

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

**Márcio Manoel Sagaz**

**ESTUDO COMPARATIVO ENTRE MODELOS EM REDES NEURAIS E  
MODELO BAYESIANO PARA CONSCIÊNCIA SITUACIONAL**

**Florianópolis - SC**

**2021/1**



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

ESTUDO COMPARATIVO ENTRE MODELOS EM REDES NEURAIS E  
MODELO BAYESIANOS PARA CONSCIÊNCIA SITUACIONAL

Márcio Manoel Sagaz

Proposta de trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação pela Universidade Federal de Santa Catarina.

Orientador: Prof. Dr. Elder Rizzon Santos

Florianópolis - SC

2021/1



Márcio Manoel Sagaz

ESTUDO COMPARATIVO ENTRE MODELOS EM REDES NEURAIS E  
MODELOS BAYESIANOS PARA CONSCIÊNCIA SITUACIONAL

Este Trabalho de Conclusão de Curso foi julgado aprovado para  
a obtenção do Título de “Bacharel em Sistemas de Informação”.

Florianópolis, de 2021.

---

Prof. Dr. Cristian Koliver  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Dr. Elder Rizzon Santos  
Orientador

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Juliana Eyng

---

Dr. Thiago Ângelo Gelaim



À minha família.





## **AGRADECIMENTOS**

Agradeço ao nosso senhor Jesus Cristo pela minha existência e sabedoria, pois todos os dias tem guiado meu caminho, agradeço aos meus pais Manoel Nadir Sagaz e minha mãe Vera Lucia de Souza Sagaz que são responsáveis por guardar no meu coração o amor pela família, sendo eles seres humanos exemplares, ao qual busco me espelhar, agradeço ao meus irmãos Maycon Manoel Sagaz e Mauricio Manoel Sagaz, que tem torcido pelas minhas vitórias e conquistas. Agradeço à minha amada esposa Cláudia Regina Albano da qual tenho a felicidade enorme de dividir minha vida e que é uma incentivadora das minhas conquistas. Agradeço ao Prof. Dr. Elder Rizzon Santos pela paciência e ensinamento.

Obrigado a todos os amigos e professores pelas experiências trocadas e pelos aprendizados que provém de suas experiências de vida. Obrigado a todos vocês.



## RESUMO

Rede neural artificial é um modelo de processamento computacional baseado no modelo de funcionamento do neurônio biológico, essas redes têm a capacidade de aprendizagem através de exemplos e de generalização desse aprendizado. Elas são formadas por conjuntos de neurônios interligados, onde cada elemento chamado neurônio é definido matematicamente como sendo um conjunto de entradas que são multiplicadas por pesos para cada entrada, estas são integradas através de um somador que tem seu resultado enviado para a função de ativação ou função de transferência. A interligação desses neurônios artificiais propicia uma rede com processamento paralelo e não linear. Essas redes têm diversas aplicações como: detecção e reconhecimento de padrões, robótica, análise de imagens, classificação de dados, processamento de sinais. O uso de dispositivos móveis por pedestres em vias públicas tem ocasionado um elevado número de acidentes. O motivo é que o dispositivo causa ao pedestre uma falta de consciência sobre o tráfego urbano ao seu redor. Em uma tentativa de reduzir a problemática exposta relacionada aos dispositivos móveis, foi proposto no projeto “*Road Awareness*” um modelo de consciência situacional com usuários usando smartphones em vias urbanas. Somando-se a este esforço o trabalho de TCC “Aplicação de Aprendizagem por Reforço para um Modelo Bayesiano de Consciência” programou e aplicou um algoritmo de aprendizagem por reforço na aprendizagem dos parâmetros do modelo em questão, a rede Bayesiana. Com o objetivo de contribuir com o projeto de consciência situacional, este trabalho apresenta um modelo proposto em redes neurais artificiais para avaliar se este método comparado ao de redes bayesiana poderá melhorar a consciência situacional.

**Palavras-chave:** Aprendizagem de Máquina, Consciência Situacional, Redes Neurais Artificiais.



## **ABSTRACT**

Artificial neural networks are a model of computational processing based on the model of biological neuron functioning. These networks can learn through examples and generalization of this learning. They are formed by sets of interconnected neurons, where each element called neuron is defined mathematically as having a set of inputs that are multiplied by weights for each input; these are integrated through an adder that has its result sent to the activation function or transfer function. The interconnection of these artificial neurons provides a network with parallel and non-linear processing. These networks have several applications such as: pattern detection and recognition, robotics, image analysis, data classification, signal processing. The use of mobile devices by pedestrians on public roads has caused a high number of accidents. The reason is that the device causes the pedestrian a lack of awareness about the urban traffic around them. In an attempt to reduce the exposure problematic related to mobile devices, a model of situational awareness with users using smartphones on urban roads was proposed in the "Road Awareness" project. In addition to this effort, the TCC study “Aplicação de Aprendizagem por Reforço para um Modelo Bayesiano de Consciência” programmed and applied a learning algorithm for reinforcement in learning the parameters of the model in question, the Bayesian network. In order to contribute to the situational awareness project this work will implement a proposed model in artificial neural networks to evaluate if this method compared to Bayesian Networks can improve the situational awareness.

**Keywords:** Artificial Neural Networks, Machine Learning, Situational Awareness.



## LISTA DE FIGURAS

Figura 1 - Exemplo estrutural de um simples de neurônio.....	25
Figura 2 - Exemplo estrutural rede perceptrons multicamadas.....	26
Figura 3 – Estrutura de uma rede RBF.....	28
Figura 4 – Camadas de uma RNP.....	29
Figura 5 – Diagrama de fluxo do desenvolvimento.....	38
Figura 6 – Imagem do ambiente Octave - (Fonte: salford).....	40
Figura 7 – Usuário imerso no ambiente da simulação - (Fonte: PEREIRA, 2017).....	41
Figura 8 – Possíveis trajetórias alcançadas pelos carros - (Fonte: GELAIM, 2019).....	42
Figura 9 – Pedestre no ambiente de simulação - (Fonte: GELAIM, 2019).....	42
Figura 10 – Usuário imerso no ambiente da simulação - (Fonte: PEREIRA, 2017).....	45
Figura 11 – Modelo de RB usada na validação de CS - (Fonte: PEREIRA, 2017).....	47
Figura 12 – Proporção de em % para cada modelo da percepção útil.....	51
Figura 13 – Proporção de em % para cada modelo da percepção útil.....	52
Figura 14 – Ilustração de um evento do experimento.....	54
Figura 15 – Ilustração de um evento do experimento.....	56
Figura 16 - Gráfico de Risco para cada carro.....	59
Figura 17 – Tipos de consciência levantados na base de dados.....	60
Figura 18 – Estrutura da rede RBF treinada no trabalho.....	65
Figura 19 – Fator de alisamento para diferentes valores de desvios padrões.....	70
Figura 20 – Treino 1: desempenho dos dados teste X treino.....	72
Figura 21 – Treino 2: desempenho dos dados teste X treino.....	72
Figura 22 – Treino 3: desempenho dos dados teste X treino.....	73
Figura 23 – Fator de alisamento para diferentes valores de desvios padrões.....	74
Figura 24 – Fator de alisamento para diferentes valores de desvios padrões.....	77
Figura 25 – Fator de alisamento para diferentes valores de desvios padrões.....	77
Figura 26 – ROC/AUC para o algoritmo da rede RBF.....	80
Figura 27 – ROC/AUC para o algoritmo RPN.....	80
Figura 28 – ROC/AUC para o algoritmo RB.....	81

## LISTA DE TABELAS

Tabela 1 – Levantamento de trabalhos relacionados.....	37
Tabela 2 – Média de tempo de Consciente - (Fonte: PEREIRA, 2017).....	46
Tabela 3 – Modelo de percepção consciência x tempo de movimento.....	51
Tabela 4 – O risco é dado pela razão do atropelamento por evento em %.....	56
Tabela 5 – Mapeamento Simulation Type para Distração.....	57
Tabela 6 - A média de atropelamento do carro e o desvio padrão.....	58
Tabela 7 - Classificação de risco.....	59
Tabela 8 – Fragmento da base de dados.....	61
Tabela 9 – Fragmento da base com novas definições.....	61
Tabela 10 – Valores para Consciência.....	62
Tabela 11 – Valores para Percepção Útil.....	62
Tabela 12 – Valores para a Distração.....	62
Tabela 13 – Valores para Direção Carro.....	63
Tabela 14 – Valores para Chance Percepção.....	63
Tabela 15 – Valores para Risco.....	63
Tabela 16 – Valores IDs em sequência experimental.....	66
Tabela 17 – Treino 1: valores IDs em sequência aleatória.....	66
Tabela 18 – Treino 2: valores de forma binarizada.....	67
Tabela 19 – Treino 3: diferenciando classe por pesos.....	68
Tabela 20 – Pesos de categorização.....	69
Tabela 21 – Dados de treinamento para RNP.....	71
Tabela 22 – Matriz Confusão rede RBF.....	74
Tabela 23 – Matriz Confusão RNP.....	75
Tabela 24 – Dados de entrada Rede Bayesiana.....	76
Tabela 25 – Exemplo de saída para uma entrada na RB.....	78
Tabela 26 – Matriz Confusão RB.....	79



## LISTA DE ABREVIATURAS E SIGLAS

ANN	<i>Artificial Neural Network</i> .....	24
MLP	<i>Multilayer Perceptron</i> .....	26
RBF	Função de Base Radial.....	27
RNP	Rede Neural Probabilística.....	28
CS	Consciência Situacional .....	29
FOREX	<i>Foreign Exchange</i> .....	31
BPNN	<i>Back Propagation Neural Network</i> .....	31
FLANN	<i>Functional Link Artificial Neural Network</i> .....	31
RBFNN	<i>Radial Basis Function Neural Network</i> .....	31
RNN	<i>Recurrent Neural Network</i> .....	31
FBNN	<i>Feed Forward Back Propagation Neural Network</i> .....	31
LM	<i>Levenberg-Marquardt</i> .....	31
MLANN	<i>Multilayer ANN</i> .....	32
CFLANN	<i>Cascade FLANN</i> .....	32
GRNN	<i>Generalized Regression Neural Network</i> .....	32
FFNN	<i>FeedForward Neural Network</i> .....	32
RBM	<i>Restricted Boltzmann machines</i> .....	33
DBN	<i>Deep Belief Networks</i> .....	33
AE	<i>AutoEncoder</i> .....	33
CNN	<i>Convolutional Neural Network</i> .....	33
PLN	Processamento de Linguagem Natural.....	34
ENN	<i>Ensemble Neural Network</i> .....	35
BNN	Redes Neurais Bayesianas .....	36
RB	Redes Bayesiana .....	37
AGNN	<i>Attention Guided Neural Network</i> .....	36
PC	Probabilidade condicional.....	44



## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>21</b>
1.1 OBJETIVOS.....	21
1.1.1 Objetivo Geral.....	21
1.1.2 Objetivos Específicos.....	22
1.2 METODOLOGIA.....	22
1.3 ESTRUTURA DO TRABALHO.....	22
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>24</b>
2.1 APRENDIZADO DE MÁQUINA.....	24
2.2 REDES NEURAIS ARTIFICIAIS.....	24
2.3 PERCEPTRON MULTICAMADAS.....	26
2.4 REDES NEURAIS DE FUNÇÃO DE BASE RADIAL.....	27
2.5 REDE NEURAL PROBABILÍSTICA.....	28
2.6 CONSCIÊNCIA SITUACIONAL.....	29
<b>3 TRABALHOS RELACIONADOS.....</b>	<b>31</b>
3.1 PREVISÃO DE TAXA DE CÂMBIO.....	31
3.2 A ARQUITETURA DE 4 REDES NEURAIS PROFUNDAS.....	32
3.3 ANÁLISE PARA CONSCIÊNCIA SITUACIONAL.....	34
3.4 OUTROS TRABALHOS.....	35
3.5 COMPARAÇÃO ENTRE OS TRABALHOS.....	36
<b>4 DESENVOLVIMENTO.....</b>	<b>38</b>
4.1 IDENTIFICAÇÃO DOS DADOS.....	39
4.1.1 O local do ambiente de realidade virtual.....	39
4.1.2 Salvando Lemmings no experimento 1.....	40
4.1.3 Desviando de Carros no experimento 2.....	41
4.2 ENTENDIMENTO DO MODELO BAYESIANO.....	44
4.2.1 A motivação para produção do modelo.....	44
4.2.2 Os dados usados no modelo.....	44
4.2.3 A nova base de dados e as definições.....	47
4.2.4 Uso neste trabalho.....	49
4.3 ANÁLISE DESCRITIVA DE DADOS.....	49
4.3.1 Preparando a percepção útil.....	49
4.3.2 Preparando a chance de percepção.....	52
4.3.3 Preparando a direção do carro.....	53
4.3.4 Preparando a distração.....	57
4.3.5 Preparando o risco.....	57
4.3.6 Preparando a consciência.....	59

4.3.7 Base de dados preparada.....	61
4.4 DESCRIÇÃO DAS REDES NEURAIIS.....	64
4.4.1 Escolha da Rede Função de Base Radial.....	64
4.4.2 Treinamentos da Rede de Função de Base Radial.....	64
4.4.3 Escolha da Rede Neural Probabilística.....	69
4.4.4 Treinamentos da Rede Neural Probabilística.....	69
4.5 RESULTADOS DAS REDES NEURAIIS.....	71
4.5.1 Resultados da rede RBF.....	71
4.5.2 Resultados da RNP.....	74
4.6 ANÁLISE DO MODELO BAYESIANO.....	75
4.6.1 Escolha da Rede Bayesiana.....	75
4.6.1 Treinamento da Rede Bayesiana.....	76
4.7 RESULTADO DA REDE BAYESIANA.....	78
4.7.1 Resultados da RB.....	78
4.8 ANÁLISE COMPARATIVA DOS MODELOS.....	79
<b>5 CONCLUSÃO E TRABALHOS FUTUROS.....</b>	<b>82</b>
5.1 TRABALHOS FUTUROS.....	83
<b>REFERÊNCIA.....</b>	<b>84</b>
<b>6 ANEXOS.....</b>	<b>86</b>
6.1 ANEXO A - CÓDIGO FONTE REDE RBF.....	86
6.2 ANEXO B - CÓDIGO FONTE REDE RNP.....	90
6.3 ANEXO C - CÓDIGO FONTE REDE RB.....	93
6.4 ANEXO D - ARTIGO.....	95

## 1 INTRODUÇÃO

A inteligência artificial envolve o desenvolvimento de paradigmas ou algoritmos para que as máquinas realizem tarefas cognitivas semelhantes às como os seres humanos são capazes de realizar (HAYKIN, 2001). Isto está relacionado ao processo de aprendizagem de máquina que pode apresentar-se em quatro abordagens, sendo estas a abordagem simbólica, a abordagem conexionista, a abordagem genética (evolucionária) e a dinâmica (estocástica) (LUGER, 2013). Neste trabalho será desenvolvida uma abordagem conexionista usando-se redes neurais. A base de uma rede neural é o neurônio que é caracterizado por sinais de entrada, um conjunto de pesos de valor real, um nível de ativação e uma função limiar (LUGER, 2013).

Uma rede neural artificial em sua característica é definida segundo a topologia, o algoritmo de aprendizagem e o esquema de codificação (LUGER, 2013). Entre as topologias mais comuns estão a feedforward, a realimentada e as reticuladas com camadas simples e múltiplas (SILVA; SPATTI; FLAUZINO, 2010), este trabalho inicialmente não adotará uma topologia específica, pois sua escolha fará parte do estudo deste trabalho. Sobre os algoritmos de aprendizagem encontramos os métodos de treinamento supervisionado, não supervisionado e por reforço, assim como aprendizagem usando um lote de padrões e padrão a padrão (SILVA; SPATTI; FLAUZINO, 2010). Neste trabalho o método de aprendizagem também será estudado, avaliado e definido para aplicar a comparação com o algoritmo de redes bayesianas do trabalho de conclusão de curso (TCC) intitulado “Aplicação de Aprendizagem por Reforço para um Modelo Bayesiano de Consciência”. Em termos de codificação serão usados os mesmos dados utilizados na pesquisa “Road Awareness” para uma adequada aplicação no estudo de consciência situacional.

O uso da abordagem de rede neural artificial neste trabalho de conclusão de curso tem como objetivo auxiliar no desenvolvimento e experimentação de um modelo de estudo de consciência situacional. A construção deste modelo considera ainda uma avaliação em comparação à abordagem bayesiana, já desenvolvida em trabalhos anteriores.

### 1.1 OBJETIVOS

#### 1.1.1 Objetivo Geral

Este trabalho de conclusão de curso tem como objetivo definir um modelo conexionista para consciência situacional.

### **1.1.2 Objetivos Específicos**

1. Descrever as características das redes neurais artificiais;
2. Definir os critérios de avaliação do modelo de consciência situacional;
3. Analisar modelos de redes neurais adequados ao contexto deste trabalho;
4. Comparar os resultados das redes neurais juntamente com modelos já definidos;
5. Implementar os algoritmos com os modelos propostos para realização do item 4;
6. Preparar, normalizar, discretizar e aplicar os dados nos modelos de rede neurais implementados no item 5;
7. Testar e comparar um modelo de rede neural contra um modelo de rede bayesiana implementado 5.

## **1.2 METODOLOGIA**

O método de pesquisa utilizado é o exploratório, pois o objetivo é que através de estudos na literatura existente se desenvolva no mínimo dois modelos de algoritmos de consciência situacional utilizando redes neurais artificiais. O método de pesquisa também se caracteriza por ser quantitativo, onde os dados utilizados para treinamento e a validação são os mesmos da base de dados que são fornecidos de um projeto de pesquisa já realizado "Road Awareness". Serão treinados esses modelos de redes neurais artificiais criados com dados da pesquisa e após a aferição dos resultados o modelo com a rede neural com melhor desempenho será comparada também com a rede bayesiana produzida a partir do TCC citado. Ao final ter-se-á o resultado dessa comparação de modelos sobre a consciência situacional.

## **1.3 ESTRUTURA DO TRABALHO**

O trabalho é dividido em 5 capítulos. O primeiro aborda a introdução, os objetivos, a metodologia de pesquisa e a estrutura deste trabalho. No capítulo seguinte é abordado a fundamentação teórica realizada, focada na explanação sobre redes neurais. O capítulo três apresenta os trabalhos relacionados, que apresentam questões próximas ao desenvolvimento deste trabalho, suas características e abordagens. O quarto capítulo apresenta como é a

estrutura de dados utilizada, as modificações e adaptações aplicadas, as estruturas de redes aplicadas, bem como a avaliação e seleção da rede. Neste capítulo também será apresentado a análise com o método de redes bayesianas e as análises para as redes neurais escolhidas. Por fim, o capítulo cinco apresenta a conclusão e os possíveis trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 APRENDIZADO DE MÁQUINA

Aprendizagem de máquina é uma área da Inteligência Artificial onde a máquina tenta aprender a partir de um conjunto de dados de treinamento, buscando generalizar os padrões observados (COPPIN, 2013). Os dados de treinamento contém padrões da qual a máquina busca adquirir uma base de conhecimento sobre esses dados, que deve ser medido por algum fator de desempenho (HAYKIN, 2001).

A propriedade que é de importância primordial para uma rede neural é a sua habilidade de aprender a partir de seu ambiente e de melhorar o seu desempenho através da aprendizagem (HAYKIN, 2001).

Os dois métodos clássicos de aprendizado de máquina são o supervisionado e o não supervisionado. Na aprendizagem supervisionada, um conjunto de dados rotulados de treinamento são usados para estimar e mapear os dados da entrada para a saída desejada. Na aprendizagem não supervisionada, nenhum exemplo rotulado é fornecido e não há noção da saída durante o processo de aprendizagem (KOUROU, et al, 2015), sendo que o modelo tenta agrupar os dados, em alguma métrica como, por exemplo, a similaridade entre os dados.

### 2.2 REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais ou *Artificial Neural Network* (ANN), são métodos conexionistas, que são estruturas com forte capacidade de generalizar, ou seja, é capaz de reconhecer elementos de entrada que não estão no conjunto de treinamento (HAYKIN, 2001). As ANNs são modeladas em relação ao cérebro humano e são compostas por vários neurônios artificiais (COPPIN, 2013). Na figura 1 podemos observar o modelo de um neurônio artificial. Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural (HAYKIN, 2001).

Os elementos de composição de um neurônio são as entradas representadas na figura 1 pelas variáveis  $x_0$ ,  $x_1$  e  $x_i$ , onde  $x_0$  representa o viés (bias) que é um elemento de entrada normalmente utilizado no efeito de aumentar ou diminuir a entrada líquida da função de ativação (HAYKIN, 2001). Ou serve para deslocar a função de limiar sobre o eixo horizontal (LUGER, 2013). Os pesos sinápticos são representados por  $w_0$ ,  $w_1$  e  $w_i$  e tem como



propriedade básica aumentar ou diminuir a intensidade do sinal de entrada  $x_i$ , note que o peso  $w_0$  relacionado ao viés é igual a 1. O  $\Sigma$  representa uma soma ponderada dos sinais de entradas em relação ao seu peso, também pode ser chamado de nível de ativação (LUGER, 2013), onde cada sinal de entrada é escalado pelo peso de conexão  $w_i$  ao longo da linha de entrada. Outro elemento da composição de um neurônio artificial é a função de ativação  $\varphi$ , ela recebe o nível de ativação de (LUGER, 2013) proveniente do somador para restringir a amplitude da saída de um neurônio (HAYKIN, 2001). Cabe ressaltar que na literatura o termo ‘nível de ativação’ para Coppin (2013) significa o sinal de saída do neurônio após a função de ativação.

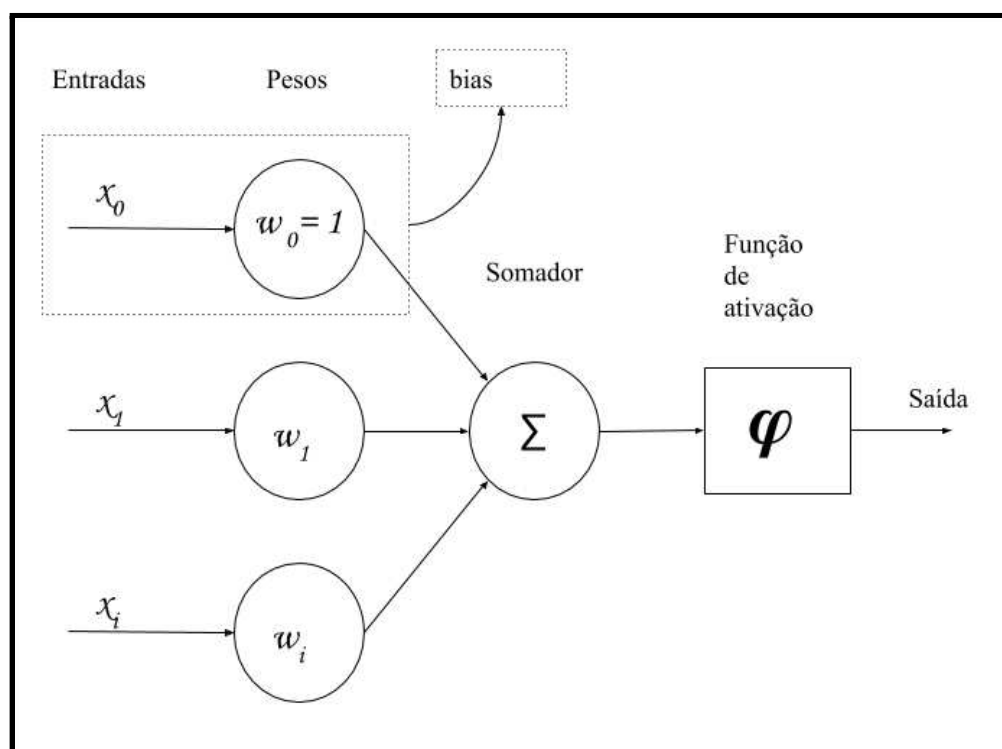


Figura 1 - Exemplo estrutural de um simples de neurônio

A união dos vários neurônios pode realizar uma rede neuronal inspirada no modelo neural do cérebro animal, olhando de uma perspectiva matemática, uma ANN pode modelar qualquer função com base em um conjunto de funções básicas (CHEN, CHANG, 2019).

Na literatura atual há diversos modelos de ANNs, desde modelos mais clássicos com *perceptron*, até modelos mais avançados como Rede Neural de Máquina de Turing, Rede Neural Convolutacional, Rede Adversarial Generativa entre outras.

### 2.3 PERCEPTRON MULTICAMADAS

O perceptron é a forma mais simples de uma rede neural, basicamente é um neurônio descrito no tópico 2.2, usado para a classificação de padrões linearmente separáveis (HAYKIN, 2001). A rede perceptrons de múltiplas camadas ou *multilayer perceptron* (MLP) é uma ligação de neurônios classificados como perceptrons, onde existe um conjunto de neurônios que formam a camada de entrada, a camada oculta, que é formada por uma ou mais camadas, e uma camada de saída.

Para entender o modelo básico de treinamento de uma rede neural, toma-se classicamente a MLP como exemplo, um algoritmo comum usado para o treinamento supervisionado da MLP é alimentada adiante (em inglês *feedforward*) é a retropropagação do erro (em inglês *error back-propagation*) que busca diminuir o erro na saída da rede. A figura 2 ilustra uma rede MLP *feedforward*.

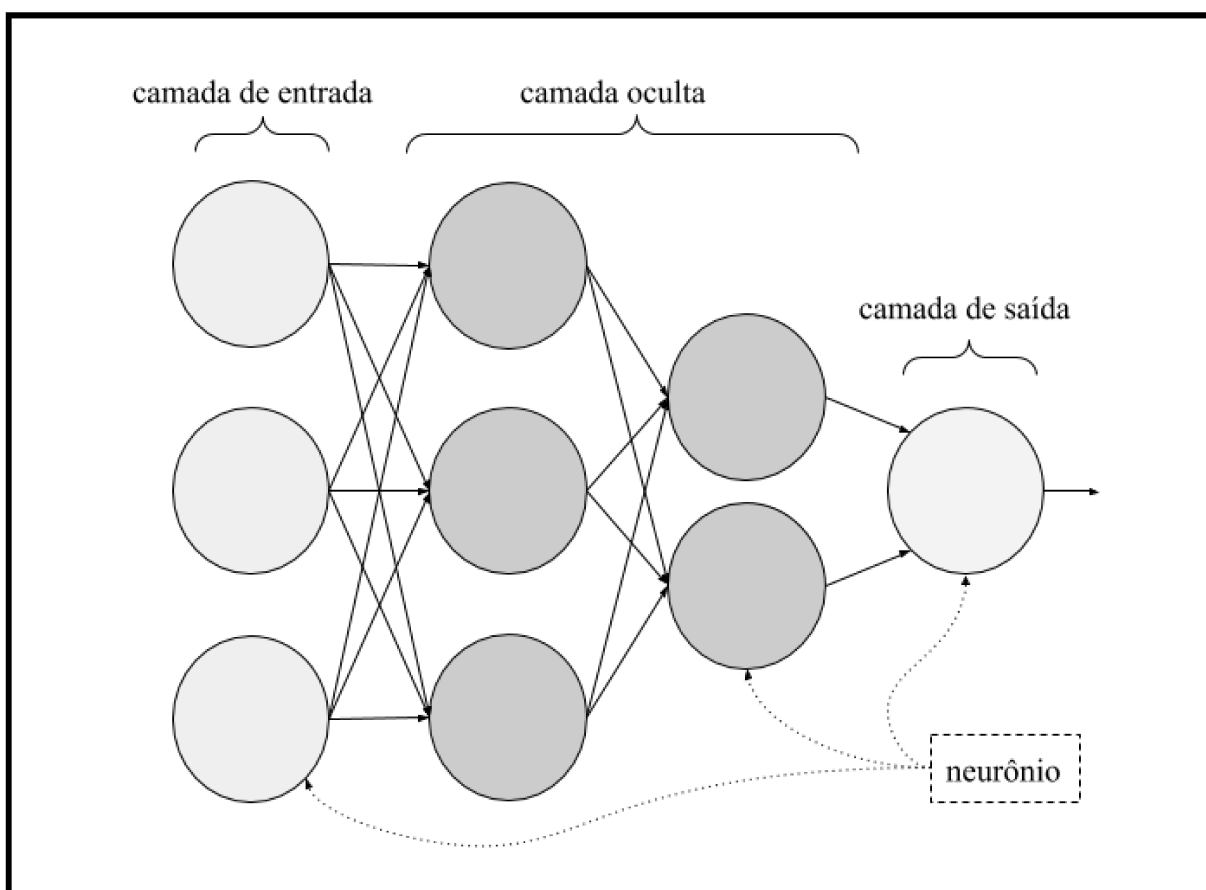


Figura 2 - Exemplo estrutural rede perceptrons multicamadas

Assim, buscando uma diminuição do erro que aparece na saída da rede, faz-se um ajuste dos pesos baseados neste mesmo erro de saída. O erro encontrado na saída é usado para

ajustar todos os pesos pelo algoritmo de retropropagação. O objetivo é, que através de uma função de custo que leve o erro de saída em consideração, seja ajustado os parâmetros livres da rede, que são os pesos sinápticos, usando o método matemático de gradiente descendente.

