Projeto Sensorless Drive Diagnosis

Leonardo Martelli Oliveira, Marcello Fabrizio

I. Introdução

Automóveis são uma parte crucial do mundo moderno, sendo utilizados para transporte pessoal, público, de mercadorias e em demais necessidades da sociedade. Com tamanha importância, um nível de segurança é exigido, garantindo o bom funcionamento de veículos. Com a tecnologia disponível hoje, é possível arquitetar soluções que ajudam na confiabilidade dos carros, como por exemplo a detecção prévia de problemas no sistema de transmissão elétrico dos mesmos. A detecção é tarefa que pode ser feita por monitoramento e coleta de sinais elétricos em componentes do sistema foco. Após pode se realizar análises e classificações de tais amostras para previsão de possíveis erros.

Este trabalho realiza a análise de classificadores paramétricos e não-paramétricos aplicados sobre um conjunto de dados composto por amostras de sinais elétricos de sistemas de transmissão veiculares, e comparando com resultados obtidos por trabalhos anteriores, consultados na literatura.

II. Base de Dados

O objetivo do projeto abordado no artigo de origem é a utilização de um método para redução de complexidade para implementação de sistemas de controle e monitoramento do sistema de transmissão elétrica de veículos, que consiste em uma instalação composta de várias componentes e uma parte crucial da máquina. Os autores não utilizam sensores para o monitoramento de tal sistema, eles abordam o uso das fases da corrente alternada do motor para isso[1]. Os autores trazem o estudo novamente em um periódico diferente, trazendo mais informações sobre o contexto [2].

A base de dados **Sensorless Drive Diagnosis** foi retirada do repositório online de aprendizado de máquina da **University of California, Irvine**[3]. A base de dados origina do artigo **Sensorless Drive Diagnosis Using Automated Feature Extraction, Significance Ranking and Reduction**[1].

Os dados foram obtidos através de aparelhagem desenvolvida em um fundo de pesquisa alemão [2].

No processo de geração dos dados, foi utilizado o *Empirical Mode Descomposition* (EMD), parte da transformação de Hilbert-Huang[2], para decompor os sinais coletados em *Intrinsic Mode Functions* (IMF) - componentes de frequência intrínsecos, transformando-os em dados trabalháveis.

O conjunto de dados utilizado no projeto possui 58508 amostras e é composto por 48 variáveis sendo dos sinais do sistema de transmissão, classificadas em um conjunto de 11 classes distintas, sendo estes os possíveis estados do sistema - 1 para sem defeitos e os 10 remanescentes, 10 defeitos distintos. O conjunto original não contém dados de cabeçalho. Todas variáveis são numéricas de ponto flutuante.

Para o desenvolvimento do projeto, as colunas referentes às variáveis independentes, foram nomeadas da seguinte maneira: **Feature*X***, sendo *X* um número de 1 a 48; a coluna contendo as classes foi representada como **label**.

As classes no *dataset* apresenta um balanceamento perfeito com 5319 amostras para cada classe.

III. Avaliação de Desempenho

Rahman *et al.*[4] utilizam o método de validação cruzada *k folding* para avaliação de desempenho. A validação cruzada *k folding* divide os dados em **k** grupos, onde os grupos **k** - 1 são utilizados para treinamento do modelo e o grupo restante para teste, chamado de grupo de validação[5, p. 3]. O seguinte processo é aplicado para os grupos:

- Um modelo é treinado sobre os **k** - 1 grupos
- O modelo é validado sobre o grupo restante
- O processo é repetido até que cada grupo tenha sido usado para validação

O artigo utiliza três valores para **k**, sendo **k** o número de partições: 4, 10 e 100. A acurácia da validação cruzada é obtida através das **k** médias obtidas nos grupos de validação.

Scardapane *et al.*[6] propõem em seus estudos otimizar pesos dos neurônios, quantidade de neurônios em cada camada oculta e seleção de características na geração de redes neurais simultaneamente. Um dos conjuntos de dados selecionados para o experimento, foi o *Sensorless Drive Diagnosis*. Os resultados obtidos no estudo são comparados de acordo com a regularização para a rede neural utilizada, comparando os estilos clássicos - L1 e L2, com um método de esparsidade em nível de grupo.

Regularização no escopo de redes neurais, significa um modo de diminuir a complexidade das redes neurais durante o treinamento, minimizando o *overfitting*, ou seja, sobreajuste/superestimação dos resultados [7, p. 28].

