

**FEDERAL UNIVERSITY OF ITAJUBÁ - UNIFEI
POSTGRADUATE PROGRAM IN
ELECTRICAL ENGINEERING**

Electric Motor Fault Detection under
Nonstationary Operation using Koopman
Operator Spectral Analysis.

Eduardo Vaz Fagundes Rech

Itajubá, 8th July 2026

**FEDERAL UNIVERSITY OF ITAJUBÁ - UNIFEI
POSTGRADUATE PROGRAM IN
ELECTRICAL ENGINEERING**

Eduardo Vaz Fagundes Rech

**Electric Motor Fault Detection under
Nonstationary Operation using Koopman
Operator Spectral Analysis.**

Dissertation submitted to the Postgraduate Program in Electrical Engineering as part of the requirements for obtaining the title of Master of Science in Electrical Engineering.

Concentration Area: Automation and Industrial Electrical Systems

**Supervisor: Prof. Dr. Wilson Cesar Sant'Ana
Co-supervisor: Dr. Levy Ely de Lacerda de Oliveira**

**8th July 2026
Itajubá**

Acknowledgements

The completion of this dissertation marks the end of an important chapter of my academic journey, and it would not have been possible without the support, guidance, and encouragement of many people and institutions to whom I am deeply grateful.

First, I would like to express my sincere gratitude to the Federal University of Itajubá (UNIFEI) and its professors for providing an outstanding academic environment, excellent education, and the infrastructure necessary for the development of this research. I am especially thankful for the financial support provided throughout my master's studies, which allowed me to dedicate myself to research activities. I am honoured to have been part of an institution that continually invests in scientific advancement and the development of future researchers.

I would also like to express my profound appreciation to PS Soluções and Instituto Gnarus for their invaluable collaboration throughout this project. Their technical expertise, professional support, financial assistance, and provision of the experimental data used in this research were essential to the successful completion of this work. The opportunity to work closely with experienced professionals and to address real-world industrial challenges significantly enriched both the practical relevance and scientific quality of this dissertation. I am particularly grateful for the trust they placed in this research and for the countless opportunities to learn from their knowledge and experience.

My deepest gratitude goes to my supervisor, Prof. Wilson Cesar Sant'Ana, and my co-supervisor, Dr. Levy Ely de Lacerda de Oliveira. Their guidance, patience, encouragement, and unwavering support were instrumental throughout every stage of this journey. More than providing technical advice, they continuously challenged me to think critically, pursue excellence, and grow as a researcher. Their confidence in my abilities often exceeded my own and served as a constant source of motivation during the most challenging moments of this work.

I would also like to thank all professors, colleagues, researchers, and collaborators who contributed, directly or indirectly, to my academic development. Every discussion, suggestion, shared experience, and constructive criticism helped shape both this dissertation and my growth as an engineer and researcher.

A special word of gratitude is reserved for my friends and family. Their support extended far beyond the academic sphere. Through moments of uncertainty, long nights of work, deadlines, experiments, paper submissions, and revisions, they remained a constant source of encouragement, understanding, and motivation. Their belief in me made this journey lighter and more meaningful, and I am deeply grateful for their presence in my life.

Finally, I would like to thank the members of the examination committee for dedicating their time, expertise, and attention to the evaluation of this dissertation. Their comments, suggestions, and critical insights have contributed significantly to improving the quality, rigour, and clarity of this work. Beyond the formal evaluation process, their contributions represent an important step in refining ideas presented here and in helping this research reach its fullest potential.

Research is often presented as the work of an individual, but in reality it is the result of the collective effort of many people who contributed in different ways along the journey. To everyone who supported, encouraged, taught, challenged, and inspired me throughout these years, my most sincere thanks. This achievement is as much yours as it is mine.