O método de gradiente descendente calcula o gradiente da função de custo com relação a cada peso pela regra da cadeia, ele reduz o gradiente de erro até que um mínimo no espaço de erro seja encontrado (COPPIN, 2013). Assim, ao final deste processo do ajuste dos pesos, a MLP foi treinada.

## 2.4 REDES NEURAIIS DE FUNÇÃO DE BASE RADIAL

Uma função de base radial (RBF) é uma função que depende da distância a um ponto específico. Uma função como a função gaussiana é uma função radial, pois cada ponto de interesse na função  $f(x)$  é uma distância entre o ponto  $f(x)$  e o centro da gaussiana. Sendo assim, qualquer função onde o ponto dado por  $f(x)$  dependa da distância a um ponto específico pode ser chamada de função de base radial.

Dado um conjunto de  $N$  pontos diferentes onde  $\{x_i \in \mathfrak{R}^{m_0} | i = 1, 2, \dots, N\}$  e o conjunto de  $N$  números reais  $\{d_i \in \mathfrak{R}^1 | i = 1, 2, \dots, N\}$ , procura-se uma função que satisfaça a interpolação onde  $F: \mathfrak{R}^N \rightarrow \mathfrak{R}^1$  (HAYKIN, 2001). A técnica de RBF usa a seguinte forma para encontrar essa função  $F(x) = \sum_{i=1}^N \omega_i \varphi(\|x - x_i\|)$ , onde  $\varphi(\|x - x_i\|) | i = 1, 2, \dots, N\}$  é um conjunto de **funções radiais**,  $\|\cdot\|$  é a **norma** euclidiana geralmente e  $x_i \in \mathfrak{R}^{m_0} | i = 1, 2, \dots, N$  são os **centros** das funções de base radial (HAYKIN, 2001).

A figura 3 ilustra a estrutura de uma rede RBF mostrando a camada de entrada, a camada de saída e a camada oculta. Entre a camada de entrada e a camada oculta não há pesos sinápticos como em um clássica MLP, isso ocorre devido a sua característica de não linearidade. Mas entre a camada oculta e a camada de saída temos pesos sinápticos que são treinados.

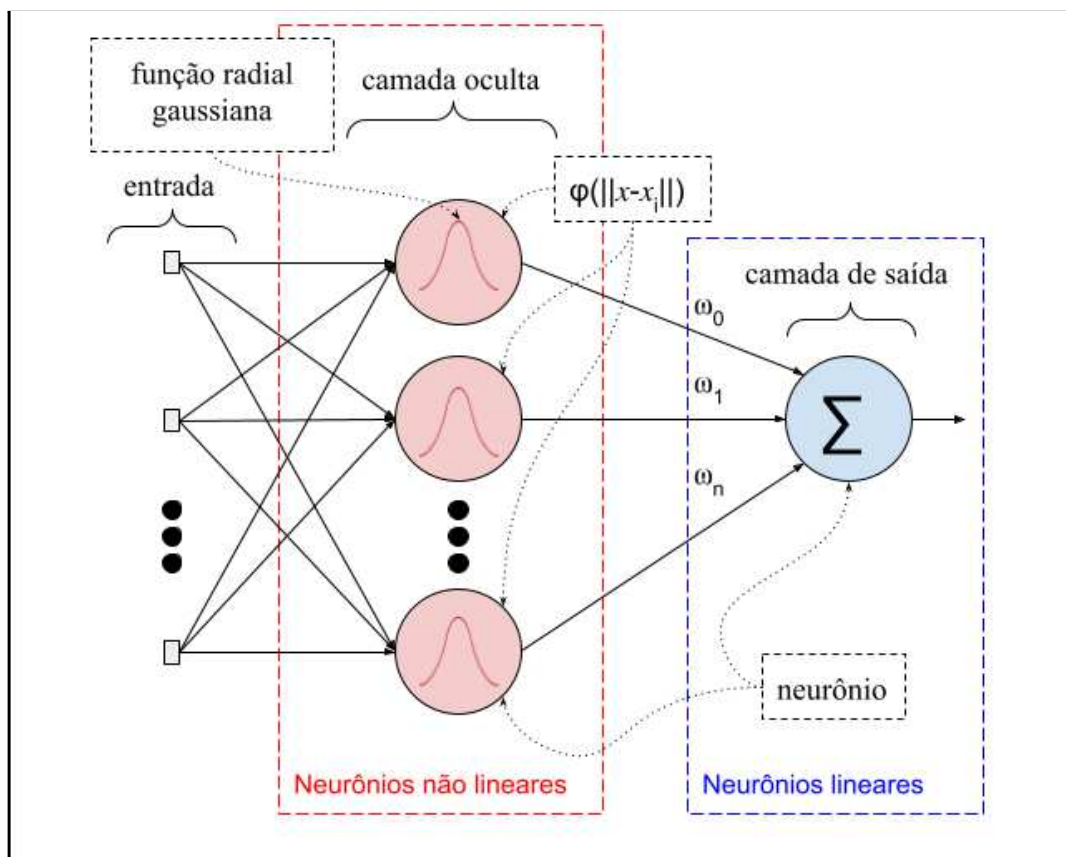


Figura 3 – Estrutura de uma rede RBF.

## 2.5 REDE NEURAL PROBABILÍSTICA

Uma rede neural probabilística (RNP) é uma implementação de um algoritmo estatístico chamado análise discriminante de kernel em que as operações são organizadas em uma rede feedforward multicamadas (EMARY, SRINIVASAN, 2008). No algoritmo RNP, uma janela de Parzen é usada para aproximação não paramétrica da função de distribuição de probabilidade (FDP) de cada população e a regra de Bayes é então empregada para alocar a classe com maior probabilidade posterior para novos dados de entrada (ZEINALI, STORY, 2017). Assim RNP está intimamente relacionada à janela de Parzen e consiste em várias sub-redes, cada uma das quais é um estimador para a FDP de Parzen para cada classe.

Uma RNP pode ser dividida em 4 fases (camadas). Na primeira fase (camada 1) considera-se os nós de entrada como sendo os conjuntos de medidas. A segunda fase (camada 2) consiste nas funções gaussianas usando um determinado conjunto de pontos de dados como centros. A terceira fase (camada 3) executa uma operação de soma das saídas da segunda camada para cada classe. A quarta fase (camada 4) é realizada uma ponderação para

selecionar o maior valor, e por fim o rótulo da classe é então determinado. A figura 4 exemplifica as fases da RNP.

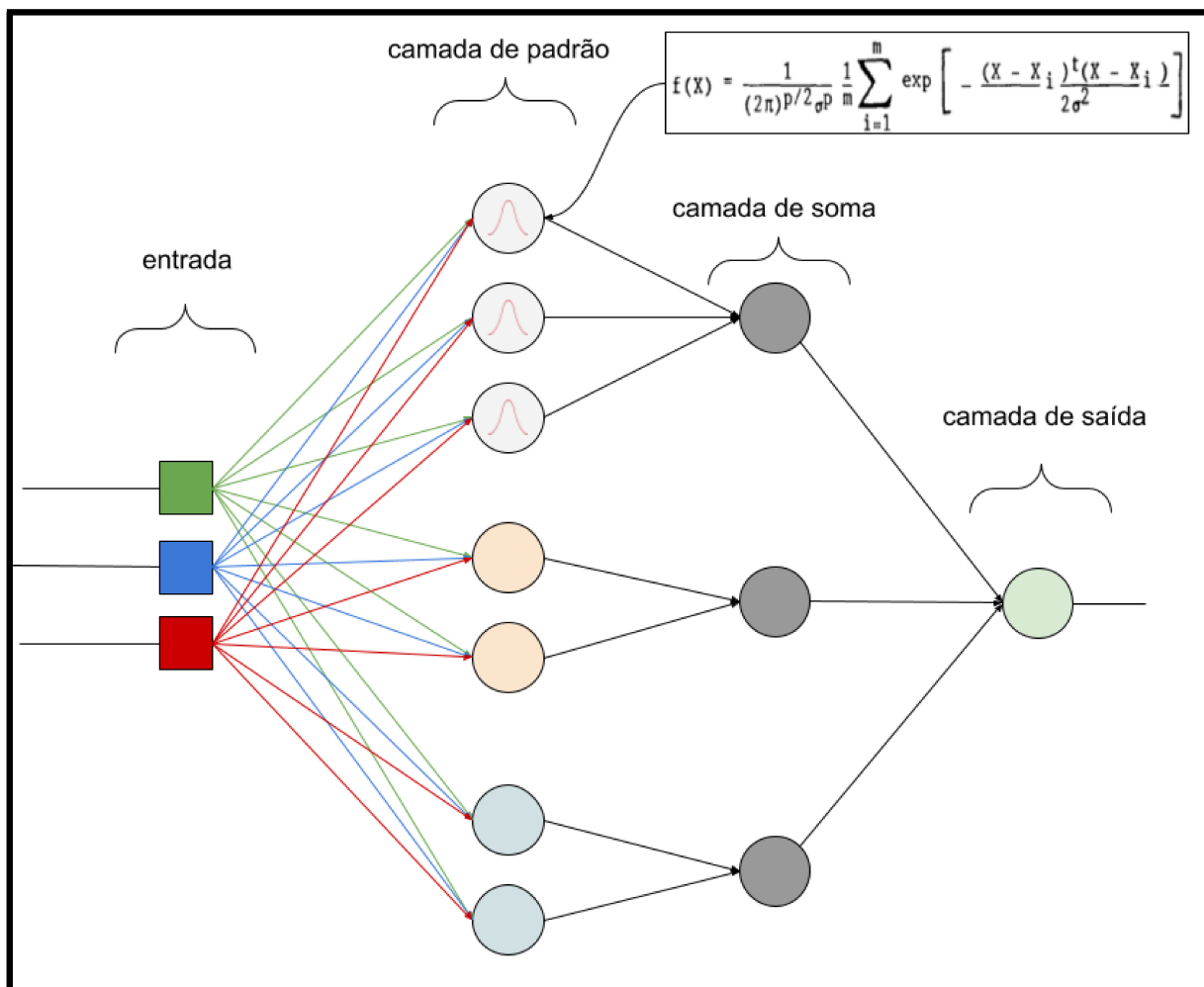


Figura 4 – Camadas de uma RNP.

## 2.6 CONSCIÊNCIA SITUACIONAL

A consciência situacional (CS) em ambientes dinâmicos é provocada por um processo que leva a um estado de conhecimento, ou seja, a CS é esse estado de conhecimento que é provocado por um processo e para alcançar tal estado o processo divide-se em três fases hierárquicas: a **percepção** dos elementos na situação atual, a **compreensão** da situação atual e

a **projeção** de um estado futuro próximo (ENDSLEY, 1995). Esse processo é tratado como avaliação situacional (ENDSLEY, 1995).

Para um cenário de trânsito os elementos de percepção são os carros, pedestres, sons, luzes de trânsito, pessoas, animais, localização, etc, ou seja, tudo que envolve um ambiente de trânsito. O processo de compreensão parte dos elementos percebidos em um ambiente para formar uma padronização e dar significado semântico para os elementos desconexos no nível de percepção (ENDSLEY, 1995). Para este nível do processo, um sinal vermelho conjuntamente com um barulho de motor de um carro em alta velocidade e com sirene deve representar algo para seu objetivo específico de atravessar uma via. Neste caso, um indivíduo sem a compreensão das prioridades de uma ambulância não poderia ter uma compreensão adequada da situação de atravessar a via naquele momento. É sobre os dois níveis de processo de CS descritos acima que o terceiro nível se realiza, neste nível o indivíduo pode projetar as ações futuras para conhecer os estados dinâmicos e estáticos dos elementos e compreender a situação apresentada (ENDSLEY, 1995). Ao alcance deste último nível o pedestre tem conhecimento suficiente para uma tomada de decisão, sabendo que o carro não vai parar no semáforo, pois tem prioridade sobre sua passagem na faixa.

A CS é observada como parte de um sistema que envolve a tomada de decisão e a ação realizada no ambiente (ENDSLEY, 1995). Portanto, a CS é uma entrada para a tomada de decisão que, conseqüentemente, é a entrada de uma ação que afetará o ambiente observado. Além disso, a CS, a decisão e a ação, são influenciados por processos externos.

São basicamente dois conjuntos de fatores externos ao sistema que afetam a CS, a decisão e a ação (ENDSLEY, 1995). O primeiro fator é o individual, que engloba os objetivos pessoais, as experiências, os conhecimentos, os pré conceitos, as habilidades e o treinamento. O segundo fator é o ambiente e as tarefas, que incluem carga de trabalho, complexidade, design do sistema e estressores. Sendo assim, esses dois conjuntos de fatores externos influenciam na CS (ENDSLEY, 1995).

### 3 TRABALHOS RELACIONADOS

Nesta etapa será abordado trabalhos correlatos a área de interesse deste trabalho. No primeiro trabalho é apontada uma pesquisa *survey* onde são descritos o uso de alguns modelos de redes neurais para a previsão da taxa *foreign exchange* (FOREX). O segundo trabalho é uma pesquisa também *survey* que aponta para 4 arquiteturas de deep learning com redes neurais. O terceiro trabalho descreve a aplicação e a comparação de técnicas que utilizam análise de dados associadas a conceitos como o aprendizado de máquina para exploração de conhecimento em uma pesquisa sobre consciência situacional. Nos outros trabalhos vemos algumas pesquisas na área de ANN.

#### 3.1 PREVISÃO DE TAXA DE CÂMBIO

Modelos de predição de taxa de câmbio têm sido pesquisados usando-se ANNs. Para a construção destes modelos (PARAMITA et al, 2018) traz algumas abordagens sobre esse tema. A taxa FOREX refere-se à taxa na qual a moeda de um país é trocada pela moeda de outro país. Existem muitos fatores correlacionados que afetam as taxas FOREX, como fatores micro e macroeconômicos relacionados, inflação, taxa de juros, custo de fabricação, dívida, política e psicologia, por causa da complexidade, ruído e volatilidade, a ANN é amplamente usada para previsão (PARAMITA et al, 2018). Sendo assim, redes como *Back Propagation Neural Network* - (BPNN), *Functional Link Artificial Neural Network* (FLANN), *Radial Basis Function Neural Network* (RBFNN), *Recurrent Neural Network* (RNN) foram usadas na previsão (PARAMITA et al, 2018).

Em relação a BPNN, (PARAMITA et al, 2018) aponta um trabalho que desenvolveu um sistema de negociação automatizado para a previsão da taxa de câmbio com uma única rede de *feedforward* de camada oculta. Outro trabalho examinou os dados de um período de 12 anos para prever a inflação da Índia usando *feed forward back propagation neural network* (FBNN). Em outro trabalho usou-se o modelo FBNN com abordagem gradiente descendente usando o algoritmo *Levenberg-Marquardt* (LM) para prever a taxa de câmbio diária. Também em outros modelos usou-se a técnica BPNN para aumentar a precisão de predição de diferentes taxas de câmbio de moedas e descobriram que os modelos apresentam melhor desempenho quando treinados com algoritmos de retropropagação (PARAMITA et al, 2018).

O trabalho de (PARAMITA et al, 2018) aponta o uso de FLANN em substituição às *multilayer ANN* (MLANN) devido ao custo computacional desta última. O uso do modelo FLANN pode ser amplamente visto na previsão do mercado financeiro. Em uma pesquisa, apontada por (PARAMITA et al, 2018), foram desenvolvidos dois modelos adaptativos não lineares, FLANN e *cascade FLANN* (CFLANN), para prever a taxa de câmbio entre o USD e três outras moedas. Em outro, aponta, o uso da técnica baseada em FLANN para prever o valor patrimonial líquido, num modelo de predição eficiente que envolve menos computação que MLANN.

O RBFNN é eficaz na previsão de séries temporais financeiras. O RBFNN tem a capacidade de detectar as relações não lineares complexas entre as variáveis dependentes e independentes (PARAMITA et al, 2018). Há trabalhos, descreve (PARAMITA et al, 2018), em que a camada oculta em RBFNN difere de outras redes neurais, onde cada nó representa um cluster de dados com um raio pré definido e cada nó mede a distância do vetor de entrada ao seu centro. Outros autores usaram *Generalized Regression Neural Networks* (GRNNs) que é um tipo de RBFNN. Em outro, RBFNN sai consideravelmente melhor do que MLP. Já em outros trabalhos há saídas combinadas de várias redes neurais frequentemente agregadas que aumentam a precisão. Também teve trabalhos em que o RBFNN foi usado em conjunto com outras técnicas para previsão de taxas de câmbio.

A propagação reversa é um método comum para treinar redes neurais *feedforward*, mas para a RNN é diferente e requer mais memória para simulação do que a propagação reversa, e mais conexões. Porém, tem a tendência a não ficar presa em mínimos locais, pois tem a capacidade de superar a memória de curto prazo (PARAMITA et al, 2018). O RNN fornece uma previsão melhor do que o *feedforward neural network* (FFNN). Modelos usando RNN para previsão prevêm taxas de câmbio melhores do que outros métodos e os pesquisadores concluem que RNN pode ser usado com sucesso para previsão dessas taxas (PARAMITA et al, 2018).

### 3.2 A ARQUITETURA DE 4 REDES NEURAS PROFUNDAS

O algoritmo de aprendizado profundo consiste em uma arquitetura hierárquica com muitas camadas, cada uma das quais constitui uma unidade de processamento de informações não lineares (LIU et al, 2016), podendo assim representar funções com maior complexidade se o número de camadas e unidades em uma única camada for aumentado. Este trabalho,



(LIU et al, 2016), apresenta quatro arquiteturas de aprendizado profundo, que são máquinas *restricted Boltzmann machines* (RBMs), *deep belief networks* (DBNs), *autoencoder* (AE), and *convolutional neural networks* (CNNs).

Os RBMs têm sido usados para gerar modelos estocásticos de ANNs que podem aprender a distribuição de probabilidade com relação a suas entradas. Os neurônios são restritos para formar um gráfico bipartido em uma RBM e há uma conexão completa entre as unidades visíveis e as ocultas, enquanto não existe conexão entre as unidades da mesma camada (LIU et al, 2016). Os RBMs são importantes na redução de dimensionalidade, filtragem colaborativa, classificação e aprendizado de recursos e podem ser usados como modelo generativo ou como modelo discriminativo. Em outros casos são usados como extratores de recursos no processo de pré-treinamento para tarefas de classificação (LIU et al, 2016).

O modelo de um DBN é formado por camadas onde a cada duas camadas adjacentes formam um RBM (LIU et al, 2016). A camada visível de cada RBM é conectada à camada oculta do RBM anterior e as duas camadas superiores não são direcionais. O treinamento é feito em duas fases, um pré-treinamento não supervisionado para extração de recursos e outro supervisionado para ajuste fino. Um algoritmo de aprendizado camada por camada, que otimiza os pesos de um DBN é utilizado, algoritmo de pré-treinamento também chamado de algoritmo de treinamento não supervisionado, camada por camada, ganancioso (LIU et al, 2016).

Um AE é um algoritmo de aprendizado não supervisionado usado para codificar eficientemente o conjunto de dados com o objetivo de reduzir a dimensionalidade (LIU et al, 2016), uma rede neural de *feedforward* de camada oculta semelhante à *MLP*. Semelhante aos DBNs, o processo de treinamento para um AE também pode ser dividido em dois estágios: o primeiro estágio é o aprendizado de recursos usando o aprendizado não supervisionado e o segundo é o ajuste fino da rede usando o aprendizado supervisionado (LIU et al, 2016).

As CNNs são redes neurais multicamadas que consiste em dois tipos diferentes de camadas, isto é, camadas de convolução e camadas de subamostragem com um subtipo de arquitetura discriminativa profunda com desempenho satisfatório no processamento de dados bidimensionais com topologia em grade, como imagens e vídeos (LIU et al, 2016). Nas CNNs, a convolução substituiu a multiplicação geral da matriz nas ANNs padrão. Dessa maneira, o número de pesos diminui, reduzindo a complexidade da rede (LIU et al, 2016). Atualmente, as CNNs já foram aplicadas com sucesso ao reconhecimento de manuscrito,

detecção de rosto, reconhecimento de comportamento, reconhecimento de fala, sistemas de recomendação, classificação de imagem e processamento de linguagem natural (PLN). O procedimento de treinamento para uma CNN é semelhante ao de uma RNA padrão usando retropropagação.

O aprendizado profundo lida com a análise de grandes dados usados para extrair recursos de alto nível e obter representações hierárquicas. As áreas beneficiadas são o processamento de imagens, a classificação de manuscrita, as interfaces cerebrais para computador, a classificação de sentimentos e PLN (LIU et al, 2016).

### 3.3 ANÁLISE PARA CONSCIÊNCIA SITUACIONAL

Os dados da pesquisa provêm de um experimento no qual o objetivo foi analisar o comportamento do ser humano quando inserido em um contexto que simula o tráfego urbano. O ser humano interage com o aparelho de smartphone para evitar ser atropelado. O experimento foi realizado por vinte participantes (LANGER, 2018).

Para seu objetivo (LANGER, 2018) tinha uma quantidade relativamente pequena de dados (menos de 1000 amostras). Os dados coletados de tipos variados foram agrupados e separados em arquivos salvos no formato json, para a pesquisa foram necessárias transformações e interpretação do contexto do experimento na busca de encontrar padrões e classificar se um pedestre está correndo algum risco (LANGER, 2018). As informações extraídas dos dados do experimento tem o objetivo de criar um modelo preditivo e verificar a correlação entre as variáveis para definir abordagens que levam a uma variável dependente e assim representar o nível de consciência situacional (LANGER, 2018).

A consciência situacional foi modelada através de duas abordagens: atribuição por classes e por nível quantitativo (LANGER, 2018). Na primeira, duas variáveis podem fornecer uma resposta sobre os riscos do pedestre, onde combinadas formam as classificações ‘perigo’, ‘desatento’, ‘em risco’ e ‘seguro’ (LANGER, 2018). Na segunda foram também usadas duas variáveis para criação de uma fórmula com *threshold* para uma avaliação quantitativa da consciência situacional (LANGER, 2018).

Os métodos escolhidos para a classificação foram: Regressão Logística, *K Nearest Neighbors*, Máquinas de Vetores de Suporte, Árvores de decisão, *Adaptive Boosting*, *Bagging*, *Gradient Boosting*, Floresta aleatória e Perceptron Multicamadas. Avaliou-se três aplicações, uma com a primeira abordagem (categorias) e as outras duas com a segunda abordagem nível

quantitativo com o valor de *threshold* 0.5 e 0.7. Para cada método utilizado, foi empregado o método de validação cruzada (LANGER, 2018).

Nas métricas de avaliação partindo de 15 variáveis, utilizando o método de eliminação recursiva de variáveis independentes com o método de validação cruzada resultou em uma acurácia máxima com 12 variáveis. As métricas para (LANGER, 2018) incluem também validação cruzada com 6 folds, matriz de confusão, análise de Curva ROC, métrica extraída sob a curva (AUC) média e análise de componentes principais.

Na primeira abordagem a maioria dos métodos acarretaram numa quantidade proporcional de falsos negativos muito alta, os valores de acurácia e área sob a curva ROC não demonstram o quão insatisfatório é o resultado das previsões (LANGER, 2018). A confiabilidade em classes desbalanceadas, é uma hipótese confirmada (LANGER, 2018), assim as classes desbalanceadas e as variáveis binárias, para a regressão logística e máquinas de suporte à vetor acabam por ignorar situações onde existem poucos casos de perigo (LANGER, 2018). As técnicas ensemble learning e o método de perceptron multicamada que utiliza todas as variáveis teve bom desempenho, mas utilizando PCA e redução de dimensionalidade o modelo não convergiu (LANGER, 2018). Na segunda abordagem com o valor de *threshold* em 0.5 os resultados foram similares à primeira aplicação pelo motivo do desbalanceamento de classes. Com o valor de *threshold* em 0.7 os resultados estão melhores pelo fato das classes estarem balanceadas e obterem maior acurácia e diminuição de falsos negativos para cerca de 25%. Uma análise por curvas ROC e Precisão-Sensibilidade confirmam a efetividade das análises obtidas resultando em resultados mais íntegros (LANGER, 2018).

Dados os resultados a abordagem com threshold 0.7 foi escolhida como a principal linha a ser seguida. Regressão Logística e Perceptron Multicamada melhor acurácia, e confiabilidade. A Árvore de decisão e *Adaptive Boosting* desempenho para falsos negativos, mas a diferença para as restantes não foi significativa. Técnicas de *undersampling* e *oversampling* podem ser utilizadas em conjunto com a análise de componentes principais e redução de dimensionalidade (LANGER, 2018).

### 3.4 OUTROS TRABALHOS

No trabalho de (CHEN, CHANG, 2019) é desenvolvido um método estocástico eficiente, livre de gradiente, o *ensemble neural networks* (ENN). No ENN, o processo de

otimização depende de matrizes de covariância em vez de derivadas. Este método fornece quantificação de incertezas, uma vez que é construído sob a estrutura bayesiana. Assim, (CHEN, CHANG, 2019) demonstra em seu trabalho que o ENN tem um desempenho muito melhor do que as redes neurais *bayesianas* tradicionais (BNN).

Um trabalho que busca a detecção de pedestres ocluídos em visão computacional é o de (ZOU et al, 2019). Em seu trabalho ele propõe um modelo de rede neural guiada por atenção ou *attention guided neural network* (AGNN) para busca de pedestres sobre uma oclusão. Neste modelo ele usa duas abordagens de redes neurais e uma delas é a rede neural convolucional empregada para extrair os recursos de alto nível. Na segunda, as sequências de recursos são classificadas por rede neural recorrente na ordem adequada com base nos resultados ponderados (ZOU et al, 2019).

Na busca de um método confiável para diagnóstico para predição do diabetes (EL\_JERJAWI, ABU-NASER, 2018) usa rede neural artificial visando prever se uma pessoa é diabética. Usando um modelo de ANN busca minimizar o erro médio e alcançando acurácia de 87,3%.

### 3.5 COMPARAÇÃO ENTRE OS TRABALHOS

Trabalhos	Utiliza redes neurais	Aborda consciência situacional	Faz avaliação com métodos bayesianos	Aplicada em predição de informação	Aplicada em ambiente com tráfego real ou simulado
(PARAMITA et al, 2018)	x			x	
(LIU et al, 2016)	x				
(LANGER, 2018)	x	x		x	x
(CHEN, Y.; CHANG, H., 2019)	x		x (BNN)		
(ZOU et al, 2019)	x			x	

(EL_JERJAWI, ABU-NASER, 2018)	x			x	
Este Trabalho	x	x	x	x	x

Tabela 1 – Levantamento de trabalhos relacionados

Este trabalho se diferencia dos outros, pois aborda dados que são provenientes de um *dataset* que modela um ambiente urbano com tráfego de carros onde o pedestre pode ou não estar consciente dos carros que se aproximam. Neste caso, é feita uma avaliação buscando observar a consciência situacional do pedestre. O trabalho visa usar modelos de ANN e comparar seus resultados de predição de consciência situacional com um algoritmo produzido no trabalho de (PEREIRA, 2017) que envolve uma Rede Bayesiana (RB).

## 4 DESENVOLVIMENTO

O processo de investigação e desenvolvimento deste trabalho passa por um estudo investigativo e analítico em basicamente sete etapas, que podem ser vistas no diagrama de atividade da figura 5.