São elencadas pelo artigo três regularizações L2, L1 e *Sparse Group* L1. L2 é conhecida também como *weight decay*, calculando um termo de regularização Ω obtido a partir da soma dos quadrados da matriz de pesos da rede neural, este termo é ponderado e adicionado a função de perda ou função de custo - utilizada para minimizar o erro -, obtendo-se uma nova função de perda; L1 pode ser referenciada também como regressão de Lasso, e consiste em calcular um termo de regularização Ω a partir dos pesos absolutos da rede neural, adicionando-o a função de custo [8]; A regularização L1 de espariedade de grupo (*Sparse Group* L1), onde a penalização por erro é aplicada diferente das demais - aborda em conjunto regularização Lasso em grupo e Lasso.

É utilizado *Holdout* para separação dos conjuntos de treino e teste, com $\frac{3}{4}$ para treino e $\frac{1}{4}$ para teste. A técnica *Holdout* tem funcionamento em uma separação percentual do conjunto inteiro de dados em treino e teste. Em especifico para esse artigo das amostras é feita de maneira aleatória, e há a aplicação dos dados N repetições - no caso do artigo $N = 25$.

Com um estudo focando especificamente no *dataset* do *Sensorless Drive Diagnosis*, Zhou *et al.*[9], apresentam uma

solução utilizando redes neurais Bayesianas - redes neurais que utilizam como base de aprendizado modelos Bayesiano [10].

No experimento é selecionado, sem explanação, apenas 40000 amostras (originalmente o dataset contém 58508, redução de cerca de 31%), utilizando-se do método de avaliação para os conjuntos de treino e teste, o *Holdout*, com 95% de dados para treino (38000) e 5% para teste (2000). O *Holdout* consiste em selecionar uma parcela do conjunto de dados, sem estratificação.

Grüner *et al.*[11] trazem um estudo analisando a competitividade entre métodos não-aprendizado profundo (*non Deep Learning*) com os de aprendizado profundo (*Deep Learning*), entre eles K vizinhos mais próximos (KNN), RF e Redes Neurais Artificiais com três camadas ocultas.

Jia *et al.* [12] apresentam um estudo comparando diferentes classificadores em múltiplos casos de teste, sendo um deles o conjunto de dados *Sensorless Drive Diagnosis*. Para análise de desempenho de SVM, os autores utilizaram 10-fold cross-validation. Nos demais algoritmos, não é informado método de avaliação utilizado.

IV. Referencial Bibliográfico

Rahman *et al.*[4] comparam os métodos de classificação do conjunto de dados para avaliar a precisão da classificação por RF e *Support Vector Classifier* (SVC) linear. Utilizando validação cruzada, os autores obtiveram resultados que mostram que a classificação por RF é mais precisa do que SVC linear para a classificação desta base, pois RF obteve acurácia de 99.82%, enquanto SVC linear obteve 66.04%. O método de avaliação utilizado foi o da validação cruzada, que separa o conjunto de **k** grupos, onde o grupo **k** - 1 é utilizado para e teste e os demais para treinamento.

Scardapane *et al.*[6] trazem a comparação de resultados em redes neurais utilizando três diferentes regularizações: *Weight decay* L2-NN, *Lasso penalty* (L1-NN) e *Sparse variation* (SG-L1-NN). Nos testes realizados, para L2 obteve 98% de acurácia; para L1 98%; e SG-L2 97%. Para avaliação dos dados, os autores utilizaram *Holdout*, 75% treino / 25% teste, rodando cada experimento 25 vezes para obter a média nas variações estatísticas. Os resultados obtidos são a média das acurácias das N repetições. Utilizando regularização L2 e L1 obteve-se 98% de acurácia, já utilizando *Sparse Group* L1 o resultado foi de 97% de acurácia.

Zhou *et al.*[9] propõem o estudo em um algoritmo de seleção de características, sobre o conjunto de dados. A comparação de resultados no artigo é feita treinando e testando uma rede neural Bayesiana, antes (*Baysean Network Before Feature Selection*, BN-BFS) e após (*Baysean Network After Feature Selection*, BN-AFS) seleção de características. Obtém-se 88.25% de acurácia antes e 88.30% após a seleção. Os autores separam os dados, 40000 amostras, em 95% para treinamento (38000 amostras) e 5% para teste (2000 amostras). Os resultados são comparados entre antes e depois da seleção de características. Antes é obtido 88.25%, após 88.30% de acurácia.