Abstract

Accurate motor fault detection and flow rate estimation in industrial centrifugal pumps is essential for diagnostics, performance monitoring, and process optimization, yet traditional frequency-domain techniques such as Motor Current Signature Analysis (MCSA) often rely on fixed spectral bands and assume stationarity, making them less effective under variable-speed operation, changing loading conditions, and multiphase flow environments. This work introduces a Koopman-theoretic framework for motor fault and flow-rate classification based on Hankel-embedded Higher-Order Dynamic Mode Decomposition (HODMD), enabling extraction of physically interpretable spectral-dynamical features from multichannel current, voltage, and vibration measurements. The proposed pipeline computes Koopman eigenvalues, modal energies, growth rates, temporal coherence, and spectral entropy to construct a compact descriptor of pump electromechanical behaviour, which is subsequently evaluated through supervised and unsupervised learning pipelines incorporating SMOTE balancing, nonlinear UMAP projection, and extensive hyperparameter optimization via TPESearchCV. Using 2 laboratory datasets, the first consisting of 106 signals describing 4 motor faulty operation states with different input frequencies, its results were as high as 96.8% cross validation accuracy and 80.7% balanced test accuracy. The second dataset consists of 1304 signals spanning multiple rotational speeds, heterogeneous fluid compositions, and 11 unbalanced flow-rate range classes, the system achieves up to 97.8% cross-validation balanced accuracy and up to 92.6% balanced test accuracy while revealing coherent manifold structure aligned with physical flow-rate regimes. Although overlapping operating states and speed-dependent spectral shifts introduce classification challenges, the results demonstrate that Koopman spectral feature captures the essential dynamic structure of the pump more effectively than traditional Fourier-based approaches, highlighting their potential for robust, reliable, and generalizable monitoring and diagnostics of industrial pumping systems.

Key-words: Dynamic Mode Decomposition; Flow Rate Classification; Higher-Order Dynamic Mode Decomposition; Koopman Operator; Koopman Spectral Analysis; Machine Learning; Motor Current Signature Analysis; Neural Networks.

Resumo

Detecção de falhas em motores elétricos e estimação de vazão precisa em bombas centrífugas industriais é essencial para diagnóstico, monitoramento de performance e otimização de processo, porém técnicas de análise de frequência tradicionais como Análise da Assinatura de Corrente de Motor (MCSA) se baseiam em faixas espectrais fixas e assumem estacionariedade, fazendo estas técnicas inadequadas em operações com velocidade variável, mudanças de carga, e condições com fluidos multifásicos. Este trabalho introduz uma estrutura com a teoria de Koopman para detecção de falhas em motores elétricos e classificação de taxa de vazão, baseado em Matrizes de Hankel, Higher-Order Dynamic Mode Decomposition (HODMD), habilitando a extração de características com interpretação física do espectro dinâmico de um sistema multicanal composto por medidas de corrente, tensão e vibração. A sequência de processamento proposta computa autovalores de Koopman, as energias de cada modo dinâmico, taxas de crescimento, coerência temporal e entropia espectral para construir um descritor compacto do comportamento eletromecânico do motor, o qual é posteriormente avaliado por meio de algoritmos de aprendizado supervisionado e não supervisionado, incorporando balanceamento de classes via SMOTE, projeção não linear via UMAP e extensiva otimização de hiperparâmetros via TPESearchCV. Utilizando dois bancos de dados, o primeiro consistindo de 106 sinais descrevendo 4 estados de falha distintos com diferentes frequências de alimentação, alcançando resultados tão altos como 96.8% de acurácia balanceada em validação cruzada e 80.7% de acurácia balanceada em sinais de teste. O segundo banco de dados contém 1304 sinais, abrangendo múltiplas velocidades de rotação, composição de fluidos heterogêneos, e 11 classes desbalanceadas de bandas de vazão, o sistema conquista até 97.8% de acurácia balanceada em validação cruzada e 92.6% de acurácia balanceada em sinais de teste enquanto revela uma *manifold* com estrutura coerente e alinhada com os regimes de taxa de vazão físicos. Enquanto estados de operação sobrepostos e deslocamentos espectrais dependentes de velocidade introduz desafios de classificação, os resultados demonstram que as características do espectro de Koopman extraídos capturam a estrutura dinâmica essencial da bomba de forma mais eficaz do que abordagens tradicionais baseadas no espectro de Fourier, destacando seu potencial para monitoramento e diagnóstico robusto, confiável e generalizável de sistemas industriais motorizados.