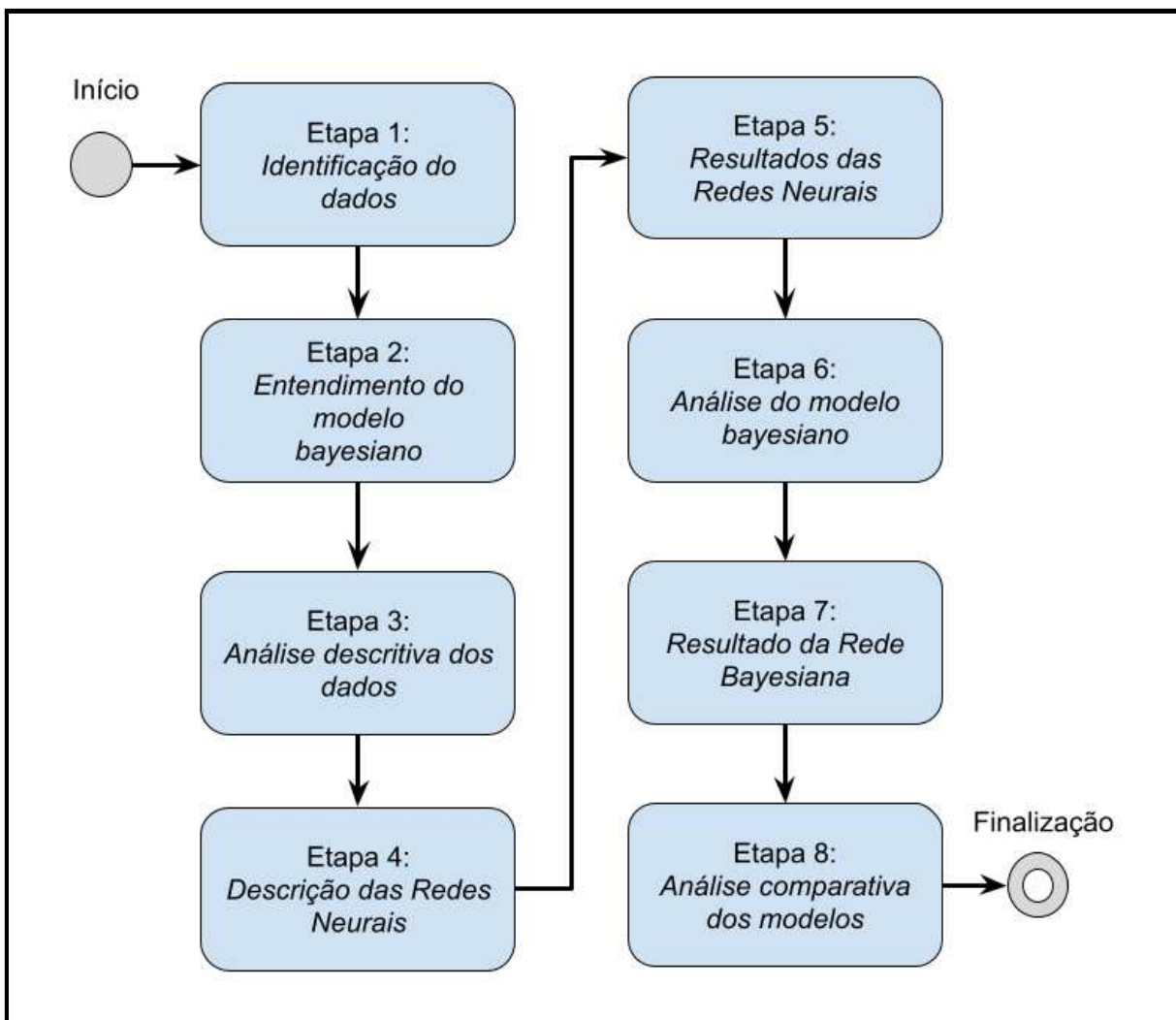


Figura 5 – Diagrama de fluxo do desenvolvimento.

Na primeira etapa deste processo de desenvolvimento está a busca e identificação da base de dados usada no trabalho. Identificação dos dados e a descrição das suas origens, os propósitos que motivaram a geração desses dados, como foi gerado, como os dados podem ser usados dentro do modelo de investigação proposto.

Na segunda etapa é descrito o estudo de entendimento e investigação do modelo de redes bayesianas escolhido, onde ele foi produzido, qual a motivação que levou a sua

produção, como são os dados que produziram tal modelo, como a nova base de dados pode assumir definições semelhantes para o modelo, como ele pode ser usado neste trabalho.

Nesta etapa, que é terceira no desenvolvimento, tem-se uma análise descritiva dos dados, ela descreve como os dados foram limpos, alterados e adaptados para se adequar às necessidades de cada modelo com o intuito de alcançar o mesmo objetivo de consciência situacional. Também são analisadas por meio de gráficos e tabelas como os dados se relacionam e quais são importantes para o objetivo proposto.

A descrição dos modelos utilizados, bem como o treinamento das redes neurais, os seus desempenhos e suas respostas para os problemas levantados neste trabalho, são os pontos importantes descritos nestas quarta e quinta etapas do desenvolvimento. Também é abordada nesta quinta etapa a questão, diante das análises feitas, de qual é a rede neural que melhor responde ao objetivo deste trabalho.

A rede bayesiana escolhida é descrita na sexta e sétima etapas do desenvolvimento de forma analítica, ou seja, como ela se comporta com os novos dados no treinamento, qual o seu desempenho, sua árvore, e os valores de probabilidade.

A última etapa é a finalização das análises, onde são comparados os resultados de cada rede com os dados selecionados e preparados para o alcance do objetivo deste documento. Tabelas e figuras de ilustração de desempenho, bem como uma análise, são apresentados nesta sétima etapa.

## 4.1 IDENTIFICAÇÃO DOS DADOS

Os dados dessa pesquisa são originários de um ambiente de realidade virtual que visa coletar informações de usuários no uso de um *smartphone* com imersão em um ambiente de trânsito. Foram realizados dois experimentos distintos, neste contexto de trânsito, e que gerou duas bases de dados. Neste trabalho será usado a base de dados do segundo experimento.

### 4.1.1 O local do ambiente de realidade virtual

O ambiente de realidade virtual chamado *Octave* localizado na Universidade de Salford - UK, visto na figura 6, foi o local onde foram produzidos dois experimentos para analisar o comportamento do ser humano em situações que envolvam tomadas de decisão e que envolvam tráfego de veículos em via urbana.

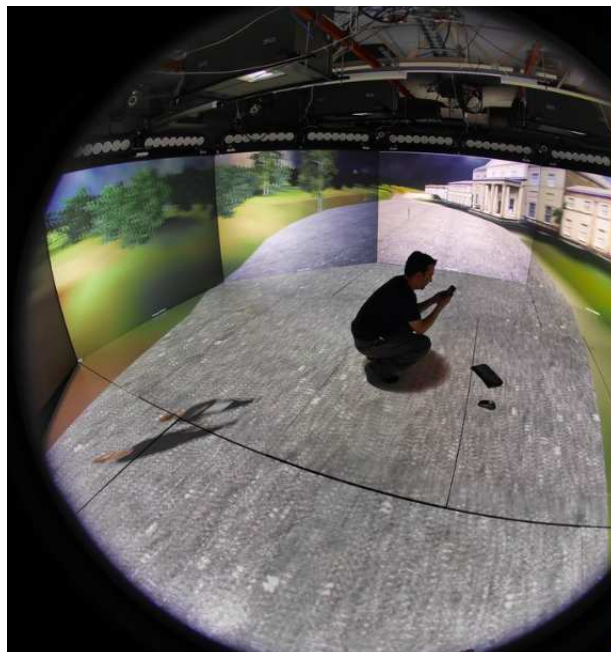


Figura 6 – Imagem do ambiente *Octave* - (Fonte: [salford](#))

Este ambiente é totalmente imersivo visualmente e auditivamente, dando ao pedestre, inserido no ambiente, projeções de um ambiente em 3D, que permite, de forma segura, analisar o comportamento deste pedestre diante de situações de distração sem um risco à sua integridade física.

#### 4.1.2 Salvando *Lemmings* no experimento 1

Este experimento como descrito em (PEREIRA, 2017) é um ambiente que simula quatro vias e que na perspectiva do usuário são quatro direções de onde virá um carro por vez. Assim, para este usuário, o carro virá da frente, de trás, da direita e da esquerda. Neste ambiente, além dos carros, podem aparecer pequenos animais chamados *lemmings*, que atravessarão a via por uma faixa de pedestre como ilustra a figura 7.

Com objetivo de validar a consciência do usuário nesta situação de trânsito, ele tem um *smartphone* em seu poder com um botão para alertar sua consciência, que deve ser apertado sempre que ele perceber que há um carro se aproximando na via da simulação. Por consequência ao apertar o botão de consciência o semáforo de pedestre fecha e o *lemmings* param de atravessar a faixa de pedestre.



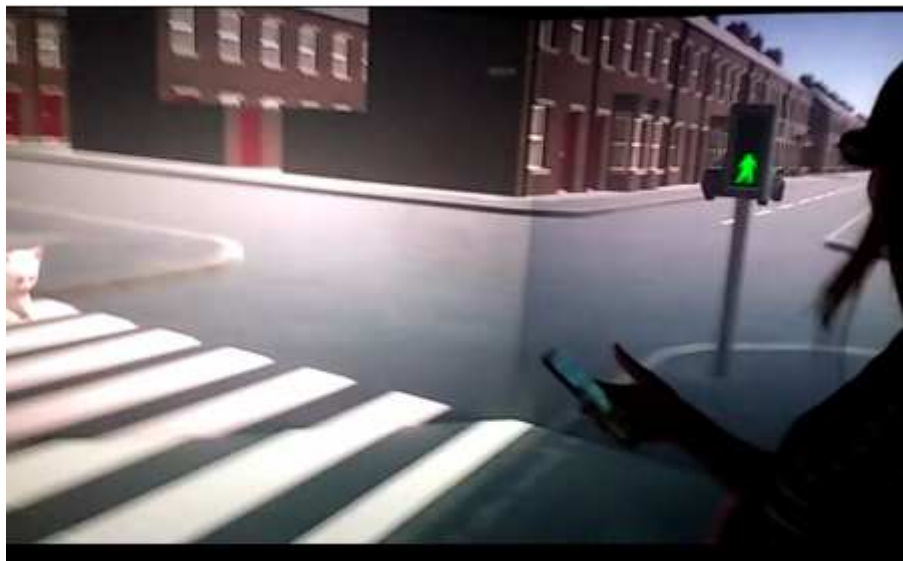


Figura 7 – Usuário imerso no ambiente da simulação - (Fonte: PEREIRA, 2017)

Outras características apontadas por (PEREIRA, 2017) no experimento é que os carros podem emitir som ou não emitir e as formas de distração se dividem em três fases no experimento. Em uma fase há apenas o celular sem outra distração, em outra há perguntas a serem respondidas pelo pedestre e noutra há o uso de fones de ouvido com som.

Cabe salientar que a base de dados produzida nesse experimento foi usada por (PEREIRA, 2017) para o desenvolvimento de seu trabalho, que visava criar um modelo de consciência situacional usando uma rede bayesiana treinada com aprendizado por reforço. O algoritmo de rede bayesiana de (PEREIRA, 2017) será aplicado à base de dados do experimento 2 como objeto de estudo e comparação com as redes desenvolvidas neste trabalho.

#### **4.1.3 Desviando de Carros no experimento 2**

Neste cenário de trânsito o objetivo do pedestre é desviar dos carros que estão vindo ao seu encontro, o intuito é evitar ser atropelado. O experimento (GELAIM, 2019) desenha um cenário de área residencial, onde há uma pista de mão dupla em que carros trafegam nos dois sentidos. O desenho aproximado do modelo pode ser visto na figura 8 e representa as seis possíveis trajetórias alcançadas pelos carros.



Figura 8 – Possíveis trajetórias alcançadas pelos carros - (Fonte: GELAIM, 2019)

Assim como no experimento 1, o experimento 2 também contém 3 modos de simulação da consciência. No primeiro o pedestre usa apenas o *smartphone* com um botão de notificação de consciência, já no segundo ele interage com jogo no *smartphone* enquanto tenta perceber os carros e apertar no botão de consciência, no terceiro ele repete o segundo experimento, mas agora com o fone de ouvido ligado com som.

As simulações são compostas de doze carros que são provenientes de forma aleatória de uma das seis direções possíveis, sendo que são dois carros de cada direção e onde um contém som e outro não, como explica (GELAIM, 2019). Podemos observar na figura 9 como o pedestre se posiciona e interage com o *smartphone*.



Figura 9 – Pedestre no ambiente de simulação - (Fonte: GELAIM, 2019)

Os dados coletados sobre o ambiente de trânsito com pedestres utilizando *smartphone*, no segundo experimento, são os dados utilizados neste trabalho. Esse dados foram mapeados de (Fonte: GELAIM, 2019) como descrito abaixo:

**Added** - Adicionado: tempo de execução em que o carro foi adicionado;

**Removed** - Removido: tempo de execução da experiência em que o carro foi removido;

**Sound** - Som: uma variável binária que indica se o carro produz sons;

***Is Occluded*** - É ocluído: Uma variável binária que indica se o carro vem de uma posição ocluída do ponto de vista do participante (ou seja, as ruas laterais);

***Critical AVG Speed*** - Velocidade crítica do AVG: velocidade média calculada pelo carro durante todo o percurso, usando o tempo necessário para alcançar o participante e a distância percorrida;

***Direction*** - Direção: um dos seis caminhos possíveis do veículo;

***Critical Time*** - Tempo crítico: tempo absoluto de execução no qual o veículo e o usuário ocuparam a mesma posição no cenário;

***Critical Time from Added*** - Tempo crítico de adição: Variável adicionada com o objetivo principal de obter o tempo do intervalo entre o veículo que aparece no cenário e o momento em que ele ocupa a mesma posição do usuário no cenário;

***Safe Lane*** - Faixa Segura: Faixa na rua onde o usuário está seguro;

***Simulation Type*** - Tipo de simulação: cada uma das três condições de distração testadas;

***SoundOff***: usando apenas o jogo como uma distração, sem música nos fones de ouvido;

***SoundOn***: além do jogo de distração, ativa a música tocada nos fones de ouvido;

***Moved to Current Lane*** - Movido para a faixa atual: horário em que o usuário passou para a posição atual;

***Moved to Next Lane*** - Movido para a próxima faixa: horário em que o usuário passou para outra posição;

***Time for Aware*** - Tempo de alerta: tempo necessário para o usuário pressionar o botão "CAR" no telefone sinalizando a conscientização de um veículo; medido a partir do tempo de inserção do carro no cenário;

***Run Over*** - Atropelar: indica se um usuário não mudou de faixa a tempo e foi atropelado pelo carro;

***User Movement*** - Movimento do usuário: Soma do deslocamento absoluto total de um usuário durante um evento;

***Head rotation*** - Rotação da cabeça: soma da quantidade total de rotação da cabeça de um usuário durante um evento;

***Is Aware*** - Está ciente: indica se o usuário pressionou o botão "CAR" a tempo. Esse estado mostra que o usuário estava ciente do veículo antes de atingir o tempo crítico.

As informações extraídas do uso do aplicativo durante as condições do tipo 2 ou 3 são:

- **Points** - Pontos: a quantidade total de pontos ganhos pelo usuário ao jogar o jogo do smartphone (distração);
- **Max Obstacle** - Obstáculo máximo: nível máximo atingido pelo usuário durante o período em que joga o jogo do smartphone (distração).
- **Deaths** - Mortes: o número de vezes que o usuário perdeu o jogo;

## 4.2 ENTENDIMENTO DO MODELO BAYESIANO

### 4.2.1 A motivação para produção do modelo

As RBs são modelos gráficos para raciocinar e representar um conhecimento sobre um meio incerto usando grafos que representam relações de probabilidade condicional (PC). Seus nós representam as variáveis, discretas ou contínuas, e os arcos são conexões diretas entre elas para expressar de forma simples as relações de causalidade das variáveis envolvidas.

O Modelo proposto por (PEREIRA, 2017) foi criado como uma tentativa de ser um modelo de consciência situacional, com base em RB, ele alcançou um resultado inicial utilizando a PC gerada no modelo. Posteriormente, para melhoria da sua rede ele aplicou no modelo RB a aprendizagem por reforço usando um modelo de agente na tentativa de aproximar a PC original do modelo de RB para uma PC mais realista ao experimento em 4.1.2. Ele criou uma relação entre RB e o Agente na qual o Agente a cada interação em sua decisão afeta a PC da RB original.

Com base nos dados produzidos no experimento relatado em 4.1.2 (PEREIRA, 2017) ele buscou como objetivo definir o quanto cada um dos fatores relacionados aos dados do experimento influenciam diretamente na consciência do usuário a fim de montar sua RB. Para isso, fez algumas considerações que o levou a definições de acordo com os dados provenientes das simulações.

### 4.2.2 Os dados usados no modelo

Em seu trabalho (PEREIRA, 2017) faz algumas considerações sobre como os dados serão utilizados para definição do modelo que expresse a consciência situacional, estas considerações são elencadas abaixo:

- I. Afirma que há um atributo (tempo) que está em valores contínuos e que para facilitar a análise e aplicação em seu trabalho foi decidido discretizá-lo, como visto na figura 10. Sendo assim ele define que:
- A. O atributo tempo, agora renomeado para percepção, foi dividido em 8 intervalos de tamanhos iguais;
  - B. Após a divisão, item (A), ele demonstra no histograma que há tempos que estão fora do padrão de criação e destruição do carro.

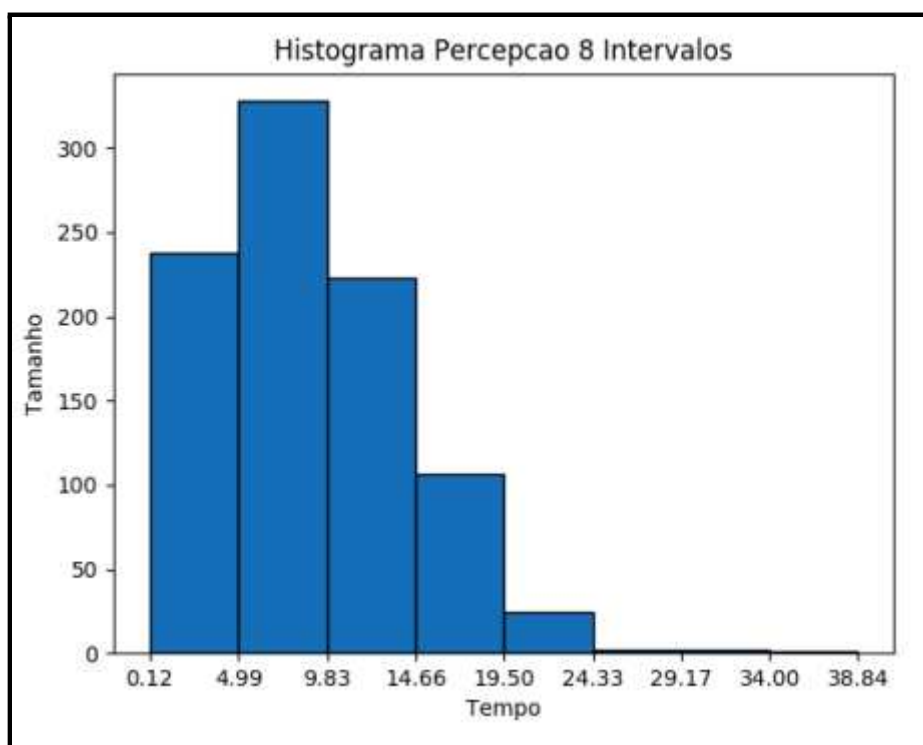


Figura 10 – Usuário imerso no ambiente da simulação - (Fonte: PEREIRA, 2017)

- C. Por causa do item (B) ele eliminou os tempos acima de 26 segundos.
- II. Ainda na base de dados ele considera que há três casos de consciência em relação a aproximação de um carro, que são:
- A. O fato dele realmente ter ficado consciente e apertado o botão do aplicativo em tempo suficiente;
  - B. O fato dele não ter apertado o botão em momento algum desde a criação e destruição do carro no ambiente virtual;
  - C. O fato dele ter apertado o botão porém o carro já ter passado de um limite relativo a zona de segurança.

- III. Outra consideração feita por (PEREIRA, 2017) foi a elaboração de uma variável chamada **tempo de consciência** que baseia-se no tempo que um usuário leva para apertar o botão do aplicativo indicando que percebeu o veículo. Tabela 2 mostra a média da consciência levantada pelo autor citado para a relação direção e som.

Direcao \ Som	Trás	Frente	Esquerda	Direita
Sim	6.279	7.069	10.635	8.297
Não	8.518	8.160	11.629	9.646

Tabela 2 – Média de tempo de Consciente - (Fonte: PEREIRA, 2017)

Neste caso a consideração é admitir um intervalo de confiança de 95%: [8.508, 9.033] para todos os tipos de carro e direções, ou seja, para o tempo médio do **tempo de consciência**. Sendo considerado em todos os tipos a direção (frente, esquerda, direita e trás) e tipo de carro (com ou sem som).

- IV. Tomando em consideração o item II de classificação, (PEREIRA, 2017) estabelece que para o item II.C que **consciência tardia** é o tempo de percepção maior que a média da sua direção e som do carro. Já o item II.B de **consciência ausente** foram todos os usuários que não sinalizaram, o restante é categorizado no item II.A como **consciência presente**.
- V. Outra ponto foi a criação da variável **DistracaoApp**, baseada nos três tipos de simulação do experimento:
- A. Sem uso de aplicativo de distração;
  - B. Com uso de aplicativo;
  - C. Com uso de aplicativo com fone de ouvido.
- Neste caso, ele juntou o item V.B com o V.C para formar um valor possível da variável **DistracaoApp**, enquanto o item V.A formaria o outro valor possível.
- VI. Por fim, (PEREIRA, 2017) estabelece, dentro de suas considerações, que o conjunto de fatores transformados e adaptados, provenientes do experimento, que influenciam a consciência e são colocados em seu modelo são: **direção do carro, tempo de**

**percepção, som do carro** e o tipo de simulação renomeada no item V para **DistracaoApp**.

O seguinte modelo apresentado na figura 11 é uma representação da RB usada por (PEREIRA, 2017) em seus experimentos e apresentando os valores originais de PC.

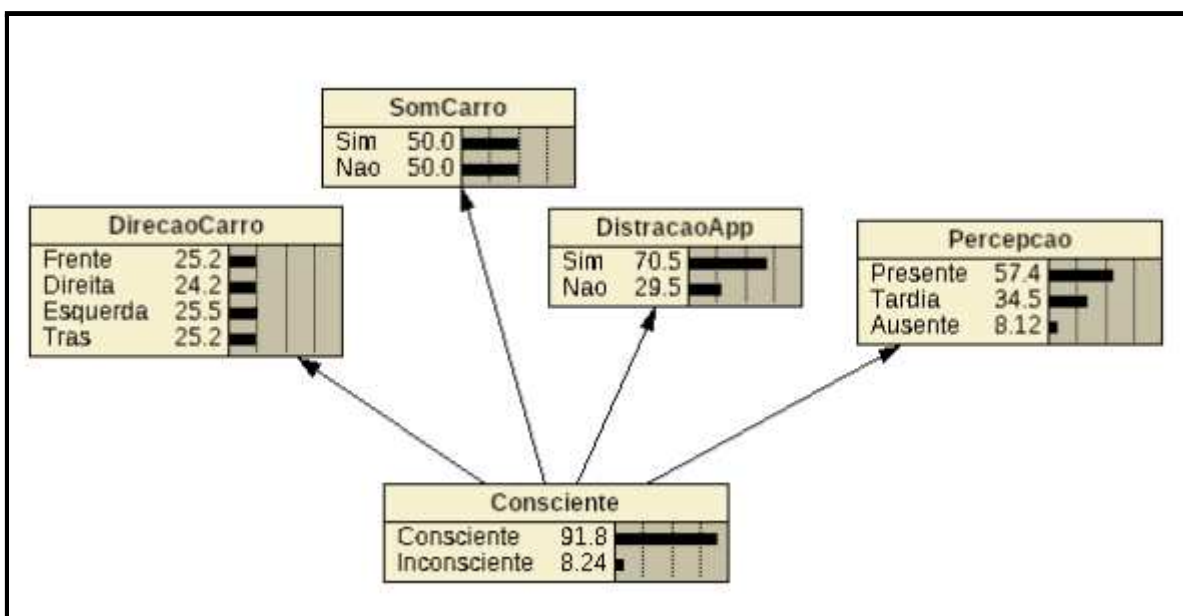


Figura 11 – Modelo de RB usada na validação de CS - (Fonte: PEREIRA, 2017)

#### 4.2.3 A nova base de dados e as definições

Analisando a base de dados descrita em 4.1.3 percebe-se grande semelhança com o experimento em 4.1.2 e o que se leva a aproveitar algumas definições colocadas por (PEREIRA, 2017). Sendo o foco dos dois experimentos uma busca por verificar a consciência do indivíduo, e observando as peculiaridades de cada experimento em simular a situação de um pedestre em meio a via urbana, pode-se objetivar encontrar uma medida de consciência também em 4.1.3. Assim como visou (PEREIRA, 2017) na base de dados descritas em 4.1.2. Define-se uma busca em 4.1.3 de uma medida de **consciência situacional**.

Outro ponto semelhante é que os dois experimentos 4.1.2 e 4.1.3 usam esquemas de distração muito semelhantes, sendo que os dois usam três estágios de distração usando celulares. Nos dois encontram-se numa primeira simulação a ausência de interação com celular, ou seja, sem uma distração forçada, na segunda uma interação forçada por uso de celular e na terceira o mesmo cenário da segunda com adição de fone de ouvido com som.

Com tal semelhança entre os experimentos, pode-se definir que o fator **distração** do aplicativo é um fator de influência na consciência também nesta base de dados, assim como definiu (PEREIRA, 2017) na base 4.1.2.

Sobre a direção dos veículos pode-se observar em 4.1.2 que o experimento traz um cruzamento onde os veículos poderão vir de 4 direções específicas e cada direção pode influenciar no tempo que levará para obter consciência. No caso do experimento 4.1.3 tem-se também a direção do carro como um fator que influencia no tempo para tomada de consciência. Esse último experimento não é um cruzamento, mas uma via de mão dupla que pode-se chamar de via A e via B, onde o pedestre estará em A ou em B. Para cada via, o carro pode vir de três direções que definiu-se como: centro, esquerda e direita. Então tem-se na verdade 6 direções que podem influenciar na tomada de consciência, diferentemente de 4.1.2 que são 4, que devem ser avaliadas e levadas em consideração no modelo. Neste caso, define-se a **direção do carro** como um fator influenciador do tempo de consciência.

Em ambos os experimentos os carros podem ou não emitir som, assim neste caso é fácil também representar o som como sendo um fator influenciador da consciência como identificou (PEREIRA, 2017), mas para o experimento 4.1.3 pode-se ir além, pois naturalmente a influência do som ativa os sentidos e consequentemente aumenta a chance de percepção. Outro ponto é que, neste experimento, alguns carros têm o estado de aproximação ao pedestre oclusa e isso afeta a percepção visual e no tempo de consciência para a identificação do carro. As informações coletadas dos carros, na base de dados, podem identificar quais os carros estão em aproximação oclusa em relação ao sentido visual do pedestre. Juntando com o som do carro, percepção auditiva, e a oclusão do carro, percepção visual, define-se um fator ativador da percepção chamado de **chance de percepção**.

Como o usuário no experimento 4.1.3 têm que indicar seu estado de consciência ao avistar o carro, então tem-se a medida da percepção do carro a qual a base de dados denomina tempo para consciência. Fazendo uso de parâmetros críticos de aproximação do carro, apontados na base de dados para cada ensaio, conjuntamente com o tempo para consciência, pode-se definir para esta base de dados um modelo de percepção que indica se o pedestre tem uma percepção útil, que seria, no caso da vida real, salvar sua vida e desviar do carro. O que em seu trabalho (PEREIRA, 2017) define como percepção (presente, tardia e ausente), neste trabalho chamar-se-á de **percepção útil**.

Nesta base de dados foi adicionada uma outra definição que pode ser chamada de **risco** que é baseada no impacto que cada carro pode oferecer na chance de atropelamento de cada pedestre do experimento. Neste caso, são 12 os carros que são utilizados para cada



pedestre do experimento e a suposição desta definição é baseada na característica do evento “acidente” que por natureza tende a ser única e normalmente sua ocorrência tem um efeito surpresa. Como os eventos da aproximação de carros tendem a serem repetidos, então por natureza e característica humana há uma chance de aprendizado sobre as características dos eventos que levam o indivíduo a ser mais atencioso e evitar acidentes através de seu aprendizado.

#### 4.2.4 Uso neste trabalho

Neste trabalho, a partir das definições realizadas e baseadas no modelo de RB definida por (PEREIRA, 2017), será implementado e testado um modelo para se verificar a PC que servirá como modelo de comparação com os modelos de redes neurais. O modelo de RB, assim como as redes neurais, usarão os dados do experimento do item 4.1.3.