Grüner *et al.*[11] comparam vários métodos de aprendizagem com diferentes categorias: tradicionais, de *ensemble* e de aprendizado profundo. São estudados diferentes classificadores: KNN, RF, e Rede Neural Artificial com 3 camadas ocultas (ANN-3). Os dados de teste e treino são avaliados usando 5-fold cross-validation. A média de acurácias utilizando 5-fold Cross-validation para os três métodos foi de mais de 99%: KNN - 99.94%, RF - 99.92% Redes Neurais Artificiais - 99.64%.

Jia *et al.* [12] comparam os resultados dos métodos pelas suas acurácias nas classificações binárias realizadas. Para *Support Vector Machine* (SVM), o estudo utiliza 10-fold cross-validation para obter conjunto de treino e teste. Obtendo-se a média das acurácias de cada resultado das classificações binárias, *Maximum Mean Discrepancy* atingiu 98% e SVM 95.54%.

Uma melhor ilustração e comparação dos resultados em trabalhos relacionados pode ser vista na Tabela I.

TABELA I: Resultados obtidos em trabalhos relacionados

Autores	Ano	Resultados (Acurácia Média)	Método de Avaliação
Rahman *et al.* [4]	2019	SVC - 66.04% RF - 98.81%	K-fold Cross -validation utilizando K como 4, 10, e 100
Scardapane *et al.* [6]	2017	L2-NN - 98% L1-NN - 98% SG-L1-NN - 97%	Holdout - 75% treino 25% teste 25 repetições
Zhou *et al.* [9]	2017	BN-BFS - 88.25% BN-AFS - 88.30%	Holdout - 95% treino 5% teste
Grüner *et al.* [11]	2020	KNN - 99.94% RF - 99.92% ANN-3 - 99.64%	5-Fold Cross -validation
Jia *et al.* [12]	2017	SVM - 95,54% MMD - 98%	10-fold Cross -validation

V. Preparação dos Dados

Bayer *et al*[1] utilizam o método Linear Discriminant Analysis (LDA) para obter o conjunto mínimo de dimensões. LDA é um método utilizado para obter uma reta **w** no espaço de dados afim de maximizar a separação das classes[13].

Zhou *et al*[9] utilizam o método *Multi-Objective Evolutionary Algorithm based on Decomposition* (MOEA/D) para seleção de variáveis. MOEA/D é um método de seleção de características através da separação em vetores de pesos das características. O método passa por um processo evolutivo para atualização dos valores dos vetores. Utilizando a técnica de *Best Compromise Solution*, se encontra a solução ótima para o conjunto de características.

Grüner *et al.*[11] separam o conjunto de dados em treinamento e teste e utiliza Principal Component Analisys (PCA) e Recursive Feature Elimination (RFE) para redução de dimensionalidade. PCA é um método que reduz o número de variáveis projetando os dados em dimensões menores com o objetivo de encontrar as componentes principais, que são novas variáveis que melhor explicam a variância dos dados[14, p. 503]. RFE é um método recursivo que classifica as características de acordo com algum processo para classificar sua importância, onde um subconjunto de características é eliminado até que seja construída a classificação final de características[15]. O algoritmo para RFE é demonstrado no Algoritmo 1.

Algoritmo 1 Pseudo-código para o algoritmo do RFE[15]

Data: Conjunto de dados de treinamento T,
Conjunto p de características $F = \{f_1, ..., f_n\}$,
Método de classificação *M(T, F)*
Result: Classificação final R
for $i = 1$ **to** p **do**
  Classifica conjunto F com base em *M(T, F)*
  $f^* \leftarrow$ última característica classificada em F
  $R(p - i + 1) \leftarrow f^*$
  $F \leftarrow F - f^*$
end

Após o conjunto de dados ser dividido em dados e rótulos, foi realizada normalização. Grüner *et al*[11] e Zhou *et al*[9] realizam a normalização dos dados, porém os métodos utilizados não são citados. Outros artigos não mencionam a normalização dos dados, então assume-se que nenhum método foi utilizado ou que faz parte do método de aprendizado. Entretanto, foi optado em normalizar os dados, utilizando-se o método *Z-Score*.

A projeção e aplicação do PCA foi realizada a fim de analisar a influência do método nos resultados de classificação.