Palavras-chave: Análise da Assinatura de Corrente de Motores; Análise Espectral; Aprendizado de Máquina; Classificação de vazão; Dynamic Mode Decomposition; ; Higher-Order Dynamic Mode Decomposition; Operador de Koopman; Redes Neurais.

List of Figures

Figure 1	Motor System Testbench	52
Figure 2	Motor Fault Dataset Distribution	53
Figure 3	Motor Fault Classification Flowchart	54
Figure 4	Flow Rate Dataset Distribution	57
Figure 5	Pump System Testbench - Source: Teste de desempenho de BCS e monitoramento de parâmetros elétricos - CEPETRO (2025) .	58
Figure 6	Pump System Layout - Source: Teste de desempenho de BCS e monitoramento de parâmetros elétricos - CEPETRO (2025)	59
Figure 7	Flow Rate Classification Flowchart	61
Figure 8	Healthy Spectrum	64
Figure 9	Broken Bars Spectrum	65
Figure 10	Inner-turn Short-Circuit Spectrum	66
Figure 11	Unbalance Spectrum	67
Figure 12	k-NN Confusion Matrix - Raw	70
Figure 13	k-NN SHAP Values Summary - Raw	70
Figure 14	k-NN Learning & Validation Curves - Raw	71
Figure 15	k-NN Confusion Matrix - FFT	71
Figure 16	k-NN SHAP Values Summary - FFT	72
Figure 17	k-NN Learning & Validation Curves - FFT	72
Figure 18	k-NN Confusion Matrix - Koopman	73
Figure 19	k-NN SHAP Values Summary - Koopman	73
Figure 20	k-NN Learning & Validation Curves - Koopman	74
Figure 21	SVM Confusion Matrix - Raw	76
Figure 22	SVM SHAP Values Summary - Raw	76
Figure 23	SVM Learning & Validation Curves - Raw	77
Figure 24	SVM Confusion Matrix - FFT	77
Figure 25	SVM SHAP Values Summary - FFT	78
Figure 26	SVM Learning & Validation Curves - FFT	78
Figure 27	SVM Confusion Matrix - Koopman	79
Figure 28	SVM SHAP Values Summary - Koopman	79
Figure 29	SVM Learning & Validation Curves - Koopman	80
Figure 30	Random Forest Confusion Matrix - Raw	82
Figure 31	Random Forest SHAP Values Summary - Raw	82
Figure 32	Random Forest Learning & Validation Curves - Raw	83
Figure 33	Random Forest Confusion Matrix - FFT	83
Figure 34	Random Forest SHAP Values Summary - FFT	84