### 4.3 ANÁLISE DESCRITIVA DE DADOS

#### 4.3.1 Preparando a percepção útil

A **percepção útil** tenta mensurar e extrair em termos práticos e baseados em parâmetros reais e nos dados dos experimentos quais os valores expressam um padrão mais realístico de consciência. A intenção é testar a suposição que considera a inconsciência, a consciência e a consciência tardia. Usando os parâmetros Time for Aware (TfA), Time Critical from Added (TCfA) e Run Over (RO) são criadas suposições baseadas em parâmetros reais como Velocidade de Caminhada Lenta (VCL) igual a 0,63, Velocidade de Caminhada Padrão (VCP) igual a 1,22 e Velocidade de Caminhada Rápida (VCR) igual a 1,83. Os três valores de caminhada são para avaliações analíticas comparativas, sendo assim, apenas a VCP será adotada para criação da **percepção útil**. Os valores de caminhada são retirados do trabalho de SILVA et al. (2012), no estudo sobre velocidade de pedestres em travessias com valores de 1,22 m/s para média, com 0,63 m/s de mínimo e máximo de 1,83 m/s. Quanto ao Tempo de Reação (TR) igual a 0,45 foi o tempo máximo retirado de FEIS (2012) para uma pessoa jovem em bom estado de saúde. A Largura Máxima a Percorrer (LMP) igual 2,6 é baseado na largura máxima de um veículo permitido pelo Contran, onde o conselho nacional

de trânsito, na resolução Nº 210 de 13 de novembro de 2006 e no art 1 do item I diz que: a largura máxima é 2,60m.

Os dados para avaliação da consciência foram separados e analisados segundo as regras abaixo:

1. Inconsciência:
  - a. Considera todas as mortes RO como inconsciência;
  - b. Considera a não marcação no campo Time for Aware (TfA) no experimento como inconsciência.
2. Consciência tardia amostrada;
  - a. Considera que o tempo de consciência é maior que o tempo crítico, onde  $TfA > TCfA$ .
3. Consciência tardia calculada mínima;
  - a.  $(TfA + (LMP/VCR) + TR) > TCfA$ .
4. Consciência tardia calculada padrão;
  - a.  $(TfA + (LMP/VCP) + TR) > TCfA$ .
5. Consciência tardia calculada máxima;
  - a.  $(TfA + (LMP/VCL) + TR) > TCfA$ .
6. Consciência Útil calculada amostrada
  - a. Total de amostra - (Inconsciência + Consciência tardia experimental).
7. Consciência Útil para Tempo de Movimento (TM) rápido
  - a. Total de amostra - (Inconsciência + Consciência tardia calculada mínima).
8. Consciência Útil para TM padrão
  - a. Total de amostra - (Inconsciência + Consciência tardia calculada padrão)
9. Consciência Útil para TM lento
  - a. Total de amostra - (Inconsciência + Consciência tardia calculada máxima)

A tabela 3 apresenta os dados para os modelos de **percepção útil** em porcentagem de influência e observa-se que aparentemente os pedestres do experimento não tem quase inconsciência tardia e 80% parecem estar conscientes.

Modelo da Percepção Útil	Amostrada	TM Rápido	TM Padrão	TM Lento
Inconsciência	17,50	17,50	17,50	17,50
Consciência tardia	3,06	18,19	32,36	40,42

Consciência Útil	79,44	64,31	50,14	42,08
------------------	-------	-------	-------	-------

Tabela 3 – Modelo de percepção consciência x tempo de movimento.

No entanto, ao aplicar ao experimento condições reais que permitam uma tempo de reação ao usuário para livrar-se de uma condição de acidente, observamos que a taxa de consciência começa a cair no experimento e a taxa de consciência tardia começa a aumentar muito, em condições reais a possibilidade de mortes aumentaria. A figura 12 ilustra esse comportamento.

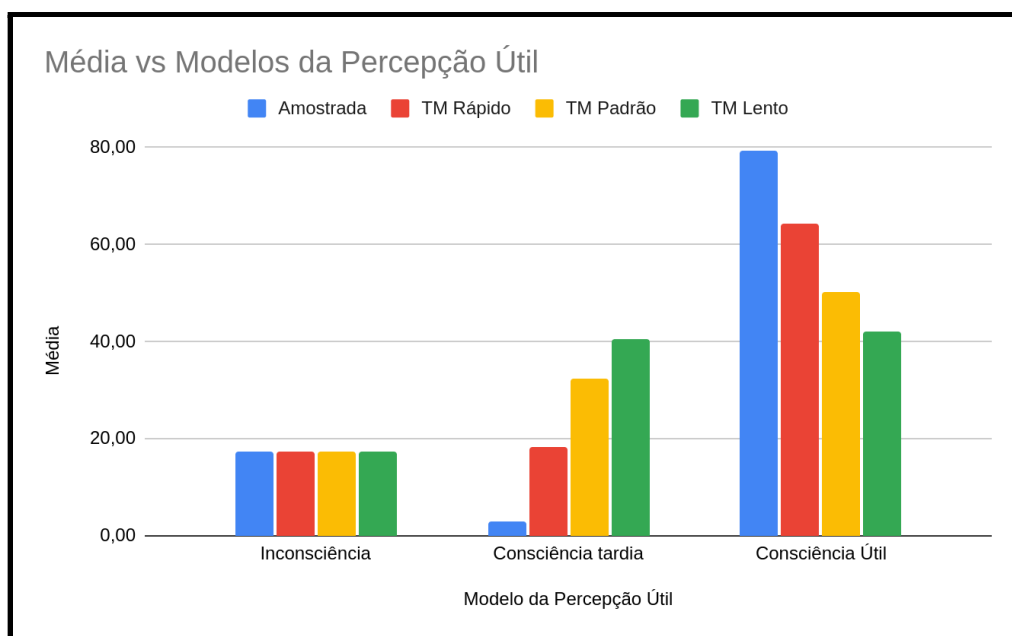


Figura 12 – Proporção de em % para cada modelo da percepção útil.

A figura 13 ilustra como a velocidade de caminhada, o percurso e o tempo de reação natural do ser humano podem influenciar no tempo de percepção útil para o pedestre, quanto mais lento ele for para reagir. Em situações reais a velocidade baixa pode ser assemelhada a pessoas idosas ou mulheres com crianças pequenas, etc.

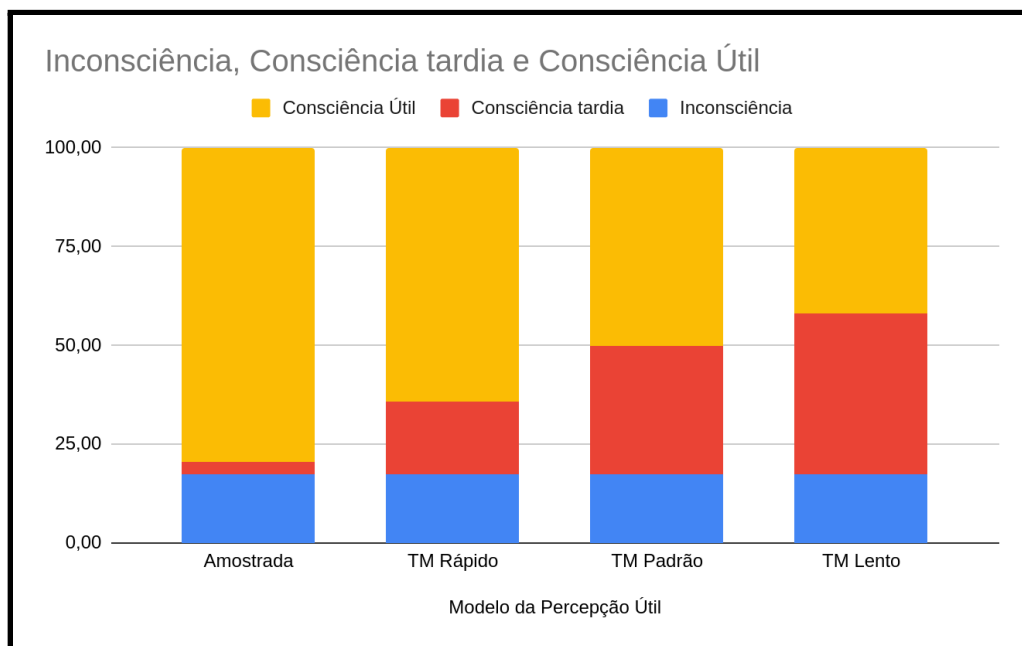


Figura 13 – Proporção de em % para cada modelo da percepção útil.

No experimento o número de atropelamento é baixo, pois os pedestres já estavam na faixa de segurança correta, mas ao adicionarmos as condições de TMs podemos observar uma quantidade de pedestres que ficou consciente tardiamente, mas que não foram atropelados neste experimento. Para este trabalho será usado uma TM Padrão.

#### 4.3.2 Preparando a chance de percepção

A chance de percepção tenta classificar qualitativamente e mapear quais os estímulos sensoriais que são estimulados durante cada evento do experimento. Sendo assim através das variáveis do experimento *Sound* e *Is Occluded* formou-se uma nova variável chamada **chance de percepção**, onde para cada tupla amostral <Sound, Is Occluded> do experimento 4.1.3 usou-se a seguinte definição dada pelo algoritmo abaixo:

---

**Algoritmo 1:** Criando a variável chance de percepção

---

```

se Sound is true and IsOccluded is false
    PerceptionChance <- ALL
else se Sound is true and IsOccluded is true
    PerceptionChance <- AUDIO
else se Sound is false and IsOccluded is false
    PerceptionChance <- VISUAL

```

```
else
  PerceptionChance <- WITHOUT
```

---

Dentro dos eventos que provocaram atropelamento, temos que 1 recebeu o valor ALL, 3 o valor AUDIO, 45 o VISUAL e 14 WITHOUT caracterizando a princípio um maior número de atropelamento relacionado a carros sem emissão de som.

### 4.3.3 Preparando a direção do carro

Para a avaliação da direção como influenciador da consciência tomou-se como base a variável *Direction*. No experimento a variável pode assumir 6 valores distintos que são *Front*, *FrontLeft*, *FrontRight*, *Back*, *BackLeft*, *BackRight* e o pedestre pode assumir dois estados espaciais, o lado da pista segura ou da pista onde passará o carro, do qual por convenção chamar-se-á de pista A e pista B.

A figura 14 é uma ilustração de um evento do experimento que mostra a pista A e a pista B, temos um cruzamento com uma certa distância do pedestre do qual poderá vir o carro. Se o carro vem da pista B, então temos que as direções poderão ser *Front*, *FrontLeft*, *FrontRight*. No entanto, se a pista for a pista A, então temos que as direções poderão ser *Back*, *BackLeft*, *BackRight*. No experimento o pedestre sempre começa na pista A para o primeiro carro como exemplificado nesta figura.

Ainda, considerando a figura 14, observa-se que mesmo o pedestre sinalizando consciência a 0,1 segundos antes que o carro ultrapasse o tempo crítico, que seria a chegada no ponto B, o pedestre continuará salvo e o atropelamento não ocorrerá, pois desde o princípio ele já está na sua pista de segurança e qualquer distração ou atenção tardia tornar-se independente do risco associado ao atropelamento, isso de forma mais estrita e sem um suposto deslocamento inesperado por outro motivo. Então diante desta evidência descrita uma análise do movimento esperado do pedestre foi realizada para dar características de risco diferentes para cada direção do carro.

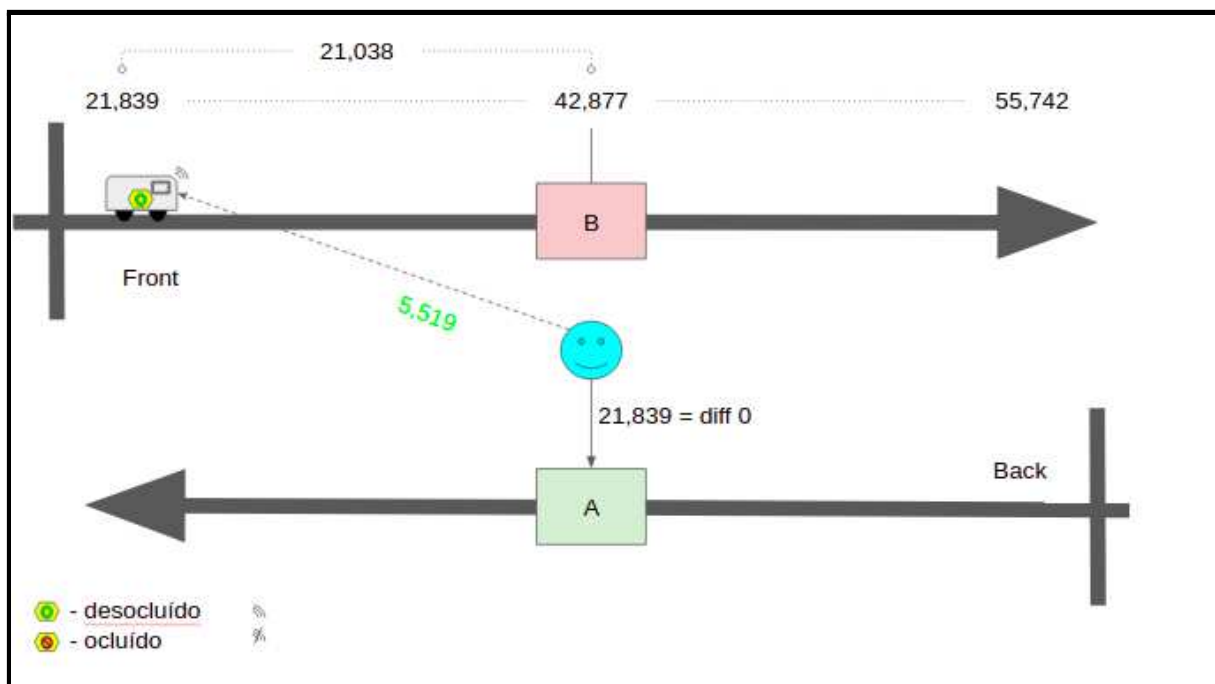


Figura 14 – Ilustração de um evento do experimento.

Um carro na direção *Front* e o pedestre na pista A, como o da figura 14, tem um risco associado diferente de um carro na mesma direção e com o pedestre na pista B. Por isso, houve a necessidade de requalificar cada direção como se ela fosse na verdade duas direções. Para alcançar tal objetivo, o primeiro passo foi criar duas novas variáveis auxiliares chamadas *ideal lane* para identificar em qual pista o carro está surgindo e a *before lane* que armazena *ideal lane* do evento anterior. O carro vem da pista A se ele surge de *Front*, *FrontLeft*, *FrontRight* e de B se ele surge de *Back*, *BackLeft*, *BackRight*.

O segundo passo consiste em criar mais uma variável auxiliar chamada de *ideal move* que mapeia se o pedestre deve mover-se ou não naquele evento. Essa variável pode assumir dois valores que são *Move* e *Stop*, sinalizando se o pedestre deve mover-se ou ficar parado no evento.

---

**Algoritmo 2:** Criando a variável auxiliar *ideal move*

---

```

se Car is Car00
  se ideal_lane is lane_B
    ideal_move <- MOVE
  else
    ideal_move <- STOP

```

```

else
  se ideal_lane equal before_lane
    ideal_move <- STOP
  else
    ideal_move <- MOVE

```

---

O último passo consiste em criar uma nova variável ‘direção do carro’ para incorporar o fator de risco associado ao movimento do pedestre e substituir a variável *Direction*. Para isso são usados os valores de *ideal move* e *Direction* formando novos valores qualitativos. Os novos valores são F (front), FL (front left), FR (front right), B (back), BL (back left), BR (back right), WF (warning front), WFL (warning front left), WFR (warning front right), WB (warning back), WBL (warning back left) e WBR (warning back right).

---

**Algoritmo 3:** Criando a variável direção do carro

---

```

se Direction is Front
  se ideal_move is MOVE
    new_direction <- WF
  else
    new_direction <- F
else se Direction is FrontLeft
  se ideal_move is MOVE
    new_direction <- WFL
  else
    new_direction <- FL
else se Direction is FrontRight
  se ideal_move is MOVE
    new_direction <- WFR
  else
    new_direction <- FR
else se Direction is Back
  se ideal_move is MOVE
    new_direction <- WB
  else
    new_direction <- B
else se Direction is BackLeft
  se ideal_move is MOVE
    new_direction <- WBL
  else
    new_direction <- BL
else se Direction is BackRight
  se ideal_move is MOVE

```

```

new_direction <- WBR
else
new_direction <- BR

```

Para um melhor entendimento de como ficou a influência dessas novas definições foi mapeado na tabela 4 o número de eventos para cada nova direção na base de dados juntamente com o atropelamento dado pela variável *Run Over*. Na tabela 4 o risco é dado pela razão de atropelamentos por eventos.

	B	BL	BR	F	FL	FR	WB	WBL	WBR	WF	WFL	WFR
Eventos	44	52	48	58	56	61	76	68	72	62	64	59
Atropelamento	1	2	2	0	0	0	37	6	6	0	6	3
Risco	2,27	3,85	4,17	0,00	0,00	0,00	48,68	8,82	8,33	0,00	9,38	5,08

Tabela 4 – O risco é dado pela razão do atropelamento por evento em %.

A figura 15 ajuda a ilustrar essa distribuição de risco quando adiciona-se a posição do usuário como fator influenciador da direção, assim para quase todos os casos observa-se que o risco aumenta no mínimo 2 vezes ou mais com a necessidade de movimento do pedestre.

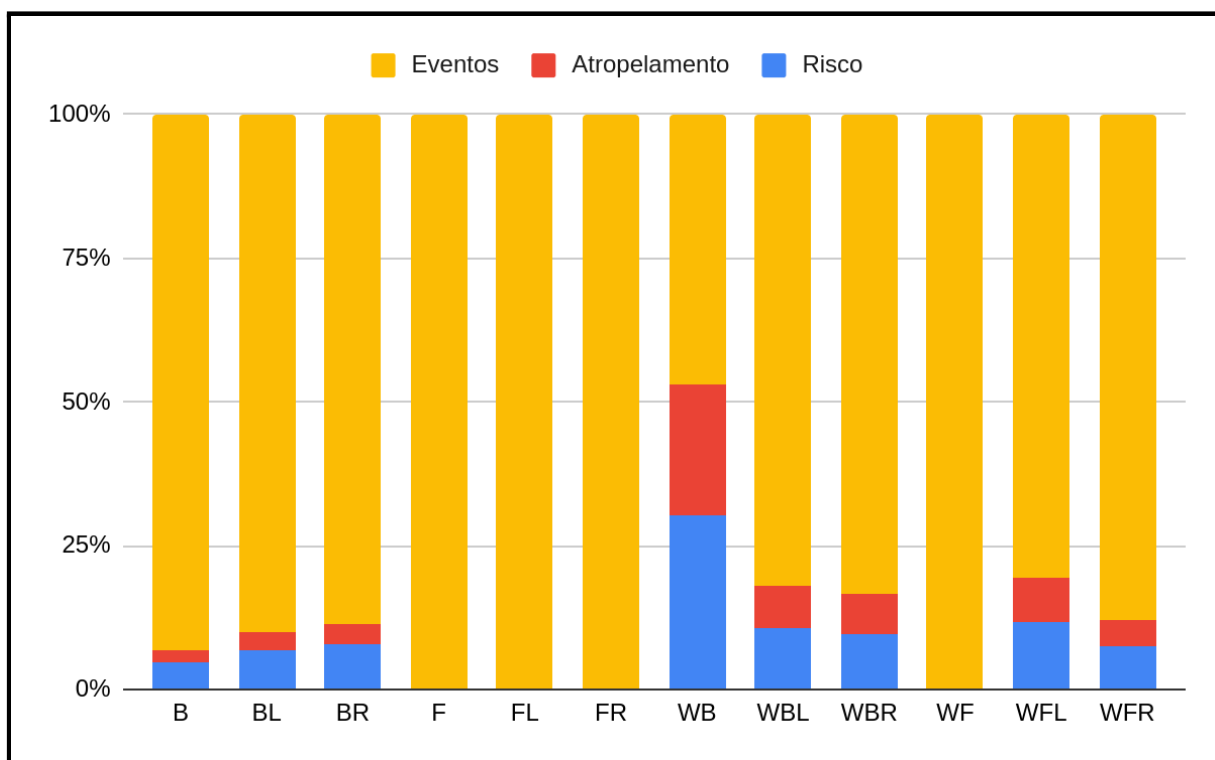




Figura 15 – Ilustração de um evento do experimento.

#### 4.3.4 Preparando a distração

A distraçãoApp do modelo de (PEREIRA, 2017) ficou com uma separação simples e a **distração** nesse trabalho seguirá uma abordagem semelhante. Ou seja, devido a enorme semelhança entre os experimentos 4.2.2 e 4.2.3 usaremos uma abordagem classificando o uso em 3 categorias, aplicativo sem som, aplicativo com som e sem aplicativo. Assim, a diferença fundamental de separação nesse modelo foi criar 3 categorias para discriminar os 3 experimentos em vez de duas como fez (PEREIRA, 2017). A transformação realizada para este caso foi renomear a variável original do experimento *Simulation Type* para **distração** como exemplifica a tabela 5.

Distração	Simulation Type
NONE	Button
SOFF	SoundOff
SON	SoundOn

Tabela 5 – Mapeamento *Simulation Type* para Distração.

#### 4.3.5 Preparando o risco

Para alcançar a variável que foi definida como risco as seguintes avaliações matemáticas foram feitas:

- Foi dividido o conjunto de dados em 3 partes considerando 3 experimentos de distração. Sendo eles *Button*, *SoundOff* e *SoundOn*;
- Para cada carro em cada conjunto foi calculado a média de atropelamento do carro e o desvio padrão. Tabela 6;
- Foi calculado o erro padrão da média como um fator dado pela variação das amostras dos carros. Assim, foi utilizada a medida da dispersão do erro padrão como uma medida para aferir a chance de atropelamento para cada carro em cada conjunto. Ou seja, um erro zero significa um carro que não provocou atropelamento.

- Assumiu-se todo conjunto como independente de outro conjunto, logo cada variável, por exemplo carro zero, tem o seu conjunto de medida independente de outro carro zero de outro conjunto.
- Usou-se a definição estatística de que ao somarmos ou subtrairmos grandezas estatisticamente independentes o erro será dado pela raiz quadrada da soma dos quadrados dos erros de cada uma das grandezas. Isso se realizou com o intuito de unir todos os conjuntos de dados em um único conjunto contendo os erros enumerados por carros;
- Por fim, uma análise sobre o conjunto dos erros, apontou a média dos erros e o desvio padrão dos erros que foram usados para avaliar e discriminar os riscos associados a cada carro. O erro padrão foi transformado em erro relativo que é o valor do erro vezes 100. A figura 16 ilustra o gráfico de distribuição dos erros e as linhas limites traçadas a partir da média da variação e do desvio padrão. A tabela 7 apresenta uma classificação baseada no erro, na média e no desvio padrão.

<b>Button</b>	<b>Car00</b>	<b>Car01</b>	<b>Car02</b>	<b>CarN</b>
erro padrão	0,092	0,000	0,050	?
desvio (S)	0,410	0,000	0,224	?
média	0,800	1,000	0,950	?
amostras	20,000	20,000	20,000	?
<b>SoundOff</b>	<b>Car00</b>	<b>Car01</b>	<b>Car02</b>	<b>CarN</b>
erro padrão	0,082	0,082	0,069	?
desvio (S)	0,366	0,366	0,308	?
média	0,850	0,850	0,900	?
amostras	20,000	20,000	20,000	?
<b>SoundOn</b>	<b>Car00</b>	<b>Car01</b>	<b>Car02</b>	<b>CarN</b>
erro padrão	0,092	0,082	0,082	?
desvio (S)	0,410	0,366	0,366	?
média	0,800	0,850	0,850	?
amostras	20,000	20,000	20,000	?
<b>Total</b>	<b>Car00</b>	<b>Car01</b>	<b>Car02</b>	<b>CarN</b>
Erro relativo	18,869	13,629	13,381	?

Tabela 6 - A média de atropelamento do carro e o desvio padrão.

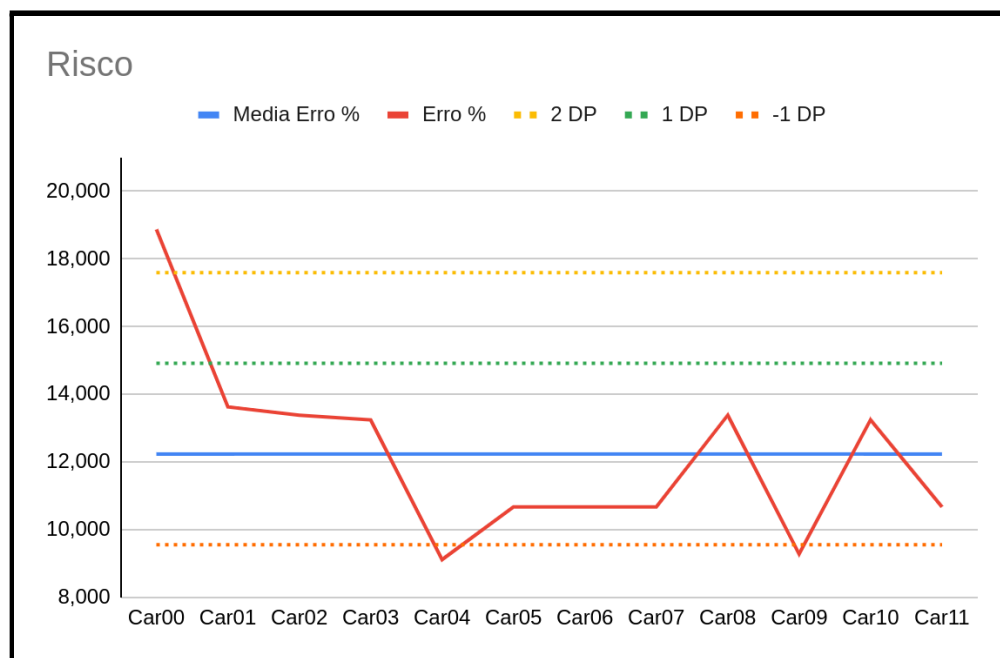


Figura 16 - Gráfico de Risco para cada carro.

RISCO	desvio (S)	-1 DP	Média	1 DP	2 DP
Valores	2,677	9,559	12,236	14,914	17,591
CRÍTICO					ERRO > 2DP
SEVERO				ERRO > 1DP	
MÉDIO			ERRO > MÉDIA		
BAIXO		ERRO > -1 DP			
MÍNIMO	ERRO <= -1 DP				

Tabela 7 - Classificação de risco.

#### 4.3.6 Preparando a consciência

Neste trabalho, busca-se um modelo de consciência, semelhante ao considerado por (PEREIRA, 2017), que considerou dois estados no seu modelo: consciente e inconsciente. No entanto, buscamos encontrar 4 estados que são: consciência imediata, consciência normal, risco de inconsciência e inconsciência. Para alcançar tal definição de valores usou-se o seguinte critério:

- Toma-se a média do tempo de consciência da variável ‘Time for Aware’ para todos os pedestres com **percepção útil** presente e obtém-se o que pode-se chamar de ‘média presente’;
- Para todo pedestre com percepção útil presente e tempo abaixo da ‘média presente’ foi considerado uma consciência IMEDIATA;
- Para todo pedestre com percepção útil presente e tempo acima ou igual a ‘média presente’ foi considerado uma consciência NORMAL
- Se o ‘Time for Aware’ for maior que ‘Time Critical from Added’ menos o tempo calculado para movimento do pedestre TM do item 4.3.1 neste trabalho, então foi considerado como RISCO de inconsciência.
- Toda a ausência de ‘Time for Aware’ e os dados que não se adequaram às categorias anteriores foram considerados como uma consciência PERDIDA;

A figura 17 mostra a contagem dos tipos de consciência levantados na base de dados e a tabela 8 abaixo ilustra de forma abreviada como ficou o mapeamento de consciência na base de dados.