VI. Análise dos Dados

Os dataset não possui desbalanceamento de classes, com todas as classes contendo o mesmo número de observações.

Após a normalização, foi realizada a análise de correlação dos dados, onde obtivera m-se 19 características com correlação acima da taxa de corte de 95%, que logo após foram removidas do conjunto de dados, deixando-o com 28 características.

Após a remoção das correlações fortes, é possível analisar melhor o poder de discriminação das classes. A Figura 1 mostra um gráfico boxplot da variável **Feature10** por classe. É possível notar que as classes 4, 7 e 10 possuem um maior poder de discriminação em comparação com as demais classes. O conjunto de dados não possui outra característica com poder de discriminação mais forte.

Como citado na seção IV, Grüner *et al*[11] utilizam o método PCA para redução de dimensões, enquanto Bayer *et al*[1] utilizam LDA. Para o projeto em questão, foi optado pela utilização do método PCA, utilizando uma taxa de corte de 98% de variância acumulada. Um segundo conjunto de dados para motivos de experimento foi criado e o método foi aplicado novamente sobre este, desta vez com taxa de corte de 100% com intuito de não realizar a redução de dimensionalidade. Removendo os componentes principais com variância acumulada, obteve-se um conjunto com 24 componentes, enquanto com o conjunto sem redução obtiveram-se 28 componentes, ou seja, a quantidade de características se manteve igual ao conjunto não projetado.

Com a aplicação do PCA, a primeira componente principal explica 14.7% da variância dos dados, enquanto a segunda componente explica 10.0%. Analisando o gráfico das PC1 e PC2 na Figura 3, pode se notar quatro grupos de variáveis que estão influenciando os dados. A primeira componente principal não apresenta discriminação se comparada com a segunda, portanto é possível verificar que discriminação depois do PCA. Isso pode ser visto na Figura 3. A projeção das

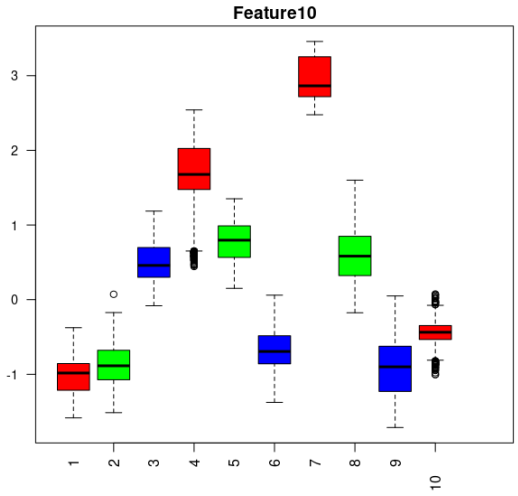

Fig. 1: Gráfico boxplot da característica **Feature10**

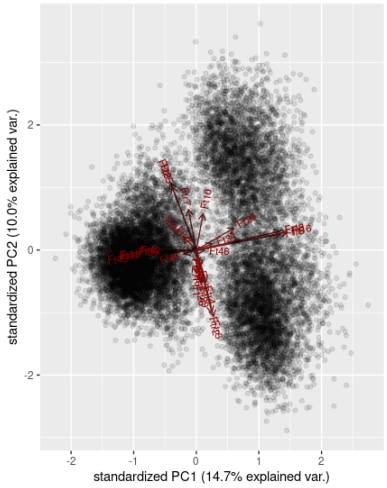

Fig. 2: Gráfico *biplot* das componentes principais 1 e 2

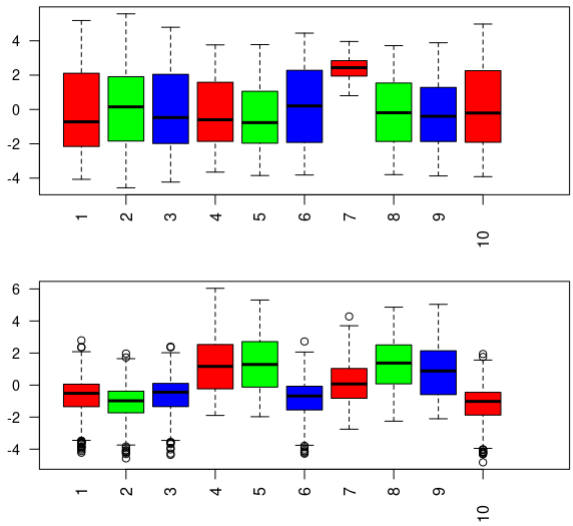

Fig. 3: Comparação entre PC1(superior) e PC2(inferior)

características para um novo espaço de dimensões não permitiu um aumento na discriminação das classes, chegando a reduzir a discriminalidade em alguns casos.