Figure 35	Random Forest Learning & Validation Curves - FFT	84
Figure 36	Random Forest Confusion Matrix - Koopman	85
Figure 37	Random Forest SHAP Values Summary - Koopman	85
Figure 38	Random Forest Learning & Validation Curves - Koopman . .	86
Figure 39	Logistic Regression Confusion Matrix - Raw	88
Figure 40	Logistic Regression SHAP Values Summary - Raw	88
Figure 41	Logistic Regression Learning & Validation Curves - Raw . . .	89
Figure 42	Logistic Regression Confusion Matrix - FFT	89
Figure 43	Logistic Regression SHAP Values Summary - FFT	90
Figure 44	Logistic Regression Learning & Validation Curves - FFT . . .	90
Figure 45	Logistic Regression Confusion Matrix - Koopman	91
Figure 46	Logistic Regression SHAP Values Summary - Koopman . . .	91
Figure 47	Logistic Regression Learning & Validation Curves - Koopman	92
Figure 48	Gaussian Naive Bayes Confusion Matrix - Raw	93
Figure 49	Gaussian Naive Bayes SHAP Values Summary - Raw	94
Figure 50	Gaussian Naive Bayes Learning & Validation Curves - Raw .	94
Figure 51	Gaussian Naive Bayes Confusion Matrix - FFT	95
Figure 52	Gaussian Naive Bayes SHAP Values Summary - FFT	95
Figure 53	Gaussian Naive Bayes Learning & Validation Curves - FFT .	96
Figure 54	Gaussian Naive Bayes Confusion Matrix - Koopman	96
Figure 55	Gaussian Naive Bayes SHAP Values Summary - Koopman . .	97
Figure 56	Gaussian Naive Bayes Learning & Validation Curves - Koopman	97
Figure 57	XGBoost Confusion Matrix - Raw	99
Figure 58	XGBoost SHAP Values Summary - Raw	99
Figure 59	XGBoost Learning & Validation Curves - Raw	100
Figure 60	XGBoost Confusion Matrix - FFT	100
Figure 61	XGBoost SHAP Values Summary - FFT	101
Figure 62	XGBoost Learning & Validation Curves - FFT	101
Figure 63	XGBoost Confusion Matrix - Koopman	102
Figure 64	XGBoost SHAP Values Summary - Koopman	102
Figure 65	XGBoost Learning & Validation Curves - Koopman	103
Figure 66	MLP Confusion Matrix - Raw	105
Figure 67	MLP SHAP Values Summary - Raw	105
Figure 68	MLP Learning & Validation Curves - Raw	106
Figure 69	MLP Confusion Matrix - FFT	106
Figure 70	MLP SHAP Values Summary - FFT	107
Figure 71	MLP Learning & Validation Curves - FFT	107
Figure 72	MLP Confusion Matrix - Koopman	108
Figure 73	MLP SHAP Values Summary - Koopman	108

Figure 74	MLP Learning & Validation Curves - Koopman	109
Figure 75	k-NN Confusion Matrix - Raw	112
Figure 76	k-NN SHAP Values Summary - Raw	112
Figure 77	k-NN Learning & Validation Curves - Raw	113
Figure 78	k-NN Confusion Matrix - FFT	113
Figure 79	k-NN SHAP Values Summary - FFT	114
Figure 80	k-NN Learning & Validation Curves - FFT	114
Figure 81	k-NN Confusion Matrix - Koopman	115
Figure 82	k-NN SHAP Values Summary - Koopman	115
Figure 83	k-NN Learning & Validation Curves - Koopman	116
Figure 84	SVM Confusion Matrix - Raw	118
Figure 85	SVM SHAP Values Summary - Raw	118
Figure 86	SVM Learning & Validation Curves - Raw	119
Figure 87	SVM Confusion Matrix - FFT	119
Figure 88	SVM SHAP Values Summary - FFT	120
Figure 89	SVM Learning & Validation Curves - FFT	120
Figure 90	SVM Confusion Matrix - Koopman	121
Figure 91	SVM SHAP Values Summary - Koopman	121
Figure 92	SVM Learning & Validation Curves - Koopman	122
Figure 93	Random Forest Confusion Matrix - Raw	124
Figure 94	Random Forest SHAP Values Summary - Raw	124
Figure 95	Random Forest Learning & Validation Curves - Raw	125
Figure 96	Random Forest Confusion Matrix - FFT	125
Figure 97	Random Forest SHAP Values Summary - FFT	126
Figure 98	Random Forest Learning & Validation Curves - FFT	126
Figure 99	Random Forest Confusion Matrix - Koopman	127
Figure 100	Random Forest SHAP Values Summary - Koopman	127
Figure 101	Random Forest Learning & Validation Curves - Koopman	128
Figure 102	Logistic Regression Confusion Matrix - Raw	130
Figure 103	Logistic Regression SHAP Values Summary - Raw	130
Figure 104	Logistic Regression Learning & Validation Curves - Raw	131
Figure 105	Logistic Regression Confusion Matrix - FFT	131
Figure 106	Logistic Regression SHAP Values Summary - FFT	132
Figure 107	Logistic Regression Learning & Validation Curves - FFT	132
Figure 108	Logistic Regression Confusion Matrix - Koopman	133
Figure 109	Logistic Regression SHAP Values Summary - Koopman	133
Figure 110	Logistic Regression Learning & Validation Curves - Koopman	134
Figure 111	Gaussian Naive Bayes Confusion Matrix - Raw	136
Figure 112	Gaussian Naive Bayes SHAP Values Summary - Raw	136