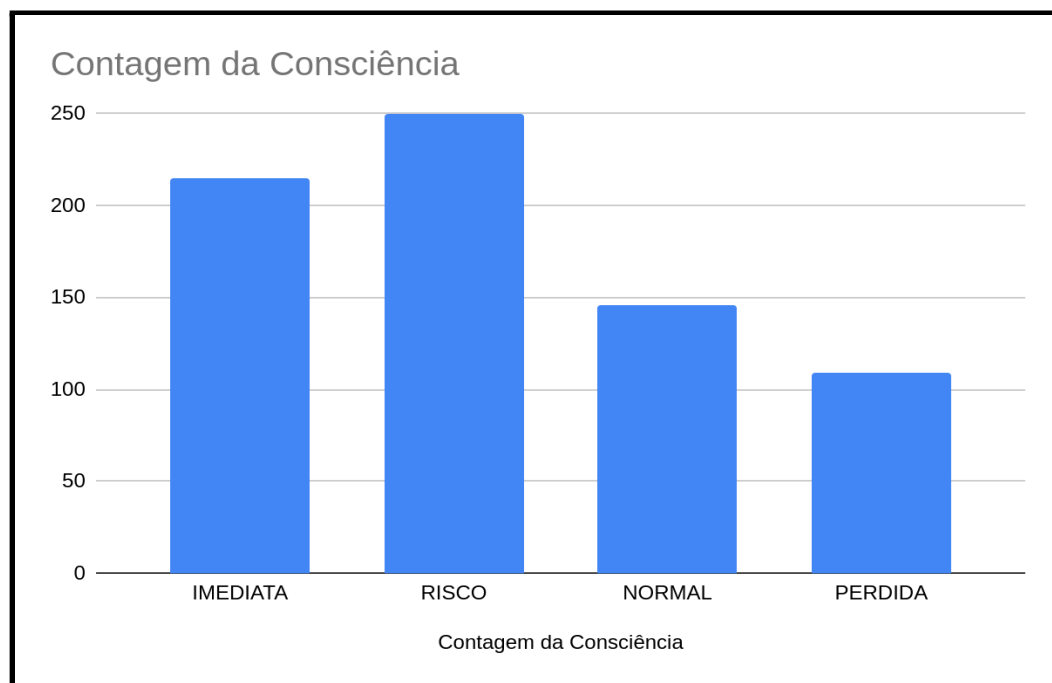


Figura 17 – Tipos de consciência levantados na base de dados.

<b>Pedestre</b>	<b>Carro</b>	<b>Consciência</b>	<b>Percepção Útil</b>	<b>...</b>
uid 11	Car 06	IMEDIATA	PRESENTE	...
uid 09	Car 03	PERDIDA	AUSENTE	...
uid 13	Car 00	RISCO	TARDIA	...
uid 03	Car 10	NORMAL	PRESENTE	...
uid 16	Car 03	RISCO	AUSENTE	...
...	...	...	...	...

Tabela 8 – Fragmento da base de dados.

#### 4.3.7 Base de dados preparada

Após trabalhar-se com os dados chegou-se a 6 variáveis que expressam características do experimento formando uma base de dados que será usada para a análise na rede bayesiana. A nova base de dados para análise é vista de forma parcial na tabela 9.

<b>User</b>	<b>Car</b>	<b>Consciência</b>	<b>Percepção Útil</b>	<b>Distração</b>	<b>Direção Carro</b>	<b>Chance Percepção</b>	<b>Risco</b>
uid 10	Car 09	NORMAL	PRESENTE	SOFF	WF	ALL	MÍNIMO
uid 02	Car 10	RISCO	TARDIA	SOFF	WFR	ÁUDIO	MÉDIO
uid 06	Car 08	PERDIDA	AUSENTE	SON	BR	VISUAL	MÉDIO
uid 13	Car 04	PERDIDA	AUSENTE	SON	WB	VISUAL	MÍNIMO
uid 17	Car 08	IMEDIATA	PRESENTE	SOFF	B	ALL	MÉDIO
uid 10	Car 05	IMEDIATA	PRESENTE	SOFF	F	VISUAL	BAIXO
uid 05	Car 01	IMEDIATA	PRESENTE	SOFF	F	ALL	MÉDIO
uid 17	Car 08	NORMAL	PRESENTE	NONE	WBR	VISUAL	MÉDIO
uid 08	Car 09	NORMAL	PRESENTE	SON	WFL	ÁUDIO	MÍNIMO
uid 14	Car 04	IMEDIATA	PRESENTE	SOFF	F	ALL	MÍNIMO
uid 06	Car 00	IMEDIATA	PRESENTE	NONE	WB	ALL	CRÍTICO
uid 16	Car 00	IMEDIATA	PRESENTE	NONE	WB	ALL	CRÍTICO

Tabela 9 – Fragmento da base com novas definições.



FR	3	0	0	1	0	0	0	0	0	0	0	0	0
WF	4	0	0	0	1	0	0	0	0	0	0	0	0
B	5	0	0	0	0	1	0	0	0	0	0	0	0
BL	6	0	0	0	0	0	1	0	0	0	0	0	0
BR	7	0	0	0	0	0	0	1	0	0	0	0	0
WFR	8	0	0	0	0	0	0	0	1	0	0	0	0
WBR	9	0	0	0	0	0	0	0	0	1	0	0	0
WBL	10	0	0	0	0	0	0	0	0	0	1	0	0
WFL	11	0	0	0	0	0	0	0	0	0	0	1	0
WB	12	0	0	0	0	0	0	0	0	0	0	0	1

Tabela 13 – Valores para Direção Carro.

ID	A	B			
Chance Percepcao	C.P	ALL	AUDIO	VISUAL	WITHOUT
ALL	1	1	0	0	0
AUDIO	4	0	1	0	0
VISUAL	8	0	0	1	0
WITHOUT	12	0	0	0	1

Tabela 14 – Valores para Chance Percepção.

ID	A	B			
Risco	R	MÍNIMO	BAIXO	MÉDIO	CRÍTICO
MÍNIMO	1	1	0	0	0
BAIXO	4	0	1	0	0
MÉDIO	8	0	0	1	0
CRÍTICO	12	0	0	0	1

Tabela 15 – Valores para Risco.

## 4.4 DESCRIÇÃO DAS REDES NEURAIS

### 4.4.1 Escolha da Rede Função de Base Radial

A pesquisa sobre taxa forex em 3.1 motivou os testes com a rede RBFNN. Na pesquisa 3.1 ela apresentou resultados satisfatórios e desempenho melhor que a MLP, então uma breve investigação sobre a rede mostrou como é fácil treiná-la e torná-la uma opção para este trabalho investigativo. Outra característica atraente é a classificação com base na dimensionalidade, pois como descreve (HAYKIN, 2001) uma RBFNN busca a classificação de padrões através de uma transformação dos espaços de alta dimensionalidade de uma forma não-linear. Esta característica onde há uma transformação da dimensionalidade com a criação de um cluster por neurônio antes de uma separação linear apontou que a RBFNN poderia ser uma rede candidata para este trabalho.

### 4.4.2 Treinamentos da Rede de Função de Base Radial

Neste trabalho foi usado um treinamento semi-supervisionado usando duas etapas. Sendo a camada oculta um conjunto de centros onde cada neurônio é um centro, o treinamento é de forma não supervisionada usando algum método de clusterização por centróides, usando k-means como método de clusterização e a gaussiana como um função radial de aproximação. A segunda parte da rede foi treinada de forma supervisionada para obtenção dos pesos sinápticos.

Sendo a saída da rede dada por  $\mathbf{G}_{p \times k} \mathbf{W}_{k \times m} = \mathbf{Y}_{p \times m}$  obtém-se  $\mathbf{G}_{p \times k}$  para uma a gaussianas através de

$$\varphi_j^{(i)} = e^v, v = \frac{num}{den}, num = - \|x^{(i)} - \mu_j\|_2^2, den = \sigma^2, \sigma = \frac{num2}{den2}, num2 = \max(\{\mu_1, \dots, \mu_k\}) - \min(\{\mu_1, \dots, \mu_k\}), den2 = \sqrt{2k}, \text{ onde } \mathbf{G}_{p \times k} = \varphi_j^{(i)}.$$

Obtém-se  $\mathbf{W}_{k \times m}$  de  $\mathbf{W}_{k \times m} = \mathbf{G}_{p \times k}^+ \mathbf{Y}_{p \times m}$  onde  $\mathbf{G}^+$  é a pseudo inversa da matriz  $\mathbf{G}$ , pois esta geralmente não é uma matriz quadrada. O algoritmo não é iterativo, sendo assim, encontrar os pesos é algo simples como uma multiplicação de matrizes.





uid 01	Car 10	0	0	1	0	6	12	6	12	8
uid 01	Car 11	1	0	0	0	1	12	8	4	4
uid 02	Car 00	0	0	1	0	6	12	10	12	12
uid 02	Car 01	0	0	0	1	12	12	5	8	8
uid 02	Car 02	0	1	0	0	1	12	8	4	8
...	...	...	...	...	...	...	...	...	...	...
uid 02	Car 11	1	0	0	0	1	12	1	8	4
uid 03	Car 01	0	0	0	1	12	12	12	8	12

Tabela 16 – Valores IDs em sequência experimental.

IDs		Consciência				Inputs				
User	Car	Imediata	normal	risco	perdida	Percepção Útil	Distração	Direção Carro	Chance Percepção	Risco
uid 10	Car 09	0	1	0	0	1	6	4	1	1
uid 10	Car 07	0	1	0	0	1	6	11	4	4
uid 06	Car 08	0	0	0	1	12	1	7	8	8
...	...	...	...	...	...	...	...	...	...	...
uid 06	Car 00	1	0	0	0	1	12	12	1	12
uid 16	Car 00	1	0	0	0	1	12	12	1	12
uid 18	Car 04	0	0	1	0	6	1	8	4	1
uid 14	Car 06	0	1	0	0	1	1	8	12	4
uid 11	Car 01	0	0	1	0	6	6	3	12	8
...	...	...	...	...	...	...	...	...	...	...
uid 15	Car 03	0	0	1	0	6	12	10	12	8
uid 09	Car 10	1	0	0	0	1	12	9	1	8

Tabela 17 – Treino 1: valores IDs em sequência aleatória.

Os dados foram modificados inicialmente para 3 treinamentos e em todas as modificações a sequência da tabela 17 foi usada. Da base, os primeiros 70% dos dados eram para treinamento e os 30% restantes eram para teste. No primeiro treinamento os dados foram categorizados como descrito na tabela 17, no segundo categorizados como na tabela 18 e no terceiro um desbalanceamento foi testado como mostrado na tabela 19. Apenas lembrando, que nas tabelas abaixo as colunas referências não são inputs para a rede.

Ref.:	Inputs												
	<b>Percepção Útil</b>												
pesos	Presente				Tardia				Ausente				
1	1				0				0				
6	0				1				0				
12	0				0				1				
	<b>Distração</b>												
pesos	SON				SOFF				NONE				
1	1				0				0				
6	0				1				0				
12	0				0				1				
	<b>Direção Carro</b>												
pesos	F	FL	FR	WF	B	BL	BR	WFR	WBR	WBL	WFL	WB	
1	1	0	0	0	0	0	0	0	0	0	0	0	
2	0	1	0	0	0	0	0	0	0	0	0	0	
3	0	0	1	0	0	0	0	0	0	0	0	0	
4	0	0	0	1	0	0	0	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	0	0	
6	0	0	0	0	0	1	0	0	0	0	0	0	
7	0	0	0	0	0	0	1	0	0	0	0	0	
8	0	0	0	0	0	0	0	1	0	0	0	0	
9	0	0	0	0	0	0	0	0	1	0	0	0	
10	0	0	0	0	0	0	0	0	0	1	0	0	
11	0	0	0	0	0	0	0	0	0	0	1	0	
12	0	0	0	0	0	0	0	0	0	0	0	1	
	<b>Chance Percepção</b>												
pesos	ALL				ÁUDIO				VISUAL				WITHOUT
1	1				0				0				0
4	0				1				0				0
8	0				0				1				0
12	0				0				0				1
	<b>Risco</b>												
pesos	Mínimo				Baixo				Médio				Crítico
1	1				0				0				0
4	0				1				0				0
8	0				0				1				0
12	0				0				0				1

Tabela 18 – Treino 2: valores de forma binarizada.

inputs	Referências												
<b>P= 0</b>	<b>Percepção Útil</b>												
input	Presente				Tardia				Ausente				
1+P	1				0				0				
6+P	0				1				0				
12+P	0				0				1				
<b>P= 24</b>	<b>Distração</b>												
input	SON				SOFF				NONE				
1+P	1				0				0				
6+P	0				1				0				
12+P	0				0				1				
<b>P= 48</b>	<b>Direção Carro</b>												
input	F	FL	FR	WF	B	BL	BR	WFR	WBR	WBL	WFL	WB	
1+P	1	0	0	0	0	0	0	0	0	0	0	0	
2+P	0	1	0	0	0	0	0	0	0	0	0	0	
3+P	0	0	1	0	0	0	0	0	0	0	0	0	
4+P	0	0	0	1	0	0	0	0	0	0	0	0	
5+P	0	0	0	0	1	0	0	0	0	0	0	0	
6+P	0	0	0	0	0	1	0	0	0	0	0	0	
7+P	0	0	0	0	0	0	1	0	0	0	0	0	
8+P	0	0	0	0	0	0	0	1	0	0	0	0	
9+P	0	0	0	0	0	0	0	0	1	0	0	0	
10+P	0	0	0	0	0	0	0	0	0	1	0	0	
11+P	0	0	0	0	0	0	0	0	0	0	1	0	
12+P	0	0	0	0	0	0	0	0	0	0	0	1	
<b>P= 72</b>	<b>Chance Percepção</b>												
input	ALL			ÁUDIO				VISUAL			WITHOUT		
1+P	1			0				0			0		
4+P	0			1				0			0		
8+P	0			0				1			0		
12+P	0			0				0			1		
<b>P= 96</b>	<b>Risco</b>												
input	Mínimo			Baixo				Médio			Crítico		
1+P	1			0				0			0		
4+P	0			1				0			0		
8+P	0			0				1			0		
12+P	0			0				0			1		

Tabela 19 – Treino 3: diferenciando classe por pesos.

Por fim, foi realizado o teste desbalanceado de seis modos diferentes, a fim de testar e garantir uma independência entre a categorização e o desbalanceamento linear. A intenção é verificar se estes desbalanceamentos influenciam na classificação ou apenas ajudam na generalização. Os experimentos apontaram igualdade de resultados em todos os 6 testes e a influência da adoção do desbalanceamento pode ser vista no item 4.5.1

Considerando o peso somador P da tabela 19 temos na tabela 20 como ficou a distribuição dos pesos para cada teste, a fim de validar o desbalanceamento.

Testes	Pesos	Percepção Útil	Distração	Direção Carro	Chance Percepção	Risco
teste 1	Peso =	0	24	48	72	96
teste 2	Peso =	24	0	72	48	96
teste 3	Peso =	48	96	0	24	72
teste 4	Peso =	72	24	96	0	48
teste 5	Peso =	96	48	24	72	0
teste 6	Peso =	96	24	0	72	48

Tabela 20 – Pesos de categorização.

#### 4.4.3 Escolha da Rede Neural Probabilística

A rede neural probabilística (RNP) é uma rede usada também para classificação, sendo baseada em uma variação da rede RBF que assim com a última tem uma forma de treinamento consideravelmente rápida por não ter o treinamento iterativo. Na RNP há o uso da estratégia de decisão Bayesiana para classificar os vetores de entrada (SKAPURA; GORDON, 1996) e os limites de decisão se aproximam dos limites ótimos de uma decisão Bayesiana quando o número de exemplos de treinamento cresce (SPECHT, 1990). Também o fato de envolver e mesclar algoritmos de rede RBF conjuntamente com algoritmos de probabilidade torna-o um algoritmo atraente para comparação neste trabalho.

#### 4.4.4 Treinamentos da Rede Neural Probabilística

Para o treinamento da rede RNP foi usada a implementação do algoritmo da biblioteca Neupy. Assim como descrito para a rede RBF o algoritmo de treinamento da RNP não é iterativo, mas neste caso há a necessidade de encontrar um ponto ótimo para o fator de

alisamento, que neste caso é o desvio padrão, onde a acurácia do algoritmo é máxima. Esse ajuste que deve ser encontrado está relacionado com as características intrínsecas das distribuições dos dados, ou seja, esse ajuste denominado fator de alisamento pode ser compreendido no exemplo da imagem da figura 19, assim vê-se que o fator de alisamento ajusta as distribuições para uma melhor estimativa da probabilidade de densidade. A estratégia é encontrar um valor ótimo para este conjunto de dados.

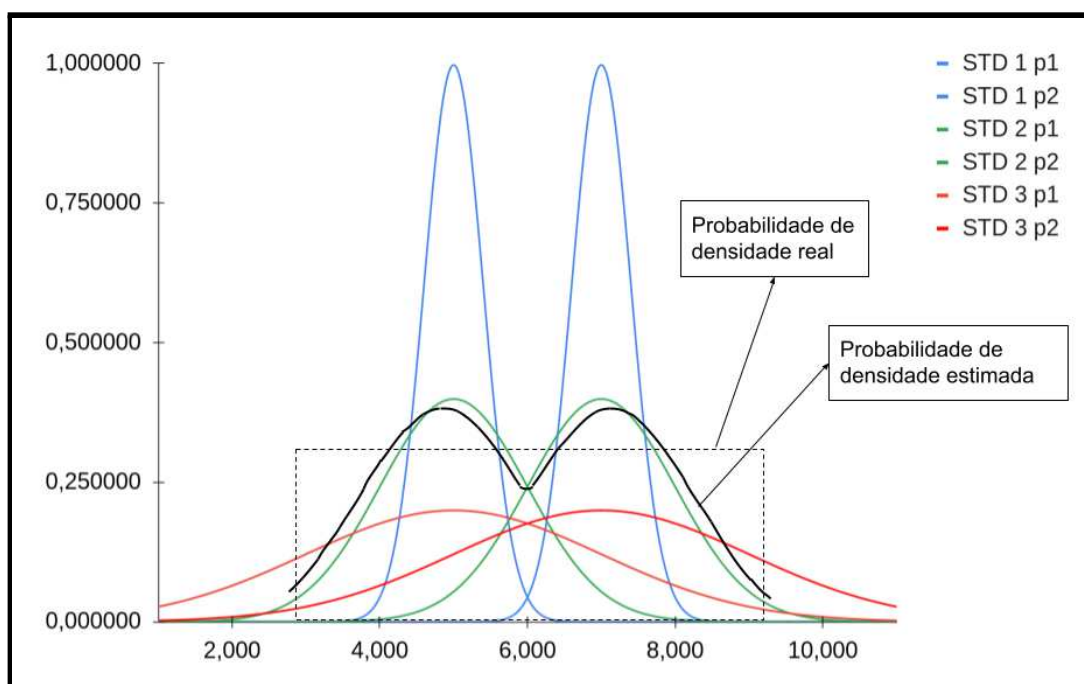


Figura 19 – Fator de alisamento para diferentes valores de desvios padrões

Os dados de entrada seguem um critério semelhante ao explanado no item 4.4.2, onde os dados foram sorteados de forma aleatória para evitar que a rede fosse treinada e influenciada pela sequência ordenada inicial.

A tabela 21 apresenta o exemplo dos dados sequenciados de forma aleatória seguindo o modelo do item já citado em 4.4.2. Mas neste caso o valor de referência das classes estão todos unidos em uma única coluna, pois é uma exigência do algoritmo RNP usado na biblioteca Neupy.

IDs		Consciência	Inputs				
User	Car	Classe	Percepção Útil	Distração	Direção Carro	Chance Percepção	Risco
uid 10	Car 09	NORMAL	1	6	4	1	1

uid 10	Car 07	NORMAL	1	6	11	4	4
uid 06	Car 08	PERDIDA	12	1	7	8	8
...	...	...	...	...	...	...	...
uid 06	Car 00	IMEDIATA	1	12	12	1	12
uid 16	Car 00	IMEDIATA	1	12	12	1	12
uid 18	Car 04	RISCO	6	1	8	4	1
uid 14	Car 06	NORMAL	1	1	8	12	4
uid 11	Car 01	RISCO	6	6	3	12	8
...	...	...	...	...	...	...	...
uid 15	Car 03	RISCO	6	12	10	12	8
uid 09	Car 10	IMEDIATA	1	12	9	1	8

Tabela 21 – Dados de treinamento para RNP.

## 4.5 RESULTADOS DAS REDES NEURAIIS

### 4.5.1 Resultados da rede RBF

Os resultados dos testes realizados para os três tipos de experimentos são apresentados na figuras 20, 21 e 22, pode-se ver que o formato dos dados desbalanceados apresentam uma performance melhor na aproximação da classificação dos dados, enquanto que o pior resultado é a estratégia onde todos os valores foram categorizados binariamente.

As figuras apresentam o comportamento da rede quando usamos os dados de treinamento e os dados de testes. Observa-se uma consistência na generalização da rede.

Na prática para o experimento foi testado uma faixa de 1 até 199 núcleos, mas essa quantidade de núcleos é uma extrapolação demasiada para a quantidade de amostras utilizadas no treino. Logo, uma faixa praticável seria 16 a 30 núcleos, o que apresenta uma acurácia de aproximadamente 79% para cerca de 20 núcleos no melhor experimento 22. Observa-se que na tabela 19 do item 4.4.2, no teste desbalanceado de seis modos diferentes todos os resultados são consistentes com o gráfico de 22 para os seis inputs diferentes da tabela 20.



Figura 20 – Treino 1: desempenho dos dados teste X treino.

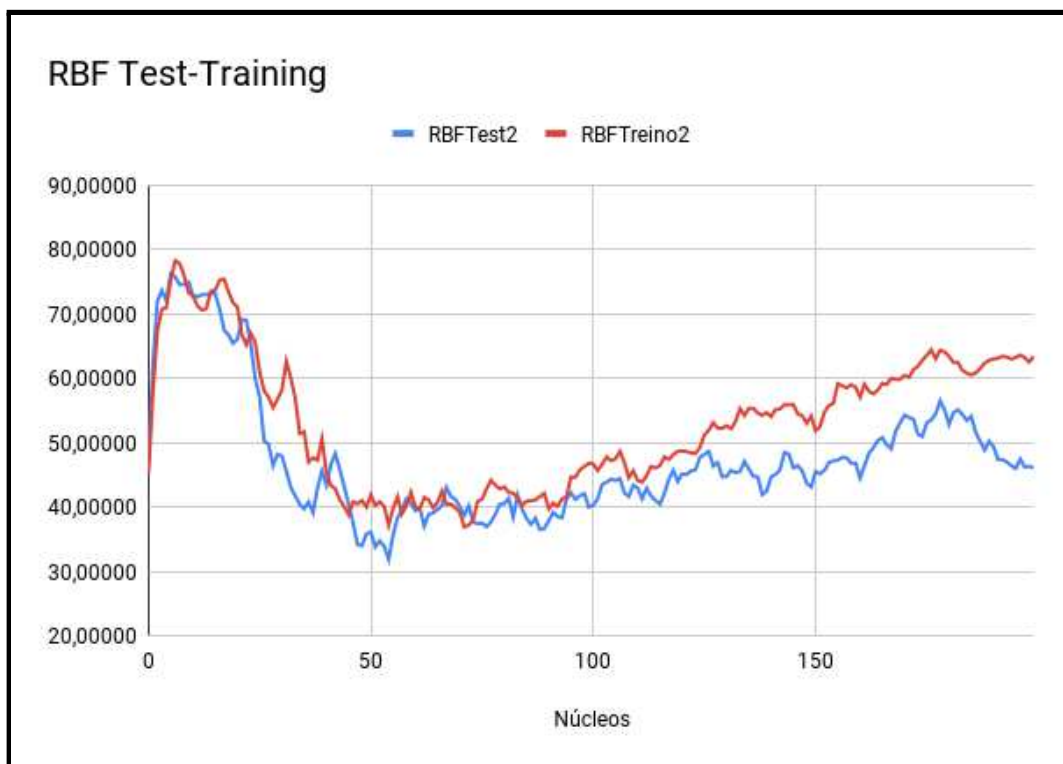


Figura 21 – Treino 2: desempenho dos dados teste X treino.



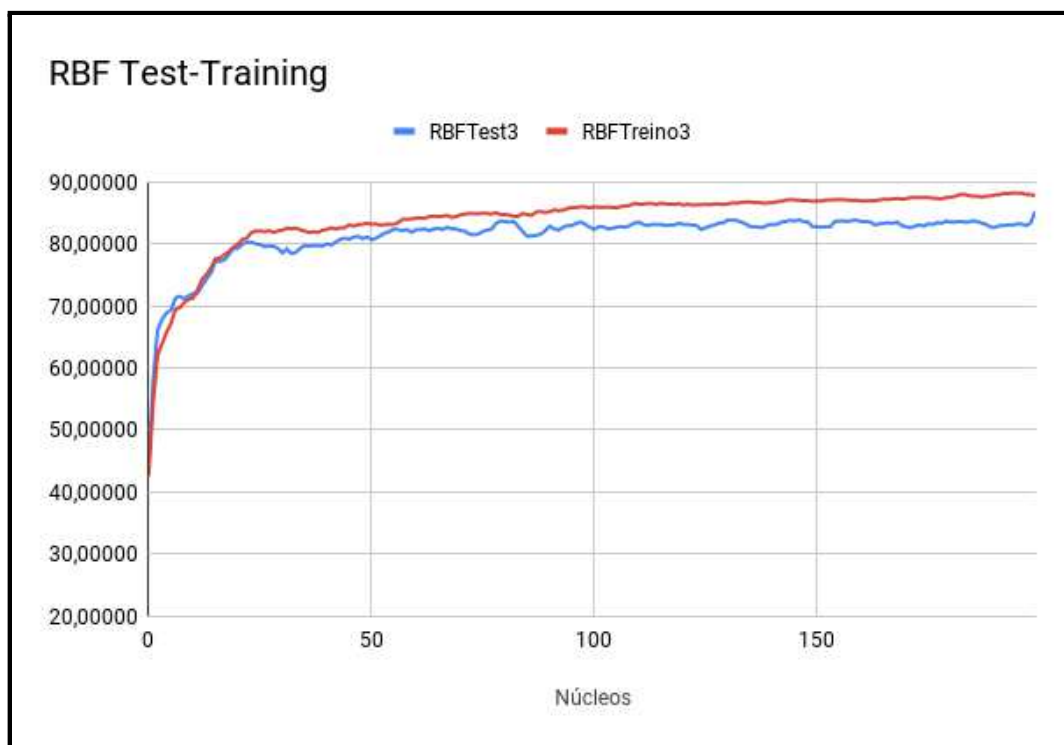


Figura 22 – Treino 3: desempenho dos dados teste X treino.

Ao analisar os conjuntos dos dados, obtém-se uma matriz de confusão em que a rede RBF tem um *F1 Score* muito baixo na identificação da classe NORMAL, pois a rede identifica a classe NORMAL como uma classe IMEDIATA na maioria dos casos. O *F1 Score* do RISCO é o mais alto para esta rede na classificação e o *RECALL* é 100% para a classificação PERDIDA.

MATRIZ CONFUSÃO						
RBF		PREDITA				FALSO NEGATIVO
		IMEDIATA	NORMAL	RISCO	PERDIDA	
REAL	IMEDIATA	56	11	0	0	11
	NORMAL	27	11	1	0	28
	RISCO	0	0	74	6	6
	PERDIDA	0	0	0	31	0
FALSO POSITIVO		27	11	1	6	45
AMOSTRAS	217					
ACURÁCIA	79,26%					
PRECISÃO	79,26%	67,47%	50,00%	98,67%	83,78%	
RECALL	79,26%	83,58%	28,21%	92,50%	100,00%	

F1 Score	79,26%	74,67%	36,07%	95,48%	91,18%	
----------	--------	--------	--------	--------	--------	--

Tabela 22 – Matriz Confusão rede RBF.

#### 4.5.2 Resultados da RNP

A estratégia para encontrar o fator de alisamento foi treinar a rede 400 vezes para prever o conjunto de teste e 400 vezes para prever o conjunto de treino, assim progressivamente foi-se variando os valores do fator para mapear a acurácia da saída da rede.