VII. Classificação

Foram selecionados para experimento os classificadores *Bagging* e Misturas Gaussianas. Tais métodos foram escolhidos por não terem sido abordados nos artigos mencionados na seção IV.

1) Bagging: O algoritmo de *bootstrap aggregation*, também chamado de *bagging* é um método de *ensemble learning*. A teoria dos métodos *ensemble* é a combinação dos resultados obtidos por diversos modelos para obter um resultado final. Tais modelos idealmente devem ser "fortes", ou seja, que possuam um histórico de alta acurácia na área em que se está aplicando. Métodos "fracos" são os modelos que possuem acurácia próxima à adivinhação aleatória, portanto, estes modelos reduzirão a acurácia do resultado final[14, p .519].

A ideia fundamental do *bagging* consiste na separação aleatória e substituição (dados previamente selecionados podem ser selecionados novamente) dos dados de treinamento em *m* subgrupos *D*, onde estes serão usados para criação de um modelo de classificação C_i treinando um algoritmo *A* em D_i. Os dados de teste serão posteriormente aplicados a cada modelo de classificação gerado, resultando em um conjunto $P_1...P_m$ de predições, das quais será calculado o resultado de predição do modelo final P_ε[14, p .525].

Os modelos $C_1...C_m$ serão aplicados a uma instância de teste **i**. O modelo com maior predição será escolhido como resultado, dado pela fórmula

$$P_\varepsilon = \underset{k}{\operatorname{argmax}} \sum_{j=1}^{m} \chi(C_j(i) = P_k)$$

[14, p .525]

2) Misturas Gaussianas: O Modelo de Misturas Gaussianas (GMM, Gaussian Mixture Model), é a soma ponderada de M densidades de componentes Gaussianos:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i g(\mathbf{x}|\mu_i, \Sigma_i),$$

[16]

Onde $\mathbf{x}$ é vetor de D dimensões, cada componente é uma função de densidade:

$$g(\mathbf{x}|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^t \Sigma_i^- 1(\mathbf{x}-\mu_i)}$$

Sendo um modelo bastante utilizado em sistemas biométricos, destaca-se pela sua facilidade de formar misturas suaves para densidades com formas arbitrárias [16]. Na literatura encontra-se referências a três tipos principais de matriz de covariância para a construção das misturas: esféricas, diagonais e completas.

As do tipo **esféricas**, são construídas em variâncias que crescem igualmente em todas dimensões [17], cada componente pode ter ou não matrizes iguais às de outros componentes. Requer um alto número de componentes e uma grande quantidade de dados para um desempenho aproximado a outros métodos [18].

As **diagonais** requerem mais componentes que uma mistura **completa**. Permitem misturas mais adaptadas aos escopos dos problemas, uma vez que as gaussianas da mistura podem assumir formas e volumes variáveis, seguindo ortogonalmente as direções dimensionais [16].

Misturas com matrizes **completas** permitem trabalhar com número de componentes menores que **esféricas** e **diagonais**, uma vez que sua adaptabilidade é maior, uma vez que suas orientações, formas e volumes podem ser variáveis [16].

Os parâmetros para *GMM* podem ser calculados utilizando o algoritmo de Maximização de Expectativa (*Expectation-maximization*), EM. De acordo com Moon[19], o algoritmo de EM é uma maneira iterativa de encontrar a função de probabilidade máxima para dados não observados, em duas diferentes etapas: estimação e maximização.

Na primeira etapa, estima-se um dado não observado utilizando um parâmetro hipotético θ^k com $k = 0$.

Na segunda etapa é computado a estimativa de probabilidade máxima para o parâmetro θ^k usando o dado estimado. Atualiza-se o parâmetro θ^k para θ^{k+1}. Itera-se os passos até atingir a convergência.

Para o seguinte experimento foi escolhido o modelo de mistura de **Matriz de Covariância Completa**, identificado no pacote R MClust com *VVV* - Volume, Forma e Orientação variáveis. Sendo feito três diferentes experimentos com *VVV*, com quantidades de Gaussianas (componentes) pré-definidos em 8, 12 e estipulados pelo pacote.

