concatenação dos valores de batimentos cardíacos, volume de sangue, atividade eletrodérmica e respiração, esses valores foram todos normalizados para ficarem entre 0 e 1. O KNN retorna os K s elementos mais próximos do conjunto de dados utilizado, onde cada elemento possui um rótulo dizendo qual classe de sentimento esse elemento está, portanto uma entrada do algoritmo KNN, retorna K rótulos e para saber a probabilidade de cada sentimento dado uma entrada é anualizado a frequência dos sentimento dentre os K rótulos, por exemplo: se uma entrar retornou 3 rótulos de euforia, então a probabilidade de ser euforia é $\frac{3}{K}$, ou seja estamos considerando que esses sentimentos são independentes entre si, ou seja, a probabilidade de sentir euforia dado medo é zero. Para testar e ver o sistema funcionando foi feito uma simulação no tinkercad.

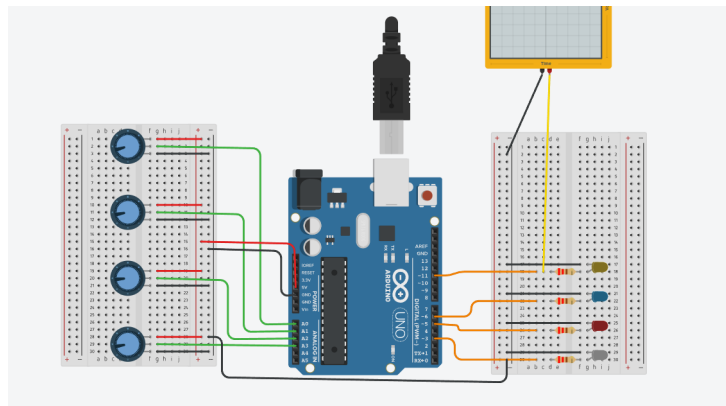


Figura 1: Montagem do circuito

Onde cada entrada do sensor é na verdade um potenciômetro, cuja a sua tensão varia de 0 até 5V, portanto para normalizar a entrada foi dividido o valor de cada potenciômetro pro cinco. Para visualizar as probabilidades de cada sentimento foi colocado um led para cada sentimento no circuito, onde cada led está ligado a um Pino do PWM e a probabilidade representa o duty cycle do PWM, ou seja, quando maior a probabilidade maior a intensidade do led. O sentimento com a sua respectiva

cor do led pode ser visto na tabela 1.

sentimento	cor do led
Eufórico	Branco
Entediado	Vermelho
Relaxado	Azul
Medo	Amarelo

Tabela 1: Sentimento e seu respectivo led

4 RESULTADOS

```
long int start_time = millis();  
find_k_nearest(kset, input, samples, K_NEAREST);  
uint8_t* label_frequencies = calculate_label_frequencies(samples);  
  
long int finish_time = millis();  
Serial.print("Prediction time: ");  
Serial.println(finish_time-start_time );
```

Figura 2: Parte do código que está sendo medido o tempo

Para medir o tempo de resposta do algoritmo KNN foi utilizado a função millis do Arduino entre o momento que as leituras já normalizadas entram no algoritmo até o momento em que as probabilidades de cada sentimento foi gerada. foi mensurado ao todo 25 medidas e gerada a tabela 2 a qual 29 milissegundos corresponde 64% dos valores e a média deu 28.96 milissegundos. O conjunto de dados utilizado nesse trabalho teve 10 amostras, então estimasse que o algoritmo gasta 2.896 milissegundos por amostra.

tempo de predição em milissegundos
29
29
30
28
29
29
28
30
29
29
29
29
28
29
30
28
30
29
29
28
29
29
29
29
29
29

Tabela 2: Captura do tempo de predição por meio da função millis

5 CONCLUSÃO

Com o decorrer do projeto, conseguimos desenvolver as diversas etapas práticas do uso do algoritmo do método do KNN - k Nearest Neighbor e como esses passos ajudam na obtenção dos dados para o cumprimento do objetivo do trabalho. Ao optar por um dataset baseado nas informações colhidas pelos sensores, nos proporcionou uma visão prática da serventia do conhecimento acadêmico para ajudar na resolução e auxílio de problemas de nossa sociedade que se baseiam em estados emocionais.

Por conseguinte, este projeto proporcionou, também, um estudo mais aprofundado acerca da utilização dos sistemas embarcados com a utilização do microcontrolador ATmega328P, em que fizemos o gerenciamento das portas de entrada (com base nos potenciômetros) que representavam os valores captados pelos sensores e a saída (representado pelos leds) e que representavam os estados emocionais de divertido, entendiado, relaxado e assustado, e ainda, os intensidade de iluminação do led, controlada pela variação da frequência, o que permite dizer se, está, por exemplo mais ou menos relaxado. Implementamos o algoritmo do método do KNN e com base no projeto de hardware mais software construído nós tivemos como retorno a classificação que procurávamos - estados emocionais, e como lidar com as tomadas de decisão após o diagnóstico. Por fim, os resultados adquiridos foram frutíferos para classificação dos estados emocionais de uma pessoa baseado nos sensores de monitoramento de batimentos cardíacos, volume de sangue, atividade eletrodérmica e respiração.

Referências

- [1] Repositório do projeto samuel-cavalcanti/embedded-knn-ufrn.
- [2] Algoritmo KNN para classificação. Disponível em: <<https://inferir.com.br/artigos/algoritmo-knn-para-classificacao>>
- [3] O algoritmo KNN em aprendizagem de máquina. Disponível em: <<https://portaldatascience.com/o-algoritmo-k-nearest-neighbors-knn-em-machine-learning>>
- [4] Introdução ao aprendizado de máquina. Disponível em: <<https://www.datageeks.com.br/aprendizado-de-maquina>>
- [5] O ATmega328p. Disponível em: <<https://www.microchip.com/en-us/product/ATmega328P>>
- [6] Datasheet ATmega328p. Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>
- [7] ATmega328P, the one microcontroller you should start with. Disponível em: <<https://www.seeedstudio.com/blog/2019/10/22/atmega328p-the-one-microcontroller-you-should-start-with>>