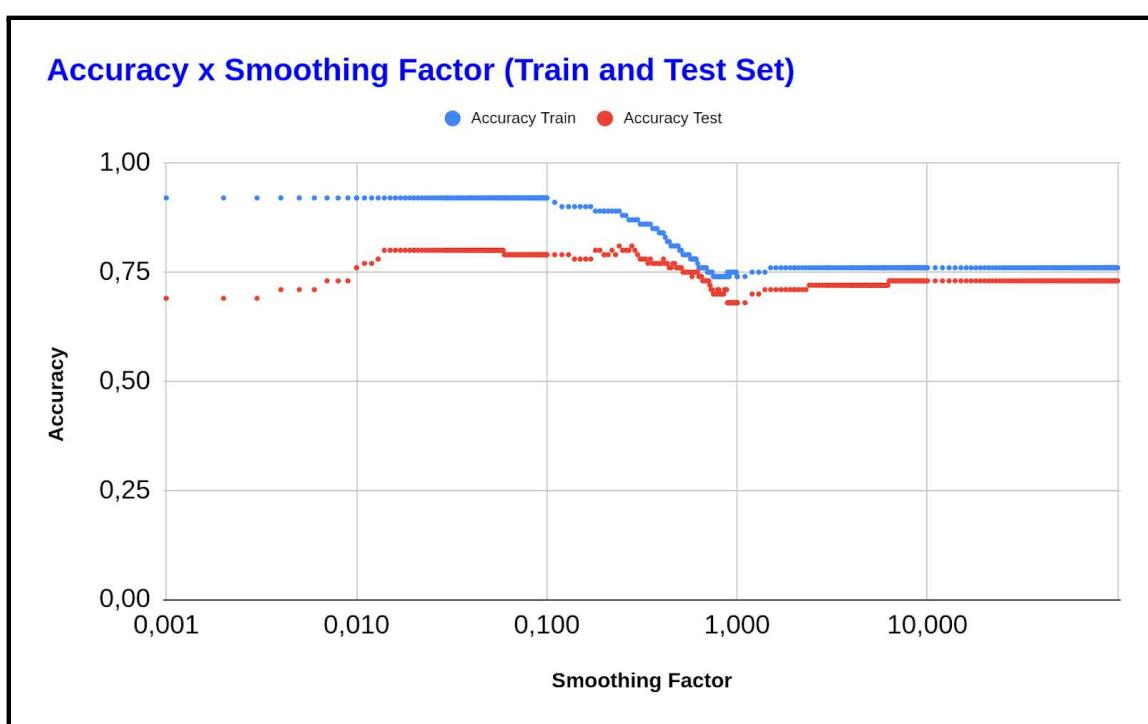


Figura 23 – Fator de alisamento para diferentes valores de desvios padrões

A figura 23 representa a relação entre acurácia e o fator de alisamento para os dois conjuntos utilizados na rede, onde o eixo do fator está representado em forma logarítmica. Os valores ótimos para este conjunto de treino estão na faixa aproximada entre 0,2 a 0,6. Com esses valores o algoritmo alcançou uma acurácia de aproximadamente 80%.

Para a matriz de confusão da rede RNP, observa-se que ela tem uma dificuldade em distinguir entre a classe IMEDIATA e NORMAL. A rede RBF e a RNP têm o mesmo desempenho quando olhamos o F1 Score do RISCO e a RNP uma vantagem 2% a mais que a rede RBF quando olhamos o F1 Score da classe PERDIDA.

MATRIZ CONFUSÃO						
RNP		PREDITA				FALSO NEGATIVO
		IMEDIATA	NORMAL	RISCO	PERDIDA	
REAL	IMEDIATA	45	22	0	0	22
	NORMAL	14	24	1	0	15
	RISCO	0	2	74	4	6
	PERDIDA	0	0	0	31	0
FALSO POSITIVO		14	24	1	4	43
AMOSTRAS	217					
ACURÁCIA	80,18%					
PRECISÃO	80,18%	76,27%	50,00%	98,67%	88,57%	
RECALL	80,18%	67,16%	61,54%	92,50%	100,00%	
F1 Score	80,18%	71,43%	55,17%	95,48%	93,94%	

Tabela 23 – Matriz Confusão RNP.

## 4.6 ANÁLISE DO MODELO BAYESIANO

### 4.6.1 Escolha da Rede Bayesiana

Como apontado na seção anterior, o modelo proposto por (PEREIRA, 2017) foi criado como uma tentativa de ser um modelo de consciência situacional, com base em RB, utilizando aprendizado por reforço para que um agente pudesse melhorar os resultados da probabilidade condicional a priori do seu modelo.

Baseado no trabalho de (PEREIRA, 2017), mas sem a preocupação de usar um agente para melhorar os resultados da PC a priori do modelo deste trabalho, aplicou-se os algoritmos e bibliotecas utilizadas por (PEREIRA, 2017) para gerar uma RB relacionada com as definições deste trabalho, assim objetivou-se comparar o desempenho da PC da RB para a consciência situacional com relação aos modelos de redes neurais deste trabalho já citadas em seções anteriores.

#### 4.6.2 Treinamento da Rede Bayesiana

Foi utilizada a linguagem de *script python* com a biblioteca de *pandas* para ler os dados e a biblioteca *pgmpy* para criação do modelo, inferência e estimação bayesiana. A estrutura dos dados pode ser vista na tabela 24 e seguem a mesma definição aplicada às redes neurais, ou seja, tem a mesma ordem de embaralhamento de dados.

IDs		Consciência	Inputs				
User	Car	Classe	Percepção Útil	Distração	Direção Carro	Chance Percepção	Risco
uid 10	Car 09	NORMAL	PRESENTE	SOFF	WF	ALL	MÍNIMO
uid 10	Car 07	NORMAL	PRESENTE	SOFF	WFL	ÁUDIO	BAIXO
uid 06	Car 08	PERDIDA	AUSENTE	SON	BR	VISUAL	MÉDIO
...	...	...	...	...	...	...	...
uid 06	Car 00	IMEDIATA	PRESENTE	NONE	WB	ALL	CRÍTICO
uid 16	Car 00	IMEDIATA	PRESENTE	NONE	WB	ALL	CRÍTICO
uid 18	Car 04	RISCO	TARDIA	SON	WFR	ÁUDIO	MÍNIMO
uid 14	Car 06	NORMAL	PRESENTE	SON	WFR	NONE	BAIXO
uid 11	Car 01	RISCO	TARDIA	SOFF	FR	NONE	MÉDIO
...	...	...	...	...	...	...	...
uid 15	Car 03	RISCO	TARDIA	NONE	WBL	NONE	MÉDIO
uid 09	Car 10	IMEDIATA	PRESENTE	NONE	WBR	ALL	MÉDIO

Tabela 24 – Dados de entrada Rede Bayesiana.

Os dados foram divididos em 70% para geração do modelo e levantamento do valores de probabilidade condicional a priori e os outros 30% restantes foram usados para testar o modelo. O modelo criado é representado na figura 24 através do *software Netica* expressando os devidos valores de probabilidade condicional a priori. A partir deste modelo é possível inferir quais seriam os níveis de consciência situacional para o pedestre em determinada situação apresentada.

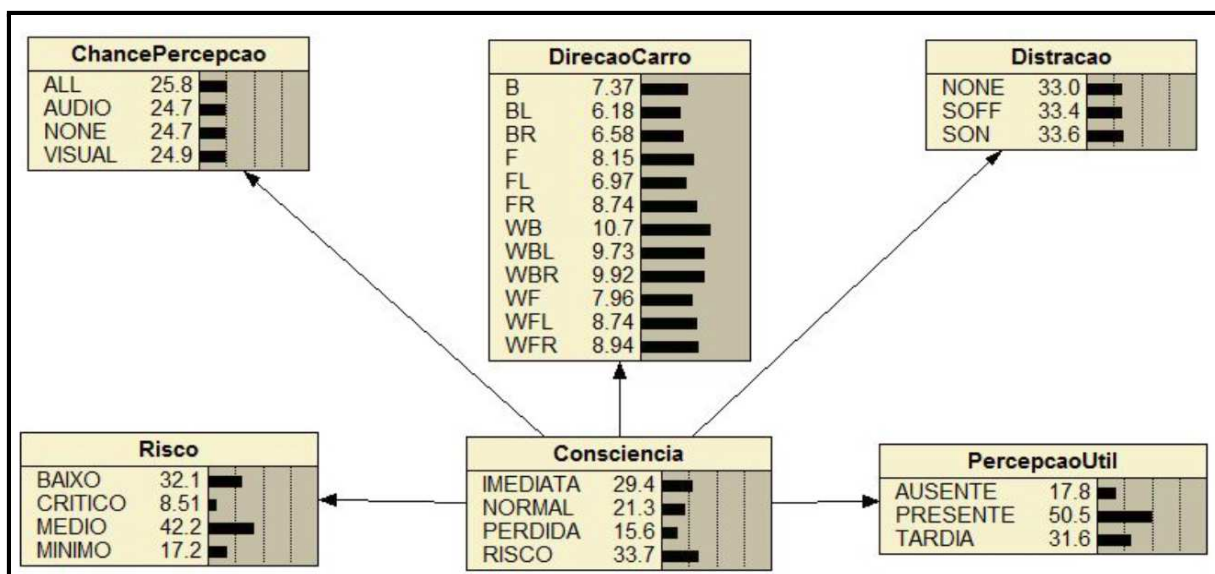


Figura 24 – Fator de alisamento para diferentes valores de desvios padrões

Para o caso da figura 25, onde percepção útil é ausente; o nível de distração é médio, ou seja, tem um nível de distração visual e não auditivo; a direção do carro é *warning back*, significando que o carro vem por trás e o pedestre está na mesma faixa que o carro; a chance de percepção é visual, assim considerando que os únicos sentidos úteis são os olhos para percepção do carro; o risco em nível médio, explicitando que há alguma experiência do pedestre com o ambiente, temos que o modelo RB aponta 99,7% para um caso de consciência perdida, o que na prática seria um possível atropelamento no trânsito.

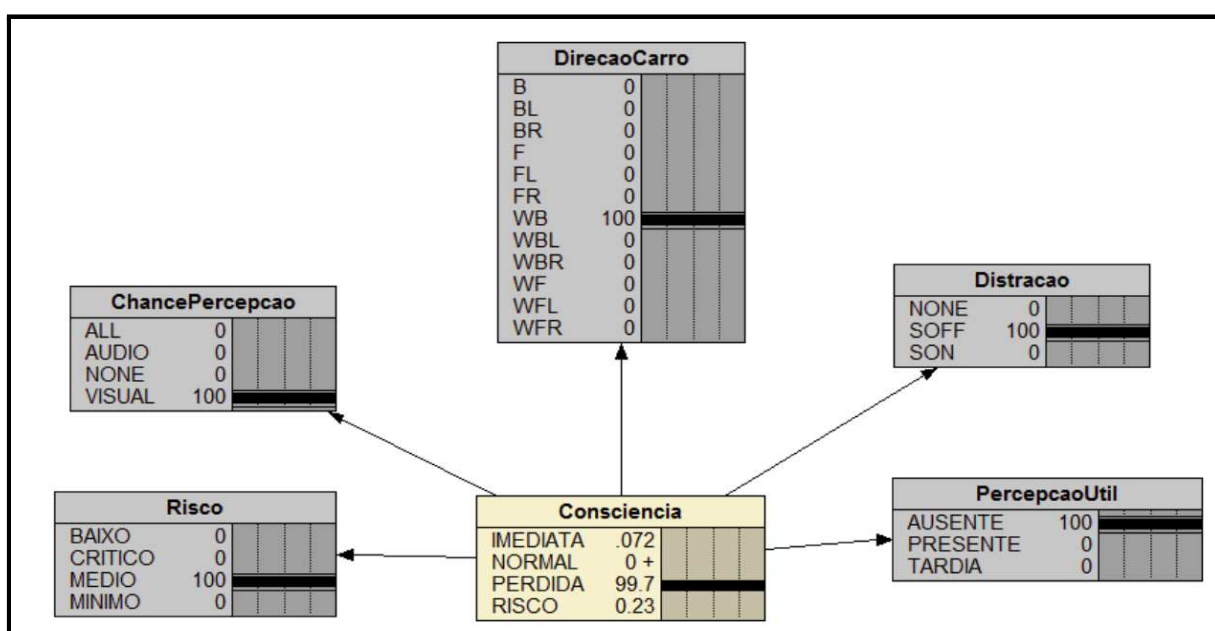


Figura 25 – Fator de alisamento para diferentes valores de desvios padrões

## 4.7 RESULTADO DA REDE BAYESIANA

### 4.7.1 Resultados da RB

Utilizando-se os 30% da base de teste destinada para verificação do modelo, gerou-se para cada tupla de entrada <Distração, DireçãoCarro, ChancePercepção, PercepçãoUtil, Risco> uma verificação da probabilidade condicional que é ilustrado na tabela 25 com o caso <NONE, F, VISUAL, PRESENTE, BAIXO>, assim em cada tupla de entrada o modelo entregou sua probabilidade de ocorrência.

Consciência	Probabilidade
IMEDIATA	0,8061
NORMAL	0,1929
PERDIDA	0,0009
RISCO	0,0000

Tabela 25 – Exemplo de saída para uma entrada na RB.

As saídas de probabilidade entregues pelo modelo foram comparadas com o valor real da base de dados (base de teste) em cada tupla para obter a assertividade do modelo, portanto usou-se a matriz de confusão como avaliação da RB, assim como para a rede RBF e a RNP, ver tabela 26. O *F1 Score* da RB ainda é baixo para a classe NORMAL, mas tem um desempenho melhor que os das redes neurais. Para a classe RISCO o *F1 Score* é tão alto quanto o das redes neurais, porém chega a ser 8% menor quando comparada a RNP para a classe PERDIDA.

RB		PREDITA				FALSO NEGATIVO
		IMEDIATA	NORMAL	RISCO	PERDIDA	
REAL	IMEDIATA	48	19	0	0	19
	NORMAL	15	24	0	0	15
	RISCO	0	0	75	5	5
	PERDIDA	0	0	4	27	4

<b>FALSO POSITIVO</b>		15	19	4	5	43
<b>AMOSTRAS</b>	217					
<b>ACURÁCIA</b>	80,18%					
<b>PRECISÃO</b>	80,18%	76,19%	55,81%	94,94%	84,38%	
<b>RECALL</b>	80,18%	71,64%	61,54%	93,75%	87,10%	
<b>F1 Score</b>	80,18%	73,85%	58,54%	94,34%	85,71%	

Tabela 26 – Matriz Confusão RB.

#### 4.8 ANÁLISE COMPARATIVA DOS MODELOS

Uma análise entre os algoritmos é necessária para evidenciar o desempenho destes usando a base de dados descrita neste trabalho, e para isso foi produzida uma análise baseada no desempenho da curva ROC (*Receiver Operating Characteristic*) e a sua área de desempenho AUC (*area under the ROC curve*). Quando olha-se para a análise ROC/AUC objetiva-se identificar uma diferença no desempenho desses algoritmos para cada classe, pois a acurácia para a rede RBF igual 79,26%, para RNP 80,18% e para a RN 80,18% que observa-se através da matriz de confusão em cada experimento não traz uma boa diferenciação. Temos na verdade um empate no desempenho quando olha-se apenas a acurácia.

As figuras 26, 27 e 28 mostram a curva ROC/AUC para cada classe usando o algoritmo da rede RBF, RNP, RN respectivamente, sendo assim os valores de AUC observados são:

- Na rede RBF são de 0,81 para IMEDIATA, 0,66 para NORMAL, 0,96 para RISCO e 0,98 para PERDIDA;
- Na RNP são de 0,79 para IMEDIATA, 0,74 para NORMAL, 0,96 para RISCO e 0,99 para PERDIDA;
- Na RB são de 0,81 para IMEDIATA, 0,75 para NORMAL, 0,95 para RISCO e 0,92 para PERDIDA.

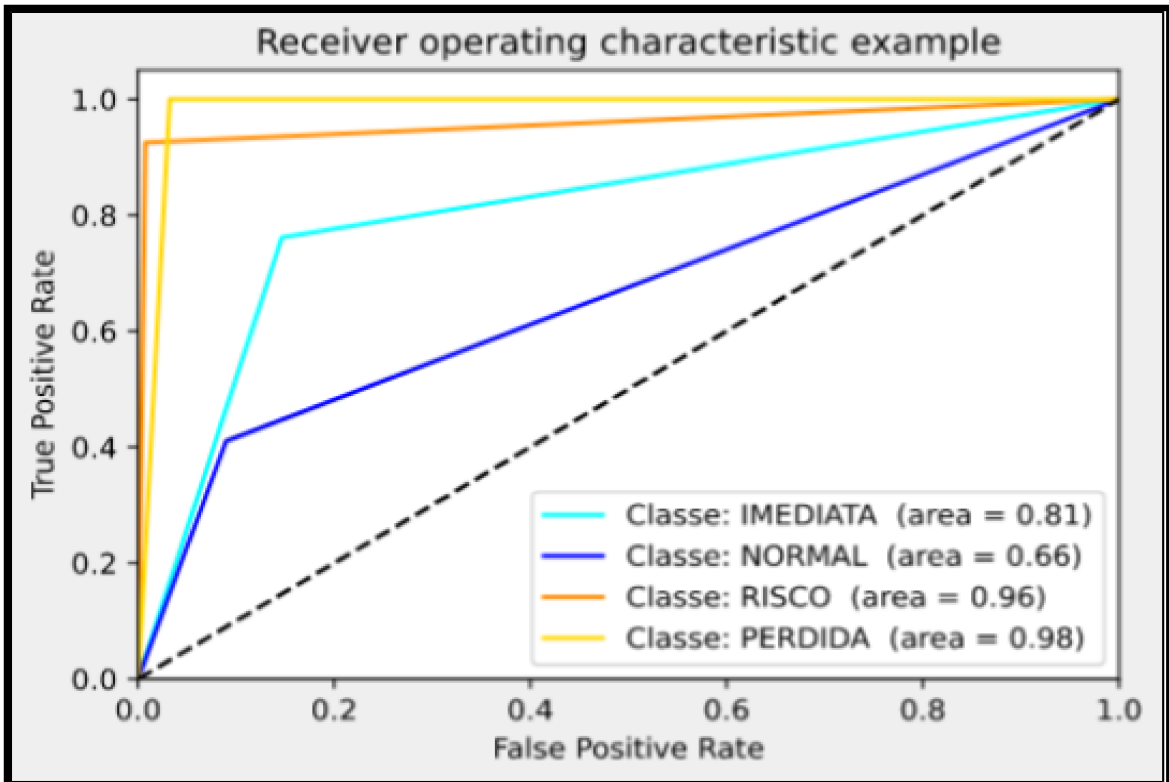


Figura 26 – ROC/AUC para o algoritmo da rede RBF

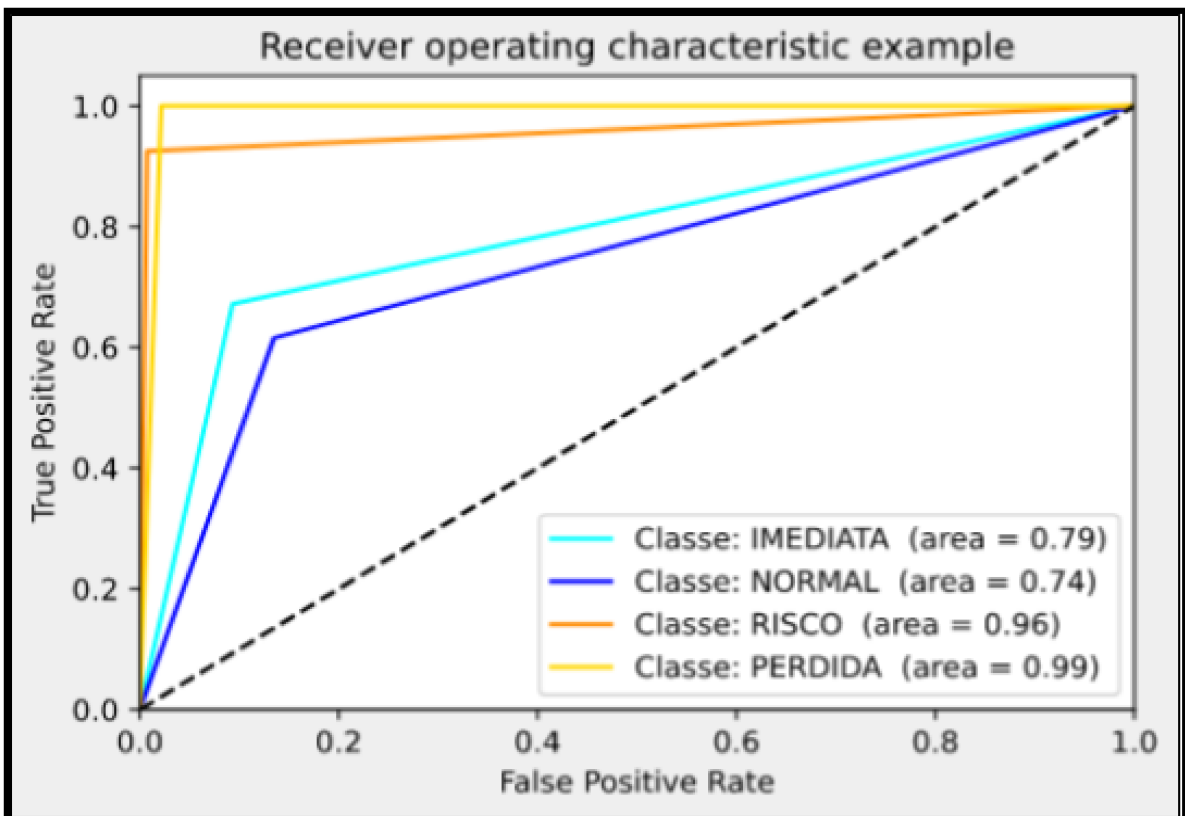


Figura 27 – ROC/AUC para o algoritmo RPN



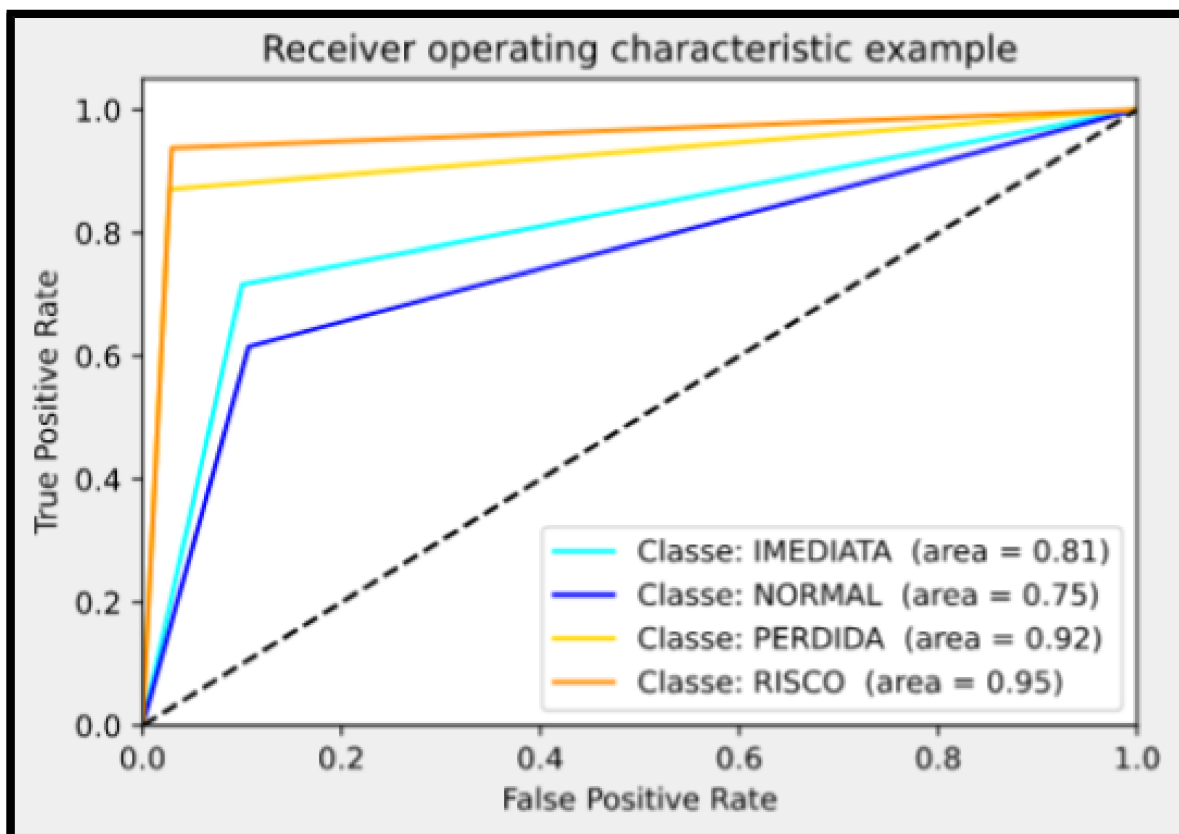


Figura 28 – ROC/AUC para o algoritmo RB

Normalmente a curva ROC utiliza uma variação de *threshold*, mas no caso dessa análise o *threshold* é fixo, pois buscou-se o ponto ótimo para cada algoritmo como descrito nas análises de cada tópico acima. Por isso, a intenção da AUC nesta análise é encontrar uma evidência do desempenho nas comparações dos algoritmos.

## 5 CONCLUSÃO E TRABALHOS FUTUROS

Com a realização dos experimentos e a avaliação dos resultados obtidos, conclui-se que os objetivos específicos apontados neste trabalho foram alcançados. O objetivo 1 foi alcançado na fundamentação teórica ao descrever o funcionamento das redes neurais artificiais em seu modelo mais básico como o perceptron, assim como as redes utilizadas neste trabalho, a rede RBF e a RNP.

Para o modelo de consciência estabelecido neste trabalho, o pedestre tem quatro estados de consciência possíveis, sendo elas consciência IMEDIATA, NORMAL, RISCO, PERDIDA, assim o critério para avaliação do modelo de consciência situacional foi definido a partir da análise da base de dados usada, juntamente com os levantamentos feitos no item 4.1, 4.2 e 4.3 deste trabalho cumprindo o objetivo específico 2.

Diante do requisito do modelo para consciência situacional e o objetivo de comparação das redes neurais com o modelo de rede bayesiana, procurou-se algoritmos que pudessem superar o desempenho de uma rede bayesiana. O item 4.4 descreve os algoritmos escolhidos e que cumprem com o objetivo específico 3.

Nos itens 4.5.1, 4.5.2 e 4.8, obteve-se os resultados analíticos sobre as redes neurais para alcançar o objetivo específico 4. Utilizou-se a matriz de confusão, acurácia, precisão, *recall* e *f1 score* para verificar o desempenho dos algoritmos. O objetivo específico 5 cumpre-se no item 4.4 e 4.6. O objetivo específico 6 é realizado em 4.3.

Através da pesquisa feita e o estudo realizado em todo item 4 e especificamente através dos itens 4.5.1, 4.5.2, 4.7.1 e 4.8 pode-se afirmar que o objetivo 7 deste trabalho também foi realizado.

Sobre os resultados alcançados pode-se dizer que ao considerarmos o nível de importância das classes para segurança do pedestre em relação a sua consciência situacional, temos a seguinte ordem: PERDIDA, RISCO, NORMAL e IMEDIATA. Assim, quando analisa-se os resultados alcançados podemos organizar o desempenho dos algoritmos em: RNP, rede RBF e RB.

Todos os algoritmos avaliados tiveram um baixo desempenho em separar os eventos da classe NORMAL dos eventos da classe IMEDIATA. Uma suposição para tal dificuldade pode ter ocorrido devido às suposições feitas em 4.3.6, pois para a separação destas classes de consciência adotou-se uma definição baseada na média do tempo, que pode não encontrar embasamento real nas características apontadas para o treinamento.

## 5.1 TRABALHOS FUTUROS

A partir do trabalho de pesquisa e desenvolvimento aqui realizado, pode-se observar as oportunidades de trabalhos a serem desenvolvidos futuramente, como por exemplo:

- Estudo, implementação e comparação entre essas redes usando outra base de dados principalmente quando usa-se RNP e os dados requerem uma análise mais probabilística das relações entre as classes;

- Para o caso de uso desta mesma base de dados as pesquisas poderão focar no estudo e implementação de novas redes neurais ampliando as abordagens já colocada neste trabalho;

- Ainda para uso da mesma base de dados, pode-se manipular algumas características dos dados para alcançar uma melhor definição na divisão das classes, pois houve uma baixa performance em separar as classes IMEDIATA e NORMAL neste trabalho e que supõe-se ocorrer devido às suposições feitas em 4.3.6, pois neste caso pode ter faltado características suficientes de definição de cada classe para que os algoritmos evidenciam uma separação;

- Uso do estudo para alavancar novas pesquisas em relação ao monitoramento da consciência do pedestre, pois a grande questão neste trabalho é identificar a consciência do pedestre. Pois foi demonstrado no estudo para identificar a consciência útil, dentro dos parâmetros estabelecidos neste trabalho em 4.3.1, que uma taxa de movimento baixa eleva consideravelmente a consciência tardia. Neste caso, uma taxa lenta pode ser um estudo sobre pessoas deficientes, idosos, mulheres com crianças pequenas entre outros;

- Coletar dados com o uso de outras aplicações e equipamentos para monitorar a consciência. Ao tomar a situação dos parágrafos acima, pode-se elaborar meios de medir a forma e a velocidade de deslocamento do usuário, pode-se usar sensores externos em um laboratório virtual como o que gerou os dados deste trabalho, ou equipamentos para contar os passos, monitorar batimentos cardíacos, pressão arterial e dados biomédicos em geral para saber se em nível de inconsciência a algum processamento biológico do perigo antes da parte consciente. Assim, poder-se-ia gerar novos modelos para continuidade das pesquisas de consciência situacional de pedestres;

- Usar algoritmos para monitorar a consciência situacional de motoristas de veículos. Apesar de não ser o caso dessa base de dados, seria um bom estudo para consciência situacional a adição do comportamento dos motoristas com relação a situação consciência do pedestre. Por exemplo, uma questão a ser respondida seria: Quanto à inconsciência do motorista em relação ao pedestre afeta o tempo de consciência IMEDIATA do pedestre em relação a sua taxa de movimentação?