Figure 113	Gaussian Naive Bayes Learning & Validation Curves - Raw .	137
Figure 114	Gaussian Naive Bayes Confusion Matrix - FFT	137
Figure 115	Gaussian Naive Bayes SHAP Values Summary - FFT	138
Figure 116	Gaussian Naive Bayes Learning & Validation Curves - FFT .	138
Figure 117	Gaussian Naive Bayes Confusion Matrix - Koopman	139
Figure 118	Gaussian Naive Bayes SHAP Values Summary - Koopman . .	139
Figure 119	Gaussian Naive Bayes Learning & Validation Curves - Koopman	140
Figure 120	XGBoost Confusion Matrix - Raw	142
Figure 121	XGBoost SHAP Values Summary - Raw	142
Figure 122	XGBoost Learning & Validation Curves - Raw	143
Figure 123	XGBoost Confusion Matrix - FFT	143
Figure 124	XGBoost SHAP Values Summary - FFT	144
Figure 125	XGBoost Learning & Validation Curves - FFT	144
Figure 126	XGBoost Confusion Matrix - Koopman	145
Figure 127	XGBoost SHAP Values Summary - Koopman	145
Figure 128	XGBoost Learning & Validation Curves - Koopman	146
Figure 129	MLP Confusion Matrix - Raw	148
Figure 130	MLP SHAP Values Summary - Raw	148
Figure 131	MLP Learning & Validation Curves - Raw	149
Figure 132	MLP Confusion Matrix - FFT	149
Figure 133	MLP SHAP Values Summary - FFT	150
Figure 134	MLP Learning & Validation Curves - FFT	150
Figure 135	MLP Confusion Matrix - Koopman	151
Figure 136	MLP SHAP Values Summary - Koopman	151
Figure 137	MLP Learning & Validation Curves - Koopman	152
Figure 138	k-NN Confusion Matrix - C	154
Figure 139	k-NN SHAP Values Summary - C	155
Figure 140	k-NN Learning & Validation Curves - C	155
Figure 141	SVM Confusion Matrix - C	156
Figure 142	SVM SHAP Values Summary - C	156
Figure 143	SVM Learning & Validation Curves - C	157
Figure 144	Random Forest Confusion Matrix - C	158
Figure 145	Random Forest SHAP Values Summary - C	158
Figure 146	Random Forest Learning & Validation Curves - C	159
Figure 147	Logistic Regression Confusion Matrix - C	160
Figure 148	Logistic Regression SHAP Values Summary - C	161
Figure 149	Logistic Regression Learning & Validation Curves - C	162
Figure 150	GNB Confusion Matrix - C	163
Figure 151	GNB SHAP Values Summary - C	164