VIII. Configuração Experimental

Com base nos trabalhos de Rahman *et al.*[4], Grüner *et al.*[11] e Jia *et al.*[12], foi optado por utilizar o método de K-Fold Cross-validation, com o valor de K igual a 5, uma vez que o estudo que obteve melhores resultados utilizou este método[11].

O K-fold é um método que separa o conjunto de dados em partições denominadas de *folds*. **K** significa o número de partições a serem criadas. O modelo será treinado utilizando **K**-1 conjuntos, que será aplicado ao *fold* restante, o conjunto de validação. Este processo é repetido até todos os **K** conjuntos terem servido como conjunto de validação. A performance é a média das performances dos **K** conjuntos[14, p. 543]. A performance utilizada para a avaliação foi a acurácia dos modelos, obtendo-se a acurácia média. A Figura 4 mostra os resultados obtidos para as acurácias utilizando o método 5-fold.

IX. Resultados

Para as experimentações com Misturas Gaussianas, foi selecionado executar treinamento e predição usando modelo de mistura completo, também chamado de matriz de covariância completa [16]. Foram realizados três experimentos com três diferentes quantidades de gaussianas: estipuladas pelo pacote R **MClust** - sendo 5 o valor, 8 e 12. São considerados acurácia média e número de parâmetros como dados para discussão. Os resultados podem ser visualizados na Tabela II

Foi avaliado a utilização de PCA na etapa de preparação dos dados. Na tabela 1 pode se comparar os resultados com e sem

TABELA II: Resultados obtidos em experimentos realizados

	Sem PCA			Com PCA		
Modelo\Acurácias	Parâmetros	Desvio padrão entre as acurácias obtidas no método K-Fold	Acurácia Média	Parâmetros	Desvio padrão entre as acurácias obtidas no método K-Fold	Acurácia Média
VVV 8 Gaussianas	4223	0,370%	97,720%	2599	0,551%	97,720%
VVV 12 Gaussianas	6335	0,349%	98,234%	3899	0,205%	98,234%
VVV	2639	3,117%	92,439%	1862	3,903%	92,439%
Bagging	-	0,116%	99,010%	-	0,146%	99,010%

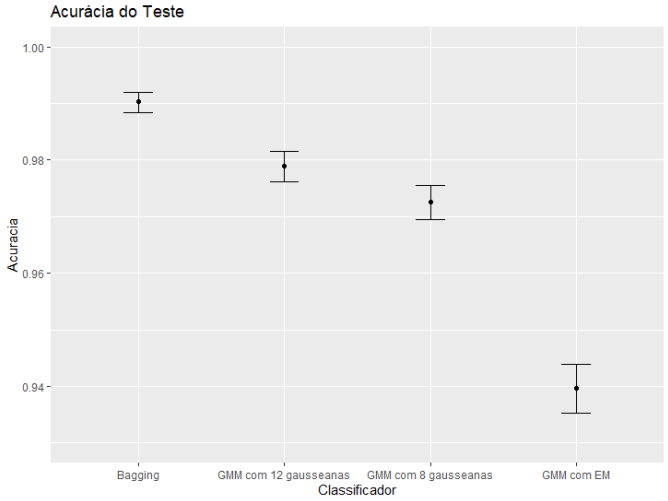

Fig. 4: Acurácias médias, superiores e inferiores para cada classificador

PCA. Destaca-se que o número de parâmetros nos métodos paramétricos diminuí em aproximadamente 40%. Nota-se que não houve melhora em termos de acurácia média. Sendo discutível a possibilidade de ganho de uma baixa porcentagem de acurácia média (em torno de 0,76%) em troca de uma ganho significante em número de parâmetros.

Em comparação com os resultados obtidos por trabalhos relacionados, apresentados na seção IV, em métodos com acurácia média foi maior que 90%, a diferença das médias das acurácias dos três experimentos com GMM e a média das acurácias dos métodos paramétricos foi 0,2% menor. Enquanto para o *bagging* em relação aos métodos não paramétricos foi 2,8% menor.