## REFERÊNCIA

CHEN, Y.; CHANG, H.; MENG, J.; ZHANG, D.; *Ensemble Neural Networks (ENN): A gradient-free stochastic method*, Neural Networks, Vol. 110, 2019, P. 170-185, ISSN 0893-6080, <http://www.sciencedirect.com/science/article/pii/S0893608018303319>, 2018

COPPIN, B. *Inteligência Artificial*. Rio de Janeiro, Brasil: LTC, 2013. 636 p.

EL\_JERJAWI, N. S.; ABU-NASER, S. S.; *Diabetes Prediction Using Artificial Neural Network*. International Journal of Advanced Science and Technology, 124, 1-10, 2018

EMARY IBRAHIEM; SRINIVASAN RAMAKRISHNAN. *On the application of various probabilistic neural networks in solving different pattern classification problems*. World Applied Sciences Journal. 4. 772-780. (2008).

ENDSLEY, MICA. *Toward a Theory of Situation Awareness in Dynamic Systems*. Human Factors: The Journal of the Human Factors and Ergonomics Society. 37. 32-64. 10.1518/001872095779049543, (1995)

FEIS. Medida do tempo de reação humano-UNIVAP. Tópico 6: aula prática (2012). Disponível em: < [www.univap.br/aula.fisica.exp](http://www.univap.br/aula.fisica.exp) >. Acesso: 23/03/2021

GELAIM, T. A. et al. *A hybrid intelligent agent for notification of users distracted by mobile phones in an urban environment*. In: ICAART. [S.l.: s.n.], 2019.

HAYKIN, SIMON. *Redes Neurais Princípios e Práticas*. 2. ed. Bookman, 2001.

KOUROU K., EXARCHOS T.P., EXARCHOS K.P., KARAMOUZIS M.V., FOTIADIS D.I. *Machine learning applications in cancer prognosis and prediction* Comput. Struct. Biotechnol. J., 13 (2015), pp. 8-17, 10.1016/j.csbj.2014.11.005

LANGER, G. A. *Análise de dados aplicada para consciência situacional de pedestres* [S.l.]: Florianópolis, SC, 2018.

LIU, W.; WANG, Z.; LIU, X.; ZENG, N.; LIU, Y.; ALSAADI, F. *A survey of deep neural network architectures and their applications*. Neurocomputing. 234. 10.1016/j.neucom.2016.12.038, 2016.

LUGER, G. F. *Inteligência Artificial*. 6. ed. [S.l.]: Pearson Education, 2013.

PARAMITA B., Pragyani; REKHA D., Smruti; MISHRA, Debahuti. *A survey on exchange rate prediction using neural network based methods*. International Journal of Engineering & Technology, [S.l.], v. 7, n. 2.6, p. 64-70, mar. 2018. ISSN 2227-524X. 2020.

PEREIRA, R. L. *Aplicação de Aprendizagem por Reforço para um Modelo Bayesiano de Consciência Situacional*. Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, 2017.

SANTOS, E.; FAZENDA, B. *Computational Model of Situational Awareness for users of smartphones in the vicinity of traffic*. [S.l.], 2016.

SILVA, A. B.; J. R. CUNHA E M. C. FERREIRA. *Velocidade pedonal em atravessamentos – Avaliação da relevância dos fatores explicativos*. Anais do XXVI Congresso de Pesquisa e Ensino em Transportes, ANPET, Joinville, v. 1, p. 278–289. 2012.

SILVA, IVAN N.; SPATTI, DANILO H.; FLAUZINO, ROGÉRIO A.; *Redes Neurais Artificiais para engenharia e ciências aplicadas*, Artliber, 2010.

SKAPURA, D. M. GORDON, P. S. (Ed.), *Building Neural Networks*, ACM Press, (pp. 41-42), 1996.

SPECHT, D. F. *Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification*. IEE Transactions on Neural Networks, vol. 1, (pp. 111-121), 1990.

UNIVERSIDADE FEDERAL DE SANTA CATARINA. Biblioteca Universitária. *Trabalho acadêmico: guia fácil para diagramação*: formato A5. Florianópolis, 2009. Disponível em: <<http://www.bu.ufsc.br/design/GuiaRapido2012.pdf>>. Acesso em: 23 mar. 2021

ZEINALI, YASHA; STORY, BRETT A.. *Competitive probabilistic neural network*. Integrated Computer-Aided Engineering. 24 (2): 105–118. doi:10.3233/ICA-170540. (2017)

ZOU, T.; YANG, S.; ZHANG, Y.; YE, M.; *Attention guided neural network models for occluded pedestrian detection*, Pattern Recognition Letters, Volume 131, 2020, Pages 91-97, ISSN 0167-8655, <http://www.sciencedirect.com/science/article/pii/S0167865519303733>, 2019

## 6 ANEXOS

### 6.1 ANEXO A - CÓDIGO FONTE REDE RBF

```
#!/usr/bin/env python
# coding: utf-8
# importação

import numpy as np
from sklearn.cluster import KMeans

# class
class RBFNN:

    def __init__(self, inputs, outputs, tests, templates,
first_layer_size=3):
        self.inputs = inputs
        self.outputs = outputs
        self.tests = tests
        self.templates = templates
        self.neurons = first_layer_size
        self.sigma = 0
        self.centroids = []
        self.Weights = []
        self.outGen = []

    def makeCentroids(self):
        # initialize
        model = KMeans(n_clusters=self.neurons)
        # run
        model.fit(self.inputs)
        self.centroids = model.cluster_centers_

    def makeSigma(self):
        # initialize
        c = self.centroids
        # run
        if np.squeeze(np.asarray(c))[0].size > 1:
            points = np.asarray(c).reshape(1, -1)[0]
            self.sigma =
(max(points)-min(points))/np.sqrt(2*self.neurons)
```

```

else:
    sigma = (max(c)-min(c))/np.sqrt(2*self.neurons)
    self.sigma = sigma[0]

def generateMatrix(self, data, size):
    # initialize
    G = np.zeros((size, self.neurons))
    sigma = self.sigma
    cent = self.centroids
    # run
    for i in range(size):
        for j in range(self.neurons):
            dist = np.linalg.norm(data[i, :]-cent[j, :], 2)
            G[i, j] = np.exp((-1/(sigma**2)) * dist**2)

    return G

def setFirstLayerSize(self, size):
    self.neurons = size
    return self

def fit(self):
    # run
    self.makeCentroids()
    self.makeSigma()
    G = self.generateMatrix(self.inputs, self.inputs[:, 0].size)
    self.Weights = np.dot(np.linalg.pinv(G), self.outputs)
    return self

def predict(self):
    # initialize
    G = self.generateMatrix(self.tests, self.tests[:, 0].size)
    # run
    self.outGen = np.dot(G, self.Weights)
    return self

def showAccuray(self):
    y = np.array([np.argmax(x) for x in self.outGen])
    g = np.array([np.argmax(x) for x in self.templates])
    diff = y - g
    msg = 'Accuracy for first layer size {} is
:'.format(self.neurons)
    print(msg, len(np.where(diff == 0)[0]) / len(diff))

```

```

def getPredictedOutputs(self):
    return self.outGen

def setTest(self, tests):
    self.tests = tests
    return self

```

---

```

#!/usr/bin/env python
# coding: utf-8

# In[1]:
# importação
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from RBFNN import RBFNN as rbfnn

def normalize(x, axis=1):
    if axis == 0:
        x = x.T
        n = np.array([(x[:, n] - min(x[:, n])) / (max(x[:, n]) - min(x[:,
n]))
                    for n in range(len(x[0, :]))])
    if axis == 1:
        n = n.T
    return n

# In[2]:
# prepare
dt = pd.read_csv('./ShuffledDatabaseRNRBF.csv', sep=',', decimal=",")
dt.drop(['Consciencia', 'Car', 'User', 'count'], axis=1, inplace=True)
dt.head(10)

# In[3]:
# prepare
SPLIT = int(len(dt) * 0.7)
TRAIN = dt[:SPLIT].copy()
TEST = dt[SPLIT:].copy()

```



```

inputs = TRAIN[['Risco', 'Distracao', 'DirecaoCarro',
               'PercepcaoUtil', 'ChancePercepcao']].to_numpy()
inputs = normalize(inputs)
outputs = TRAIN[['IMEDIATA', 'NORMAL', 'RISCO', 'PERDIDA']].to_numpy()
tests = TEST[['Risco', 'Distracao', 'DirecaoCarro',
              'PercepcaoUtil', 'ChancePercepcao']].to_numpy()
tests = normalize(tests)
templates = TEST[['IMEDIATA', 'NORMAL', 'RISCO', 'PERDIDA']].to_numpy()

rbf = rbfnn(inputs, outputs, tests, templates)

# In[4]:
# run
for i in range(2, 199):
    rbf.setFirstLayerSize(i).fit().predict().showAccuray()

# Para 20 núcleos
for i in range(20, 21):
    rbf.setFirstLayerSize(i).fit().predict().showAccuray()

pred = rbf.getPredictedOutputs()
y_pred = np.array([np.argmax(x) for x in pred])
y_true = np.array([np.argmax(x) for x in templates])
ypred = label_binarize(y_pred, classes=[0,1,2,3])
ytrue = label_binarize(y_true, classes=[0,1,2,3])
n_classes = 4
fpr = dict()
tpr = dict()
roc_auc = dict()
precision = precision_score(y_true, y_pred, average=None)
recall = recall_score(y_true, y_pred, average=None)

for i in range(n_classes):
    fpr[i], tpr[i], t = roc_curve(ytrue[:, i], ypred[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
    print("roc_auc[{}] = {}, fpr{}, tpr{}".format(i, roc_auc[i], fpr[i],
tpr[i]))

print("precision = {}".format(precision))
print("recall = {}".format(recall))

# In[5]:

```

```

Labels = ['IMEDIATA', 'NORMAL', 'RISCO', 'PERDIDA']
colors = cycle(['aqua', 'blue', 'darkorange', 'gold'])
plt.figure()
for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color, label='Classe: {0} (area =
{1:0.2f})'.format(Labels[i], roc_auc[i]))
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()

```

## 6.2 ANEXO B - CÓDIGO FONTE REDE RNP

```

#!/usr/bin/env python
# coding: utf-8
# In[1]:
# importação
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc, recall_score,
precision_score
from sklearn.preprocessing import label_binarize
from itertools import cycle
from neupy.algorithms import PNN

# In[2]:
def normalize(x, axis=1):
    if axis == 0:
        x = x.T
        n = np.array([(x[:, n] - min(x[:, n])) / (max(x[:, n]) - min(x[:,
n]))
                    for n in range(len(x[0, :]))])
    if axis == 1:
        n = n.T
    return n

```

```

# In[3]:
# prepare
dt = pd.read_csv('ShuffledDatabasePNN2.csv', sep=',', decimal=",")
dt.head(10)

SPLIT = int(len(dt) * 0.7)
TRAIN = dt[:SPLIT].copy()
TEST = dt[SPLIT:].copy()
# In[4]:
# prepare

inputs = TRAIN[['Risco', 'Distracao', 'DirecaoCarro',
                'PercepcaoUtil', 'ChancePercepcao']].to_numpy()

inputs = normalize(inputs)
outputs = TRAIN[['Consciencia']].to_numpy()

tests = TEST[['Risco', 'Distracao', 'DirecaoCarro',
              'PercepcaoUtil', 'ChancePercepcao']].to_numpy()

tests = normalize(tests)
templates = TEST[['Consciencia']].to_numpy()

# In[5]:
# run
initial = 50
result = []
print("> Start classify")
# descomentar e variar std para Accuracy x Standard Deviation
for i in range(initial, initial+1):
    std = i/1000
    pnn_network = PNN(std=std, verbose=False)
    pnn_network.train(inputs, outputs)
    result = pnn_network.predict(tests)

    n_correctly = np.sum(np.where(result == templates.T, 1, 0))
    n_samples = templates.size

    # print("Test #{:<2}: Guessed {} out of {} {}".format(
    #     1, n_correctly, n_samples, (n_correctly/n_samples).round(2)))
    #     print("Metrics accuracy =
    {}".format(metrics.accuracy_score(result, templates)))

```

```

        print("std:      {}      acc:      {}".format(std,
(n_correctly/n_samples).round(2)))

# In[6]
ypred = label_binarize(result,
classes=["IMEDIATA", "NORMAL", "RISCO", "PERDIDA"])
ytrue = label_binarize(templates.T[0],
classes=["IMEDIATA", "NORMAL", "RISCO", "PERDIDA"])
n_classes = 4

y_pred = np.array([np.argmax(x) for x in ypred])
y_true = np.array([np.argmax(x) for x in ytrue])

fpr = dict()
tpr = dict()
roc_auc = dict()

precision = precision_score(y_true, y_pred, average=None)

recall = recall_score(y_true, y_pred, average=None)

for i in range(n_classes):
    fpr[i], tpr[i], t = roc_curve(ytrue[:, i], ypred[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
    print("roc_auc[{}] = {}, fpr{}, tpr{}".format(i, roc_auc[i], fpr[i],
tpr[i]))

print("precision = {}".format(precision))
print("recall = {}".format(recall))
# In[7]:
Labels = ['IMEDIATA', 'NORMAL', 'RISCO', 'PERDIDA']
colors = cycle(['aqua', 'blue', 'darkorange', 'gold'])
plt.figure()
for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color, label='Classe: {0} (area =
{1:0.2f})'.format(Labels[i], roc_auc[i]))
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")

```

```
plt.show()
```

### 6.3 ANEXO C - CÓDIGO FONTE REDE RB

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:
# importação
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from pgmpy.models import BayesianModel
from pgmpy.inference import BeliefPropagation
from pgmpy.estimators import BayesianEstimator
from sklearn.metrics import roc_curve, auc, recall_score,
precision_score
from sklearn.preprocessing import label_binarize
from itertools import cycle

# In[2]:
# lê a base de dados
dataset = pd.read_csv('ShuffledDatabase2.csv', sep=',')
dataset.head(26)

# In[3]:
dataset.drop(['User', 'Car'], axis=1, inplace=True)
dataset.head(26)

# In[4]:
# separar elementos em conjunto de teste e treino
SPLIT = int(len(dataset) * 0.7)
TRAIN = dataset[:SPLIT].copy()
TEST = dataset[SPLIT:].copy()

# In[5]:
# mostra treino
TRAIN.head()

# In[6]:
# mostra teste
TEST.head()
```

```

# In[7]:
model_teste = BayesianModel([
    ('Consciencia', 'Distracao'),
    ('Consciencia', 'DirecaoCarro'),
    ('Consciencia', 'ChancePercepcao'),
    ('Consciencia', 'PercepcaoUtil'),
    ('Consciencia', 'Risco')
])

model_teste.fit(TRAIN, estimator=BayesianEstimator)
for node in model_teste.get_cpds():
    print(node.variable)
    print(list(zip(node.state_names[node.variable], node.values)))

# In[8]:
dict_pred = dict()
index = 0
episodes = TEST.copy().drop('Consciencia', axis=1)
count = 505
for episode in episodes.iterrows():
    episode = episode[1]
    _episode = {k: v for k, v in episode.iteritems()}
    bp = BeliefPropagation(model_teste)
    ret = bp.query(['Consciencia'], evidence=_episode,
                  joint=False, show_progress=True)
    print("predita: {} -----".format(count))
    print(ret['Consciencia'])
    print(ret['Consciencia'].values)
    dict_pred[index] = np.argmax(ret['Consciencia'].values)
    count += 1
    print(dict_pred[index])
    index += 1

# In[9]:
y_pred = np.array(list(dict_pred.values()))
ypred = label_binarize(y_pred, classes=[0,1,2,3])

# In[10]:
templates = TEST[['Consciencia']].to_numpy().T
ytrue = label_binarize(templates[0],
classes=["IMEDIATA", "NORMAL", "PERDIDA", "RISCO"])
y_true = np.array([np.argmax(x) for x in ytrue])

# In[11]:
n_classes = 4
fpr = dict()

```

```

tpr = dict()
roc_auc = dict()

precision = precision_score(y_true, y_pred, average=None)
recall = recall_score(y_true, y_pred, average=None)

for i in range(n_classes):
    fpr[i], tpr[i], t = roc_curve(ytrue[:, i], ypred[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
    print("roc_auc[{}] = {}, fpr{}, tpr{}".format(i, roc_auc[i], fpr[i],
tpr[i]))