Figure 152	GNB Learning & Validation Curves - C	165
Figure 153	XGBoost Confusion Matrix - C	166
Figure 154	XGBoost SHAP Values Summary - C	167
Figure 155	XGBoost Learning & Validation Curves - C	167
Figure 156	MLP Confusion Matrix - C	168
Figure 157	MLP SHAP Values Summary - C	169
Figure 158	MLP Learning & Validation Curves - C	169
Figure 159	GMM Confusion Matrix - C	171
Figure 160	GMM SHAP Values Summary - C	172
Figure 161	KMeans Confusion Matrix - C	173
Figure 162	KMeans SHAP Values Summary - C	173
Figure 163	HDBSCAN Confusion Matrix - C	174
Figure 164	HDBSCAN SHAP Values Summary - C	174
Figure 165	Agglomerative Clustering Confusion Matrix - C	175
Figure 166	Agglomerative Clustering SHAP Values Summary - C	175
Figure 167	Spectral Clustering Confusion Matrix - C	176
Figure 168	Spectral Clustering SHAP Values Summary - C	176
Figure 169	Mean Shift Confusion Matrix - C	177
Figure 170	Mean Shift SHAP Values Summary - C	177
Figure 171	k-NN Confusion Matrix - V	179
Figure 172	k-NN SHAP Values Summary - V	180
Figure 173	k-NN Learning & Validation Curves - V	180
Figure 174	SVM Confusion Matrix - V	181
Figure 175	SVM SHAP Values Summary - V	182
Figure 176	SVM Learning & Validation Curves - V	183
Figure 177	Random Forest Confusion Matrix - V	184
Figure 178	Random Forest SHAP Values Summary - V	184
Figure 179	Random Forest Learning & Validation Curves - V	185
Figure 180	Logistic Regression Confusion Matrix - V	186
Figure 181	Logistic Regression SHAP Values Summary - V	187
Figure 182	Logistic Regression Learning & Validation Curves - V	188
Figure 183	GNB Confusion Matrix - V	189
Figure 184	GNB SHAP Values Summary - V	190
Figure 185	GNB Learning & Validation Curves - V	191
Figure 186	XGBoost Confusion Matrix - V	192
Figure 187	XGBoost SHAP Values Summary - V	192
Figure 188	XGBoost Learning & Validation Curves - V	193
Figure 189	MLP Confusion Matrix - V	194
Figure 190	MLP SHAP Values Summary - V	195

Figure 191	MLP Learning & Validation Curves - V	195
Figure 192	GMM Confusion Matrix - V	197
Figure 193	GMM SHAP Values Summary - V	197
Figure 194	KMeans Confusion Matrix - V	198
Figure 195	KMeans SHAP Values Summary - V	198
Figure 196	HDBSCAN Confusion Matrix - V	199
Figure 197	HDBSCAN SHAP Values Summary - V	199
Figure 198	Agglomerative Clustering Confusion Matrix - V	200
Figure 199	Agglomerative Clustering SHAP Values Summary - V	201
Figure 200	Spectral Clustering Confusion Matrix - V	202
Figure 201	Spectral Clustering SHAP Values Summary - V	202
Figure 202	Mean Shift Confusion Matrix - V	203
Figure 203	Mean Shift SHAP Values Summary - V	203
Figure 204	k-NN Confusion Matrix - CV	205
Figure 205	k-NN SHAP Values Summary - CV	206
Figure 206	k-NN Learning & Validation Curves - CV	207
Figure 207	SVM Confusion Matrix - CV	208
Figure 208	SVM SHAP Values Summary - CV	208
Figure 209	SVM Learning & Validation Curves - CV	209
Figure 210	Random Forest Confusion Matrix - CV	210
Figure 211	Random Forest SHAP Values Summary - CV	210
Figure 212	Random Forest Learning & Validation Curves - CV	211
Figure 213	Logistic Regression Confusion Matrix - CV	212
Figure 214	Logistic Regression SHAP Values Summary - CV	213
Figure 215	Logistic Regression Learning & Validation Curves - CV . .	214
Figure 216	GNB Confusion Matrix - CV	215
Figure 217	GNB SHAP Values Summary - CV	216
Figure 218	GNB Learning & Validation Curves - CV	217
Figure 219	XGBoost Confusion Matrix - CV	218
Figure 220	XGBoost SHAP Values Summary - CV	219
Figure 221	XGBoost Learning & Validation Curves - CV	219
Figure 222	MLP Confusion Matrix - CV	220
Figure 223	MLP SHAP Values Summary - CV	221
Figure 224	MLP Learning & Validation Curves - CV	222
Figure 225	GMM Confusion Matrix - CV	224
Figure 226	GMM SHAP Values Summary - CV	224
Figure 227	KMeans Confusion Matrix - CV	225
Figure 228	KMeans SHAP Values Summary - CV	225
Figure 229	HDBSCAN Confusion Matrix - CV	226