X. Conclusão

A redução de dimensões com a remoção das características fortemente correlacionadas e a projeção dos dados com PCA, permitiu que se fosse obtido uma redução de 43% no número de características do conjunto, que em seu estado inicial na análise possuía 49 variáveis, e após a aplicação dos métodos, chega a 23 variáveis. Com isso podemos concluir que houve uma redução na complexidade do problema. Entretanto, será preciso realizar a classificação do conjunto de dados e comparar os resultados para se obter uma conclusão final da eficácia da aplicação destes métodos sobre o conjunto. Os resultados obtidos com a análise exploratória permitem identificar propriedades cruciais para os próximos passos no projeto. Com o desenvolvimento atual de preparação dos dados, nas próximas etapas será trabalhado com dados normalizados por *Z-Score*.

Após o estudo e implementação das classificações, foi possível comparar com os resultados obtidos no referencial bibliográfico, seção IV, concluindo-se que os desempenhos obtidos são estatisticamente relevantes. Conclui-se também que o uso de PCA para redução de dimensionalidade e projeção de dados não trouxe o aumento de desempenho (Acurácia Média), porém reduz dos classificadores significativamente em aproximadamente 40% o número de parâmetros, perdendo aproximadamente 0,76% de acurácia média nos métodos paramétricos.

XI. Bibliografia

[1] Christian Bayer, Olaf Enge-Rosenblatt, Martyna Bator, and Uwe Monks, "Sensorless drive diagnosis using automated feature extraction, significance ranking and reduction," *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*, 2013.

[2] Martyna Bator, Alexander Dicks, Uwe Mönks, and Volker Lohweg, "Feature extraction and reduction applied to sensorless drive diagnosis," in *Proceedings. 22. Workshop Computational Intelligence, Dortmund, 6.-7. Dezember 2012*. KIT Scientific Publishing, 2014, p. 163.

[3] Dheeru Dua and Casey Graff, "UCI machine learning repository," 2017.

[4] F Rahman, R R Julviar, and I N Yulita, "Sensorless synchronous motors classification using random forest and linear support vector classifier," *IOP Conference Series: Earth and Environmental Science*, vol. 248, pp. 012061, 2019.

[5] Daniel Berrar, "Cross-validation," in *Encyclopedia of Bioinformatics and Computational Biology*, Shoba Ranganathan, Michael Gribskov, Kenta Nakai, and Christian Schönbach, Eds., pp. 542–545. Academic Press, Oxford, 2019.

[6] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini, "Group sparse regularization for deep neural networks," *Neurocomputing*, vol. 241, pp. 81–89, 2017.

[7] Aurélien Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 04 2017.

[8] Artem Oppermann, "Regularization in deep learning—l1, l2, and dropout — towards data science," .

[9] Qing Zhou, Ling He, and PengFei Lu, "Fault detection based bayesian network and moea/d applied to sensorless drive diagnosis," in *MATEC Web of Conferences*. EDP Sciences, 2017, vol. 128, p. 02017.

[10] Christopher M Bishop, "Bayesian neural networks," *Journal of the Brazilian Computer Society*, vol. 4, pp. 61–68, 1997.

[11] Tobias Grüner, Falco Böllhoff, Robert Meisetschläger, Alexander Vydrenko, Martyna Bator, Alexander Dicks, and Andreas Theissler, "Evaluation of machine learning for sensorless detection and classification of faults in electromechanical drive systems," *Procedia Computer Science*, vol. 176, pp. 1586–1595, 2020.

[12] Xiaodong Jia, Ming Zhao, Yuan Di, Qibo Yang, and Jay Lee, "Assessment of data suitability for machine prognosis using maximum mean discrepancy," *IEEE transactions on industrial electronics*, vol. 65, no. 7, pp. 5872–5881, 2017.

[13] Ethem Alpaydın, *Introduction to Machine Learning*, MIT Press, MIT.

[14] Shoba Ranganathan, Michael Gribskov, Kenta Nakai, Schonbach Christian, and Mario Cannataro, *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2019.

[15] Pablo M. Granitto, Cesare Furlanello, Franco Biasioli, and Flavia Gasperi, "Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products," *Chemometrics and Intelligent Laboratory Systems*, vol. 83, no. 2, pp. 83–90, 2006.

[16] Douglas A Reynolds, "Gaussian mixture models.," *Encyclopedia of biometrics*, vol. 741, pp. 659–663, 2009.
[17] Daniel Hsu and Sham M Kakade, "Learning mixtures of spherical gaussians: moment methods and spectral decompositions," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 2013, pp. 11–20.
[18] Alan D Marrs, "An application of reversible-jump mcmc to multivariate spherical gaussian mixtures," *Advances in neural information processing systems*, pp. 577–583, 1998.
[19] Todd K Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.