print("precision = {}".format(precision))
print("recall = {}".format(recall))
# In[7]:
Labels = ['IMEDIATA', 'NORMAL', 'PERDIDA', 'RISCO']
colors = cycle(['aqua', 'blue', 'gold', 'darkorange'])
plt.figure()
for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color, label='Classe: {0} (area =
{1:0.2f})'.format(Labels[i], roc_auc[i]))
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
# %%

```

## 6.4 ANEXO D - ARTIGO

## **Estudo Comparativo Entre Modelos em Redes Neurais e Modelo Bayesiano para Consciência Situacional**

**Márcio Manoel Sagaz**<sup>1</sup>

<sup>1</sup>Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC)

Campus Universitário – Florianópolis – SC - Brasil

marciosagaz@yahoo.com.br

***Abstract.** Artificial neural networks are based on the biological neuron functioning model. The interconnection of these artificial neurons provides a network with parallel and non-linear processing for: pattern detection and recognition, data classification, among others. The use of smartphones by pedestrians causes a high number of accidents. In an attempt to reduce this problem, other works have proposed, through Bayesian networks, a situational awareness model with users using smartphones on urban streets. This work presents a proposed model in artificial neural networks to evaluate whether this method compared to Bayesian networks can improve situational awareness.*

***Resumo.** Rede neural artificial é baseada no modelo de funcionamento do neurônio biológico. A interligação desses neurônios artificiais propicia uma rede com processamento paralelo e não linear para: detecção e reconhecimento de padrões, classificação de dados, entre outros. O uso de smartphones por pedestres causa um elevado número de acidentes. Na tentativa de reduzir essa problemática, outros trabalhos propuseram, através de redes bayesianas, um modelo de consciência situacional com usuários usando smartphones em vias urbanas. Este trabalho apresenta um modelo proposto em redes neurais artificiais para avaliar se este método comparado ao de redes bayesianas poderá melhorar a consciência situacional.*



## 1. Introdução

Segundo (HAYKIN, 2001), inteligência artificial envolve o desenvolvimento de paradigmas ou algoritmos para que as máquinas realizem tarefas cognitivas semelhantes às como os seres humanos são capazes de realizar. Isto está relacionado ao processo de aprendizagem de máquina que para (LUGER, 2013) pode apresentar-se em quatro abordagens, sendo estas a abordagem simbólica, a abordagem conexionista, a abordagem genética (evolucionária) e a dinâmica (estocástica). Neste trabalho será desenvolvida uma abordagem conexionista usando-se redes neurais. A base de uma rede neural é o neurônio que é caracterizado por sinais de entrada, um conjunto de pesos de valor real, um nível de ativação e uma função limiar (LUGER, 2013).

Uma rede neural artificial em sua característica é definida segundo a topologia, o algoritmo de aprendizagem e o esquema de codificação (LUGER, 2013). Entre as topologias mais comuns estão a feedforward, a realimentada e as reticuladas com camadas simples e múltiplas (SILVA; SPATTI; FLAUZINO, 2010), este trabalho inicialmente não adotará uma topologia específica, pois sua escolha fará parte do estudo deste trabalho. Sobre os algoritmos de aprendizagem encontramos os métodos de treinamento supervisionado, não supervisionado e por reforço, assim como aprendizagem usando um lote de padrões e padrão a padrão (SILVA; SPATTI; FLAUZINO, 2010). Neste trabalho o método de aprendizagem também será estudado, avaliado e definido para aplicar a comparação com o algoritmo de redes bayesianas do trabalho de conclusão de curso (TCC) intitulado “Aplicação de Aprendizagem por Reforço para um Modelo Bayesiano de Consciência”. Em termos de codificação serão usados os mesmos dados utilizados na pesquisa “Road Awareness” para uma adequada aplicação no estudo de consciência situacional.

O uso da abordagem de rede neural artificial neste trabalho de conclusão de curso tem como objetivo auxiliar no desenvolvimento e experimentação de um modelo de estudo de consciência situacional. A construção deste modelo considera ainda uma avaliação em comparação à abordagem bayesiana, já desenvolvida em trabalhos anteriores.

## 2. Motivação

O uso de dispositivos móveis por pedestres em vias públicas tem ocasionado um elevado número de acidentes. O motivo é que o dispositivo causa ao pedestre uma falta de

consciência sobre o tráfego urbano ao seu redor. Em uma tentativa de reduzir a problemática exposta relacionada aos dispositivos móveis, foi proposto no projeto “Road Awareness” um modelo de consciência situacional com usuários usando smartphones em vias urbanas. Somando-se a este esforço o trabalho de TCC “Aplicação de Aprendizagem por Reforço para um Modelo Bayesiano de Consciência” programou e aplicou um algoritmo de aprendizagem por reforço na aprendizagem dos parâmetros do modelo em questão, a rede Bayesiano. Com o objetivo de contribuir com o projeto de consciência situacional, este trabalho apresenta um modelo proposto em redes neurais artificiais para avaliar se este método comparado ao de redes bayesiana poderá melhorar a consciência situacional.

### **3. Trabalhos Relacionados**

Nesta etapa será abordado trabalhos correlatos a área de interesse deste trabalho. No primeiro trabalho é apontada uma pesquisa *survey* onde são descritos o uso de alguns modelos de redes neurais para a previsão da taxa *foreign exchange* (FOREX). O segundo trabalho é uma pesquisa também *survey* que aponta para 4 arquiteturas de deep learning com redes neurais. O terceiro trabalho descreve a aplicação e a comparação de técnicas que utilizam análise de dados associadas a conceitos como o aprendizado de máquina para exploração de conhecimento em uma pesquisa sobre consciência situacional. Nos outros trabalhos vemos algumas pesquisas na área de ANN.

#### **3.1. Previsão de taxa de câmbio**

Modelos de predição de taxa de câmbio têm sido pesquisados usando-se ANNs. Para a construção destes modelos (PARAMITA et al, 2018) traz algumas abordagens sobre esse tema. Sendo assim, redes como *Back Propagation Neural Network* - (BPNN), *Functional Link Artificial Neural Network* (FLANN), *Radial Basis Function Neural Network* (RBFNN), *Recurrent Neural Network* (RNN) foram usadas na previsão (PARAMITA et al, 2018). Em relação a BPNN, (PARAMITA et al, 2018) aponta um trabalho que desenvolveu um sistema de negociação automatizado para a previsão da taxa de câmbio com uma única rede de *feedforward* de camada oculta. O trabalho de (PARAMITA et al, 2018) aponta o uso de FLANN em substituição às *multilayer ANN* (MLANN) devido ao custo computacional desta última. O RBFNN é eficaz na previsão de séries temporais financeiras. O RBFNN tem a capacidade de detectar as relações não lineares complexas entre as variáveis dependentes e

independentes (PARAMITA et al, 2018). O RNN fornece uma previsão melhor do que o *feedforward neural network* (FFNN). Modelos usando RNN para previsão prevêem taxas de câmbio melhores do que outros métodos e os pesquisadores concluem que RNN pode ser usado com sucesso para previsão dessas taxas (PARAMITA et al, 2018).

### 3.2 A Arquitetura de 4 Redes Neurais Profundas

O algoritmo de aprendizado profundo consiste em uma arquitetura hierárquica com muitas camadas, cada uma das quais constitui uma unidade de processamento de informações não lineares (LIU et al, 2016), podendo assim representar funções com maior complexidade se o número de camadas e unidades em uma única camada for aumentado. Neste trabalho, (LIU et al, 2016), apresenta quatro arquiteturas de aprendizado profundo, que são máquinas *restricted Boltzmann machines* (RBMs), *deep belief networks* (DBNs), *autoencoder* (AE), and *convolutional neural networks* (CNNs). Os RBMs têm sido usados para gerar modelos estocásticos de ANNs que podem aprender a distribuição de probabilidade com relação a suas entradas. Os RBMs são importantes na redução de dimensionalidade, filtragem colaborativa, classificação e aprendizado de recursos. O modelo de um DBN é formado por camadas onde a cada duas camadas adjacentes formam um RBM (LIU et al, 2016). A camada visível de cada RBM é conectada à camada oculta do RBM anterior e as duas camadas superiores não são direcionais. Um AE é um algoritmo de aprendizado não supervisionado usado para codificar eficientemente o conjunto de dados com o objetivo de reduzir a dimensionalidade (LIU et al, 2016). O AE é uma rede neural de *feedforward* de camada oculta semelhante à *MLP*. As CNNs é uma rede neural multicamada que consiste em dois tipos diferentes de camadas, isto é, camadas de convolução e camadas de subamostragem, são um subtipo de arquitetura discriminativa profunda com desempenho satisfatório no processamento de dados bidimensionais com topologia em grade, como imagens e vídeos (LIU et al, 2016). O aprendizado profundo lida com a análise de grandes dados usados para extrair recursos de alto nível e obter representações hierárquicas. As áreas beneficiadas são o processamento de imagens, a classificação de manuscrita, as interfaces cerebrais para computador, a classificação de sentimentos e PLN (LIU et al, 2016)

### 3.3 Análise para Consciência Situacional

Os dados da pesquisa provém de um experimento no qual o objetivo foi analisar o comportamento do ser humano quando inserido em um contexto que simula o tráfego urbano, ele em teste interagia com o aparelho de smartphone para evitar ser atropelado. O experimento foi realizado por vinte participantes (LANGER, 2018). Para seu objetivo (LANGER, 2018) tinha uma quantidade relativamente pequena de dados (menos de 1000 amostras). Porém os métodos escolhidos para a classificação foram: Regressão Logística, *K Nearest Neighbors*, Máquinas de Vetores de Suporte, Árvores de decisão, *Adaptive Boosting*, *Bagging*, *Gradient Boosting*, Floresta aleatória e Perceptron Multicamadas. Avaliou-se três aplicações, uma com a primeira abordagem (categorias) e as outras duas com a segunda abordagem nível quantitativo com o valor de *threshold* 0.5 e 0.7. Para cada método utilizado, foi empregado o método de validação cruzada (LANGER, 2018). Dados os resultados a abordagem com *threshold* 0.7 foi escolhida como a principal linha a ser seguida. Regressão Logística e Perceptron Multicamada melhor acurácia, e confiabilidade. A Árvore de decisão e *Adaptive Boosting* desempenho ruim para falsos negativos, mas a diferença para as restantes não foi significativa. Técnicas de *undersampling* e *oversampling* foram utilizadas em conjunto com a análise de componentes principais e redução de dimensionalidade (LANGER, 2018).

### 3.4 Outros Trabalhos

No trabalho de (CHEN, CHANG, 2019) é desenvolvido um método estocástico eficiente, livre de gradiente, o *ensemble neural networks* (ENN). No ENN, o processo de otimização depende de matrizes de covariância em vez de derivadas. Este método fornece quantificação de incertezas, uma vez que é construído sob a estrutura bayesiana. Assim, (CHEN, CHANG, 2019) demonstra em seu trabalho que o ENN tem um desempenho muito melhor do que as redes neurais *bayesianas* tradicionais (BNN).

Um trabalho que busca a detecção de pedestres ocluídos em visão computacional é o de (ZOU et al, 2019). Em seu trabalho ele propõe um modelo de rede neural guiada por atenção ou *attention guided neural network* (AGNN) para busca de pedestres sobre uma oclusão. Neste modelo ele usa duas abordagens de redes neurais e uma delas é a rede neural convolucional empregada para extrair os recursos de alto nível. Na segunda, as sequências de recursos são classificadas por rede neural recorrente na ordem adequada com base nos resultados ponderados (ZOU et al, 2019).

Na busca de um método confiável para diagnóstico para predição do diabetes (EL\_JERJAWI, ABU-NASER, 2018) usa rede neural artificial visando prever se uma pessoa é diabética. Usando um modelo de ANN busca minimizar o erro médio e alcançando acurácia de 87,3%.

### 3.5 Comparação Entre Os Trabalhos

Trabalhos	Utiliza redes neurais	Aborda consciência situacional	Faz avaliação com métodos bayesianos	Aplicada em predição de informação	Aplicada em ambiente com tráfego real ou simulado
(PARAMITA et al, 2018)	x			x	
(LIU et al, 2016)	x				
(LANGER, 2018)	x	x		x	x
(CHEN, Y.; CHANG, H., 2019)	x		x (BNN)		
(ZOU et al, 2019)	x			x	
(EL_JERJAWI, ABU-NASER, 2018)	x			x	

Tabela 1 – Levantamento de trabalhos relacionados

Este trabalho se diferencia dos outros, pois aborda dados que são provenientes de um *dataset* que modela um ambiente urbano com tráfego de carros onde o pedestre pode ou não estar consciente dos carros que se aproximam. Neste caso, é feita uma avaliação buscando observar a consciência situacional do pedestre. O trabalho visa usar modelos de ANN e comparar seus resultados de predição de consciência situacional com um algoritmo produzido no trabalho de (PEREIRA, 2017) que envolve uma Rede Bayesiana (RB).

## 4. Desenvolvimento

### 4.1 Treinamentos da Rede de Função de Base Radial

Neste trabalho foi usado um treinamento semi-supervisionado usando duas etapas. Sendo a camada oculta um conjunto de centros onde cada neurônio é um centro, o treinamento é de forma não supervisionada usando algum método de clusterização por centróides, neste trabalho foi usado k-means como método de clusterização e a gaussiana como um função radial de aproximação. A segunda parte da rede foi treinada de forma supervisionada para obtenção dos pesos sinápticos. Sendo a saída da rede dada por  $\mathbf{G}_{p \times k} \mathbf{W}_{k \times m} = \mathbf{Y}_{p \times m}$  obtém-se  $\mathbf{G}_{p \times k}$  para uma a gaussiana através de  $\varphi_j^{(i)} = e^{-v}$ ,  $v = \frac{num}{den}$ ,  $num = -\|x^{(i)} - \mu_j\|_2^2$ ,  $den = \sigma^2$ ,  $\sigma = \frac{num2}{den2}$ ,  $num2 = \max(\{\mu_1, \dots, \mu_k\}) - \min(\{\mu_1, \dots, \mu_k\})$ ,  $den2 = \sqrt{2k}$ , onde  $\mathbf{G}_{p \times k} = \varphi_j^{(i)}$ . Obtém-se  $\mathbf{W}_{k \times m}$  de  $\mathbf{W}_{k \times m} = \mathbf{G}_{p \times k}^+ \mathbf{Y}_{p \times m}$  onde  $\mathbf{G}^+$  é a pseudo inversa da matriz  $\mathbf{G}$ , pois esta geralmente não é uma matriz quadrada. O algoritmo não é iterativo, sendo assim, encontrar os pesos é algo simples como uma multiplicação de matrizes.

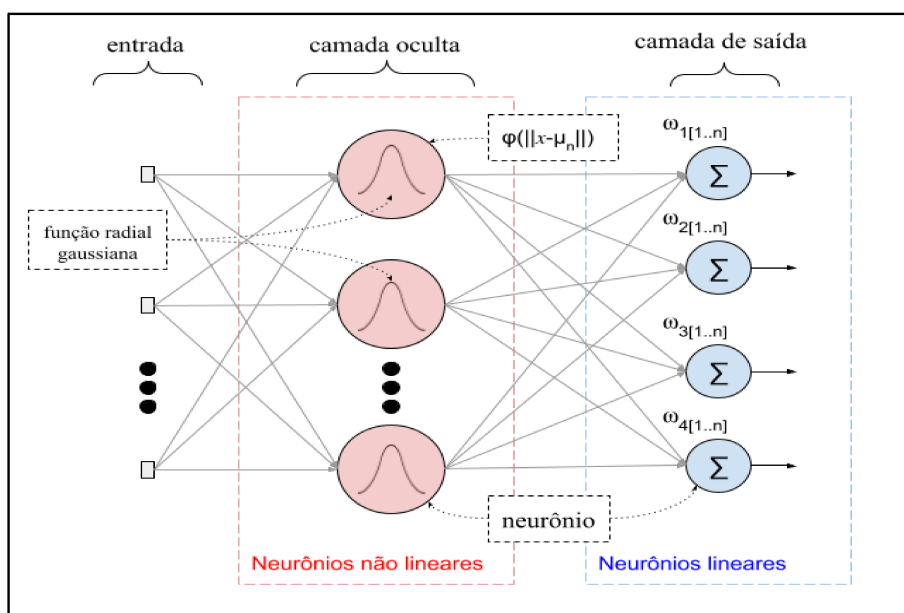


Figura 18 – Estrutura da rede RBF treinada no trabalho.

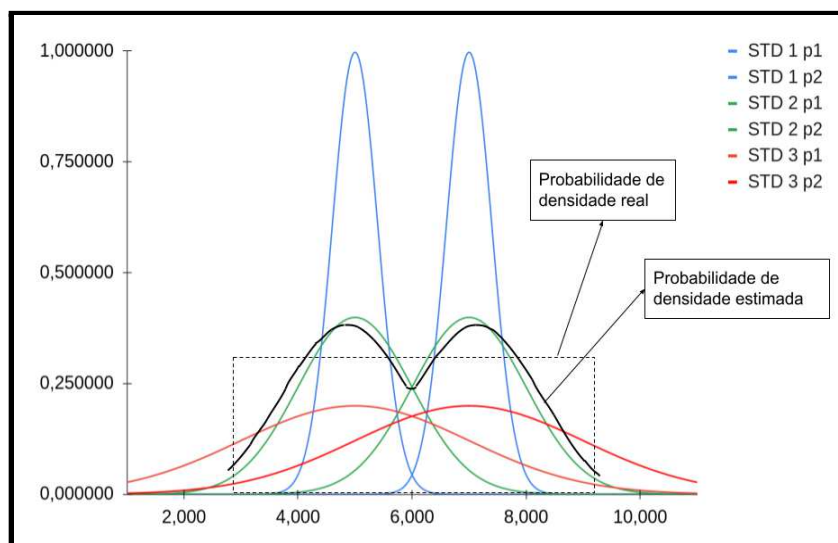
A figura 18 ilustra a estrutura da rede RBF treinada neste trabalho. Pode-se observar que são 4 neurônios de saídas e que são equivalentes para a classificação das consciências IMEDIATA, NORMAL, RISCO, PERDIDA. Os dados estavam inicialmente ordenados

segundo o experimento e seguiam sua sequência de ocorrência dada pelo experimento de simulação com cada pedestre, mas os dados foram reordenados, esses dados foram sorteados de forma aleatória para evitar que a rede fosse treinada e influenciada pela sequência ordenada inicial. Iho são entradas, amarelo apenas são referências para análise e não são usados na rede.

Os dados foram modificados inicialmente para 3 treinamentos e em todas as modificações a sequência de ordenação aleatória foi seguida. Da base, os primeiros 70% dos dados eram para treinamento e os 30% restantes eram para teste. No primeiro treinamento os dados foram categorizados de 1 a 12 para cada classe, no segundo categorizados de forma binária em cada classe. Por fim, foi realizado o teste desbalanceado de seis modos diferentes, a fim de testar e garantir uma independência entre a categorização e o desbalanceamento linear. A intenção é verificar se estes desbalanceamentos influenciam na classificação ou apenas ajudam na generalização. Os experimentos apontaram igualdade de resultados em todos os 6 testes e a influência da adoção do desbalanceamento pode ser vista no item [4.5.1](#)

## **4.2. Treinamentos da Rede Neural Probabilística**

Para o treinamento da rede RNP foi usada a implementação do algoritmo da biblioteca Neupy. Assim como descrito para a rede RBF o algoritmo de treinamento da RNP não é iterativo, mas neste caso há a necessidade de encontrar um ponto ótimo para o fator de alisamento, que é o desvio padrão, onde a acurácia do algoritmo é máxima. Esse ajuste está relacionado com as características intrínsecas das distribuições dos dados, ou seja, esse ajuste pode ser compreendido no exemplo da imagem da figura 19, onde o fator de alisamento ajusta a melhor estimativa da probabilidade de densidade. A estratégia é encontrar um valor ótimo para este conjunto de dados.



**Figura 19** – Fator de alisamento para diferentes valores de desvios padrões

Os dados de entrada seguem um critério semelhante ao explanado no item 4.1 deste trabalho, onde os dados foram sorteados de forma aleatória para evitar que a rede fosse treinada e influenciada pela sequência ordenada inicial do experimento.

### 4.3. Resultados da rede RBF

Os resultados dos testes realizados para os três tipos de experimentos são apresentados na figuras 20, 21 e 22, pode-se ver que o formato dos dados desbalanceados apresentam uma performance melhor na aproximação da classificação dos dados, enquanto que o pior resultado é a estratégia onde todos os valores foram categorizados binariamente. As figuras apresentam o comportamento da rede quando usamos os dados de treinamento e os dados de testes. Observa-se uma consistência na generalização da rede. Na prática para o experimento foi testado uma faixa de 1 até 199 núcleos, mas essa quantidade de núcleos é uma extrapolação demasiada para a quantidade de amostras utilizadas no treino. Logo, uma faixa praticável seria 16 a 30 núcleos, o que apresenta uma acurácia de aproximadamente 79% para cerca de 20 núcleos no melhor experimento 22. No teste desbalanceado de seis modos diferentes todos os resultados são consistentes com o gráfico de 22 para os seis casos.



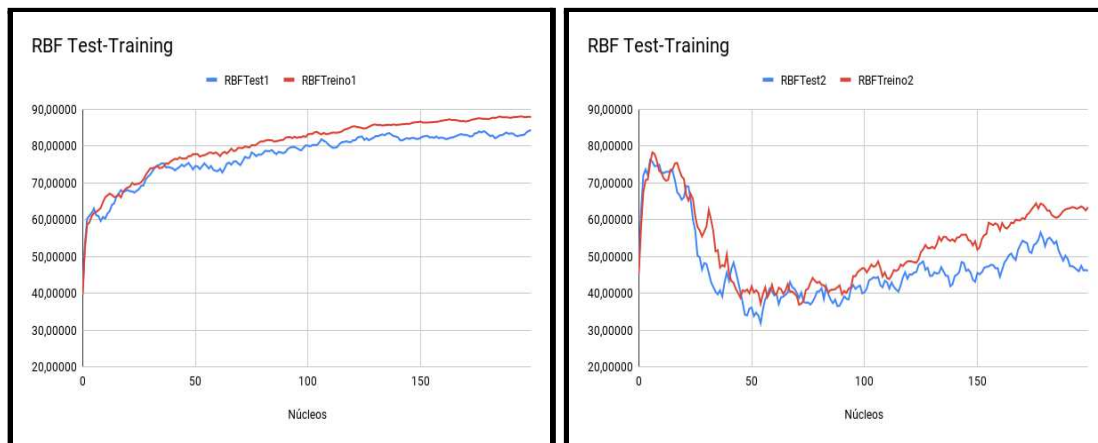


Figura 20 – Treino 1 e 2 : desempenho dos dados teste X treino.



Figura 22 – Treino 3: desempenho dos dados teste X treino.

Ao analisar-se os conjuntos dos dados, obtém-se uma matriz de confusão em que a rede RBF tem um *F1 Score* muito baixo na identificação da classe NORMAL, pois a rede identifica a classe NORMAL como uma classe IMEDIATA na maioria dos casos. O *F1 Score* do RISCO é o mais alto para esta rede na classificação e o *RECALL* é 100% para a classificação PERDIDA

MATRIZ CONFUSÃO						
RBF		PREDITA				FALSO NEGATIVO
		IMEDIATA	NORMAL	RISCO	PERDIDA	
REAL	IMEDIATA	56	11	0	0	11
	NORMAL	27	11	1	0	28
	RISCO	0	0	74	6	6
	PERDIDA	0	0	0	31	0
FALSO POSITIVO		27	11	1	6	45
AMOSTRAS	217					
ACURÁCIA	79,26%					
PRECISÃO	79,26%	67,47%	50,00%	98,67%	83,78%	
RECALL	79,26%	83,58%	28,21%	92,50%	100,00%	
F1 Score	79,26%	74,67%	36,07%	95,48%	91,18%	

Tabela 22 – Matriz Confusão rede RBF.

#### 4.4. Resultados da RNP

A estratégia para encontrar o fator de alisamento foi treinar a rede 400 vezes para prever o conjunto de teste e o conjunto de treino, assim progressivamente foi-se variando os valores do fator para mapear a acurácia da saída da rede.

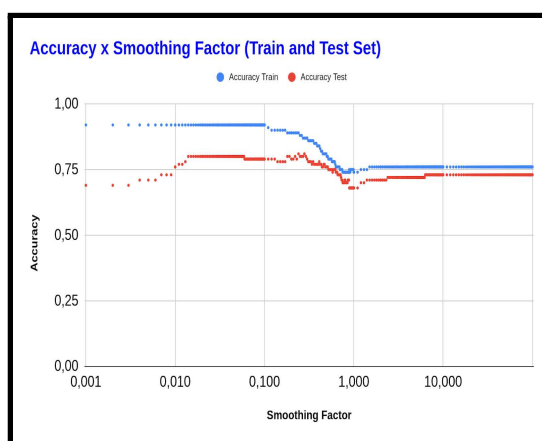


Figura 23 – Fator de alisamento para diferentes valores de desvios padrões

A figura 23 representa a relação entre acurácia e o fator de alisamento para os dois conjuntos utilizados na rede, onde o eixo do fator está representado em forma logarítmica. Os valores ótimos para este conjunto de treino estão na faixa aproximada entre 0,2 a 0,6. Com esses valores o algoritmo alcançou uma acurácia de aproximadamente 80%. Para a matriz de confusão da rede RNP, observa-se que ela tem uma dificuldade em distinguir entre a classe IMEDIATA e NORMAL. A rede RBF e a RNP têm o mesmo desempenho quando olhamos o

F1 Score do RISCO e a RNP uma vantagem 2% a mais que a rede RBF quando olhamos o F1 Score da classe PERDIDA.

MATRIZ CONFUSÃO						
RNP		PREDITA				FALSO NEGATIVO
		IMEDIATA	NORMAL	RISCO	PERDIDA	
REAL	IMEDIATA	45	22	0	0	22
	NORMAL	14	24	1	0	15
	RISCO	0	2	74	4	6
	PERDIDA	0	0	0	31	0
FALSO POSITIVO		14	24	1	4	43
AMOSTRAS	217					
ACURÁCIA	80,18%					
PRECISÃO	80,18%	76,27%	50,00%	98,67%	88,57%	
RECALL	80,18%	67,16%	61,54%	92,50%	100,00%	
F1 Score	80,18%	71,43%	55,17%	95,48%	93,94%	

Tabela 23 – Matriz Confusão RNP.

#### 4.5. Treinamento da Rede Bayesiana

Baseado no trabalho de (PEREIRA, 2017), mas sem a preocupação de usar um agente para melhorar os resultados da PC a priori do modelo deste trabalho, aplicou-se os algoritmos e bibliotecas utilizadas por (PEREIRA, 2017) para gerar uma RB relacionada com as definições deste trabalho, assim objetivou-se comparar o desempenho da PC da RB para a consciência situacional com relação aos modelos de redes neurais deste trabalho já citadas em seções anteriores. Foi utilizada a linguagem de *script python* com a biblioteca de *pandas* para ler os dados e a biblioteca *pgmpy* para criação do modelo, inferência e estimação bayesiana. A estrutura dos dados seguem a mesma definição aplicada às redes neurais, ou seja, tem a mesma lógica de embaralhamento de dados. Os dados foram divididos em 70% para geração do modelo e levantamento do valores de probabilidade condicional a priori e os outros 30% restantes foram usados para testar o modelo. O modelo criado é representado na figura 24 através do *software Netica* expressando os devidos valores de probabilidade condicional a priori. A partir deste modelo é possível inferir quais seriam os níveis de consciência situacional para o pedestre em determinada situação apresentada.

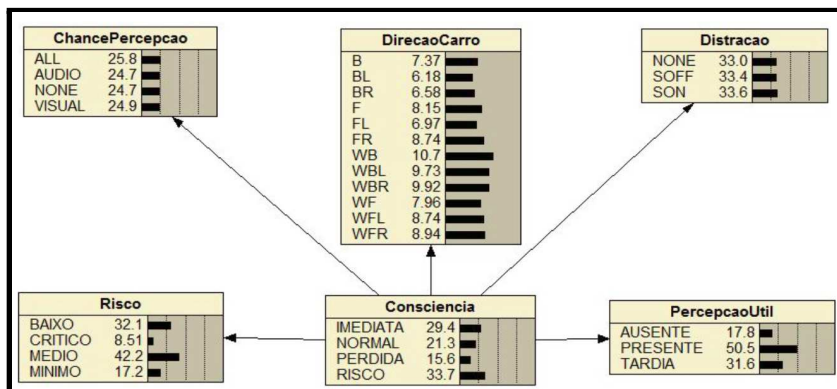


Figura 24 – Fator de alisamento para diferentes valores de desvios padrões

Para o caso da figura 25, onde percepção útil é ausente; o nível de distração é médio; a direção do carro é *warning back*, significando que o carro vem por trás e o pedestre está na mesma faixa que o carro; a chance de percepção é visual; o risco em nível médio, explicitando que há alguma experiência do pedestre com o ambiente, temos que o modelo RB aponta 99,7% para um caso de consciência perdida, o que na prática seria um possível atropelamento no trânsito.

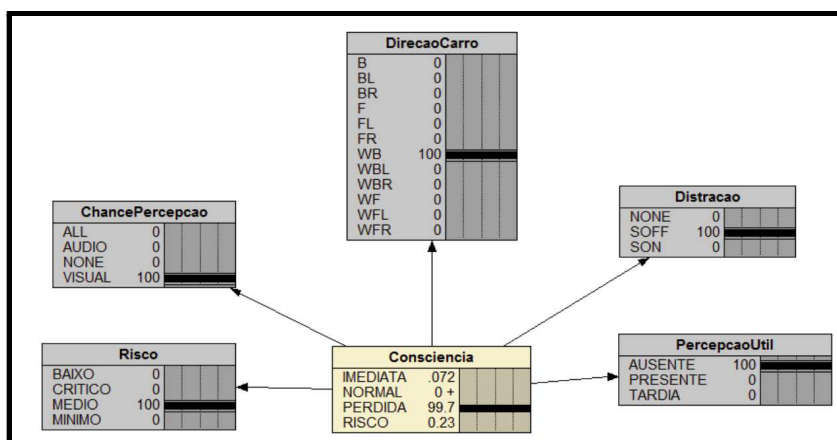


Figura 25 – Fator de alisamento para diferentes valores de desvios padrões

#### 4.6 Resultado da Rede Bayesiana

Utilizando-se os 30% da base de teste destinada para verificação do modelo, gerou-se para cada tupla de entrada <Distracao, DirecaoCarro, ChancePercepcao, PercepcaoUtil, Risco> uma verificação da probabilidade condicional para o caso <NONE, F, VISUAL, PRESENTE, BAIXO>, entrega uma [<consciência; probabilidade>] de [<IMEDIATA, 0,8061>; <NORMAL, 0,1929>; <PERDIDA, 0,0009>; <RISCO, 0,0000>]. As saídas de probabilidade entregues pelo modelo foram comparadas com o valor real da base de dados (base de teste) em cada tupla para obter a assertividade do modelo, portanto usou-se a matriz de confusão

como avaliação da RB, assim como para a rede RBF e a RNP, ver tabela 26. O *F1 Score* da RB ainda é baixo para a classe NORMAL, mas tem um desempenho melhor que os das redes neurais. Para a classe RISCO o *F1 Score* é tão alto quanto o das redes neurais, porém chega a ser 8% menor quando comparada a RNP para a classe PERDIDA.

RB		PREDITA				FALSO NEGATIVO
		IMEDIATA	NORMAL	RISCO	PERDIDA	
REAL	IMEDIATA	48	19	0	0	19
	NORMAL	15	24	0	0	15
	RISCO	0	0	75	5	5
	PERDIDA	0	0	4	27	4
FALSO POSITIVO		15	19	4	5	43
AMOSTRAS	217					
ACURÁCIA	80,18%					
PRECISÃO	80,18%	76,19%	55,81%	94,94%	84,38%	
RECALL	80,18%	71,64%	61,54%	93,75%	87,10%	
F1 Score	80,18%	73,85%	58,54%	94,34%	85,71%	

Tabela 26 – Matriz Confusão RB.

#### 4.7 Análise Comparativa dos Modelos

Foi produzida uma análise baseada no desempenho da curva ROC (*Receiver Operating Characteristic*) e a sua área de desempenho AUC (*area under the ROC curve*). Quando olha-se para a análise ROC/AUC objetiva-se identificar uma diferença no desempenho desses algoritmos para cada classe, pois a acurácia para a rede RBF igual 79,26%, para RNP 80,18% e para a RN 80,18% que observa-se através da matriz de confusão em cada experimento não traz uma boa diferenciação. Temos na verdade um empate no desempenho quando olha-se apenas a acurácia. A figura 26, 27 e 28 mostram a curva ROC/AUC para cada classe usando o algoritmo da rede RBF, RNP, RN respectivamente, sendo assim os valores de AUC observados são: Na rede RBF são de 0,81 para IMEDIATA, 0,66 para NORMAL, 0,96 para RISCO e 0,98 para PERDIDA; Na RNP são de 0,79 para IMEDIATA, 0,74 para NORMAL, 0,96 para RISCO e 0,99 para PERDIDA; Na RB são de 0,81 para IMEDIATA, 0,75 para NORMAL, 0,95 para RISCO e 0,92 para PERDIDA.

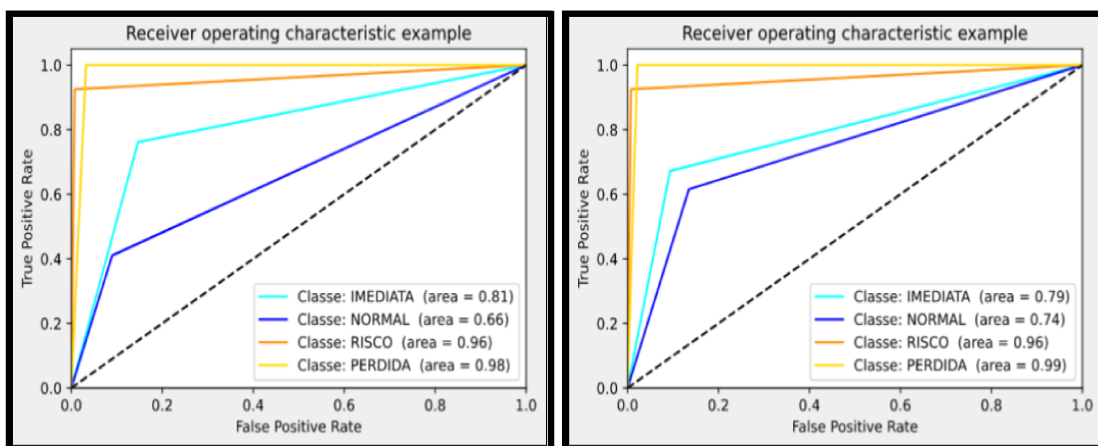


Figura 26 – ROC/AUC para o algoritmo da rede RBF e RPN

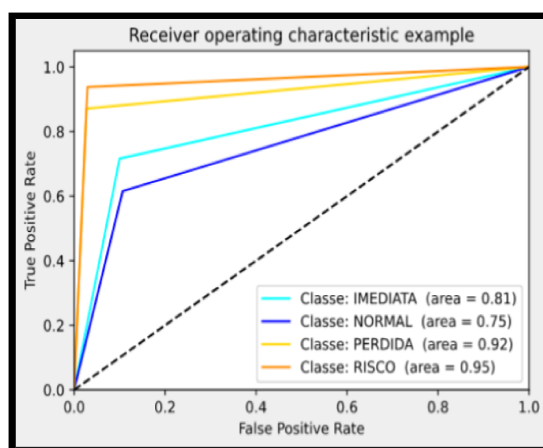


Figura 28 – ROC/AUC para o algoritmo RB

Normalmente a curva ROC utiliza uma variação de *threshold*, mas no caso dessa análise o *threshold* é fixo, pois buscou-se o ponto ótimo para cada algoritmo como descrito nas análises de cada tópico acima. Por isso, a intenção da AUC nesta análise é encontrar uma evidência do desempenho nas comparações dos algoritmos.

## 5 Conclusão e Trabalhos Futuros

Diante do requisito do modelo para consciência situacional e o objetivo de comparação das redes neurais com o modelo de rede bayesiana, procurou-se algoritmos que pudessem superar o desempenho de uma rede bayesiana. O item 4.1 a 4.4 descreve os algoritmos escolhidos e que cumprem com o objetivo. Todo item 4 mostra os resultados analíticos sobre as redes neurais para alcançar o objetivo do trabalho. Utilizou-se a matriz de confusão, acurácia, precisão, *recall* e *f1 score* para verificar o desempenho dos algoritmos.

Sobre os resultados alcançados pode-se dizer que ao considerarmos o nível de importância das classes para segurança do pedestre em relação a sua consciência situacional, temos a seguinte ordem: PERDIDA, RISCO, NORMAL e IMEDIATA. Assim, quando olha-se os resultados alcançados podemos organizar o desempenho dos algoritmos em: RNP, rede RBF e RB. Todos os algoritmos avaliados tiveram um baixo desempenho em separar os eventos da classe NORMAL dos eventos da classe IMEDIATA. Uma suposição para tal dificuldade é que neste trabalho a separação destas classes de consciência partiu de uma definição baseada na média do tempo de consciência, que pode não encontrar embasamento real nas características apontadas para o treinamento.

### 5.1. Trabalhos futuros

A partir do trabalho de pesquisa e desenvolvimento aqui realizado pôde-se observar as oportunidades de trabalhos a serem desenvolvidos futuramente, como por exemplo: Estudo, implementação e comparação entre essas redes usando outra base de dados; Estudo e implementação de novas redes neurais; Manipular algumas características dos dados para alcançar uma melhor definição na divisão das classes; Uso do estudo para alavancar novas pesquisas em relação ao monitoramento da consciência do pedestre; Uso dos algoritmos em aplicações de smartphones monitorar a consciência; Uso dos algoritmos para consciência situacional de motoristas de veículos.

## 6. Referência

CHEN, Y.; CHANG, H.; MENG, J.; ZHANG, D.; *Ensemble Neural Networks (ENN): A gradient-free stochastic method*, Neural Networks, Vol. 110, 2019, P. 170-185, ISSN 0893-6080, <http://www.sciencedirect.com/science/article/pii/S0893608018303319>, 2018

HAYKIN, SIMON. *Redes Neurais Princípios e Práticas*. 2. ed. Bookman, 2001.

KOUROU K., EXARCHOS T.P., EXARCHOS K.P., KARAMOUZIS M.V., FOTIADIS D.I. *Machine learning applications in cancer prognosis and prediction* Comput. Struct. Biotechnol. J., 13 (2015), pp. 8-17, 10.1016/j.csbj.2014.11.005

LANGER, G. A. *Análise de dados aplicada para consciência situacional de pedestres* [S.l.]: Florianópolis, SC, 2018.

LIU, W.; WANG, Z.; LIU, X.; ZENG, N.; LIU, Y.; ALSAADI, F. *A survey of deep neural network architectures and their applications*. Neurocomputing. 234. 10.1016/j.neucom.2016.12.038, 2016.

LUGER, G. F. *Inteligência Artificial*. 6. ed. [S.l.]: Pearson Education, 2013.

PARAMITA B., Pragyani; REKHA D., Smruti; MISHRA, Debahuti. *A survey on exchange rate prediction using neural network based methods*. International Journal of Engineering & Technology, [S.l.], v. 7, n. 2.6, p. 64-70, mar. 2018. ISSN 2227-524X. 2020.

PEREIRA, R. L. *Aplicação de Aprendizagem por Reforço para um Modelo Bayesiano de Consciência Situacional*. Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, 2017.

SANTOS, E.; FAZENDA, B. *Computational Model of Situational Awareness for users of smartphones in the vicinity of traffic*. [S.l.], 2016.

SILVA, IVAN N.; SPATTI, DANILO H.; FLAUZINO, ROGÉRIO A.; *Redes Neurais Artificiais para engenharia e ciências aplicadas*, Artliber, 2010.

ZOU, T.; YANG, S.; ZHANG, Y.; YE, M.; *Attention guided neural network models for occluded pedestrian detection*. ISSN 0167-8655, <http://www.sciencedirect.com/science/article/pii/S0167865519303733>, 2019