Figure 230	HDBSCAN SHAP Values Summary - CV	226
Figure 231	Agglomerative Clustering Confusion Matrix - CV	227
Figure 232	Agglomerative Clustering SHAP Values Summary - CV	227
Figure 233	Spectral Clustering Confusion Matrix - CV	228
Figure 234	Spectral Clustering SHAP Values Summary - CV	229
Figure 235	Mean Shift Confusion Matrix - CV	230
Figure 236	Mean Shift SHAP Values Summary - CV	230
Figure 237	k-NN Confusion Matrix - CTV	232
Figure 238	k-NN SHAP Values Summary - CTV	232
Figure 239	k-NN Learning & Validation Curves - CTV	233
Figure 240	SVM Confusion Matrix - CTV	234
Figure 241	SVM SHAP Values Summary - CTV	234
Figure 242	SVM Learning & Validation Curves - CTV	235
Figure 243	Random Forest Confusion Matrix - CTV	236
Figure 244	Random Forest SHAP Values Summary - CTV	236
Figure 245	Random Forest Learning & Validation Curves - CTV	237
Figure 246	Logistic Regression Confusion Matrix - CTV	238
Figure 247	Logistic Regression SHAP Values Summary - CTV	239
Figure 248	Logistic Regression Learning & Validation Curves - CTV	240
Figure 249	GNB Confusion Matrix - CTV	241
Figure 250	GNB SHAP Values Summary - CTV	242
Figure 251	GNB Learning & Validation Curves - CTV	243
Figure 252	XGBoost Confusion Matrix - CTV	244
Figure 253	XGBoost SHAP Values Summary - CTV	245
Figure 254	XGBoost Learning & Validation Curves - CTV	245
Figure 255	MLP Confusion Matrix - CTV	246
Figure 256	MLP SHAP Values Summary - CTV	247
Figure 257	MLP Learning & Validation Curves - CTV	247
Figure 258	GMM Confusion Matrix - CTV	249
Figure 259	GMM SHAP Values Summary - CTV	250
Figure 260	KMeans Confusion Matrix - CTV	251
Figure 261	KMeans SHAP Values Summary - CTV	251
Figure 262	HDBSCAN Confusion Matrix - CTV	252
Figure 263	HDBSCAN SHAP Values Summary - CTV	252
Figure 264	Agglomerative Clustering Confusion Matrix - CTV	253
Figure 265	Agglomerative Clustering SHAP Values Summary - CTV	253
Figure 266	Spectral Clustering Confusion Matrix - CTV	254
Figure 267	Spectral Clustering SHAP Values Summary - CTV	254
Figure 268	Mean Shift Confusion Matrix - CTV	255

Figure 269	Mean Shift SHAP Values Summary - CTV	255
------------	--	-----

List of Tables

Table 1	Hyperparameter Search Space Evaluated in the Motor Supervised Pipeline	55
Table 2	Hyperparameter Search Space Evaluated in the Flow Rate Supervised Pipeline	60
Table 3	Hyperparameter Search Space Evaluated in the Flow Rate Unsupervised Pipeline	62
Table 4	Raw Signal Input - Unbalance	68
Table 5	Fourier Signal Input - Unbalance	68
Table 6	Koopman Signal Input - Unbalance	68
Table 7	Raw Signal Input - Screws	110
Table 8	Fourier Signal Input - Screws	110
Table 9	Koopman Signal Input - Screws	110
Table 10	Supervised Learning Results - C	153
Table 11	Unsupervised Learning Results - C	170
Table 12	Supervised Learning Results - V	178
Table 13	Unsupervised Learning Results - V	196
Table 14	Supervised Learning Results - CV	204
Table 15	Unsupervised Learning Results - CV	223
Table 16	Supervised Learning Results - CTV	231
Table 17	Unsupervised Learning Results - CTV	248

List of Acronyms

ARI	Adjusted Rand Index	46
AUC	Area Under the Curve	45
CEEMDAN	Complete Ensemble Empirical Mode Decomposition with Adaptive Noise	21
CH	Calinski-Harabasz Index	46
CV	Cross-Validation	43
DBI	Davies-Bouldin Index	46
DBSCAN	Density-Based Spatial Clustering	39
DMD	Dynamic Mode Decomposition	22
DOE	Design of Experiments	256
FFT	Fast Fourier Transform	21
GMM	Gaussian Mixture Model	39
GNB	Gaussian Naive Bayes	37
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise	39
HODMD	Higher-Order Dynamic Mode Decomposition	22
HOSVD	Higher-Order Singular Value Decomposition	31
IMF	Intrinsic Mode Function	21
k-NN	k-Nearest Neighbours	36
KDE	Kernel Density Estimate	41
LDA	Linear Discriminant Analysis	42
MCSA	Motor Current Signature Analysis	20
MLP	Multi-Layer Perceptron	38
OGI	Optical Gas Imaging	21
PCA	Principal Component Analysis	42
RBF	Radial Basis Function	36
RF	Random Forest	37
ROC	Receiver Operating Characteristic	45
SHAP	Shapley Additive Explanations	44
SMOTE	Synthetic Minority Oversampling Technique	41
SVM	Support Vector Machine	36
TPE	Tree-structured Parzen Estimator	43
UMAP	Uniform Manifold Approximation and Projection	42
XGBoost	Extreme Gradient Boosting	37

List of Symbols

A	DMD Approximation of the Koopman Operator	31
D	Pitch Diameter	27
H	Hankel Matrix representing a Time-Delayed State Vector	28
K	Koopman Operator	29
N_b	Number of Balls in Bearing	27
R	Number of Rotor Slots	27
U	Left Singular Vectors of X	31
V	Right Singular Vectors of X	31
X	Snapshot Matrix of a signal $x(t)$	30
$X^\dagger$	Moore-Penrose Pseudoinverse of X	30
Σ	Diagonal Matrix with the Singular Values of X	31
β	Contact Angle of the Bearing	27
λ_k	Koopman's k th Eigenvalue	29
ν	Order of Stator Harmonics	27
ω_k	Koopman's k th Mode Frequency	29
ϕ_k	Koopman's k th Mode	29
σ_k	Koopman's k th Mode Grow/Decay Rate	29
d	Hankel Embedding Dimension	47
d_A	System's Compact Attractor Dimensions	28
d_B	Diameter of Ball	27
f_b	Frequency of Ball Defect	27
f_c	Characteristic Vibration Frequency of Cage Failure	27
f_i	Frequency of Inner Race	27
f_o	Frequency of Outer Race	27
f_r	Rotor Speed	27
f_s	Supply Frequency	26
$g(x)$	System Dynamics Function	29
n_d	Eccentricity order	27
p	Number of Pole Pairs	27
r	Retained Rank	47
s	Rotor Slip	26

Contents

List of Figures7

1	INTRODUCTION	21
1.1	Motivation	24
1.2	Research Objectives	24
1.3	Work Structure	26
2	THEORETICAL REVIEW	27
2.1	Motor Current Signature Analysis and Fault Frequencies	27
2.2	Nonlinear Systems Analysis	29
2.2.1	Takens' Embedding Theorem	29
2.2.2	Time-Delay Embedding and the Hankel Matrix	29
2.2.3	Koopman Operator Theory	30
2.2.4	Dynamic Mode Decomposition	31
2.2.5	Higher-Order Dynamic Mode Decomposition	32
2.2.6	Understanding HODMD Through Classical Concepts	33
2.3	Classification Algorithms	36
2.3.1	Supervised Learning Methods	37
2.3.1.1	k-Nearest Neighbours	37
2.3.1.2	Support Vector Machines	37
2.3.1.3	Random Forests	38
2.3.1.4	Logistic Regression	38
2.3.1.5	Gaussian Naive Bayes	38
2.3.1.6	Extreme Gradient Boosting	38
2.3.1.7	Multi-Layer Perceptron	39
2.3.2	Unsupervised Learning Methods	39
2.3.2.1	k-Means Clustering	39
2.3.2.2	Gaussian Mixture Models	40
2.3.2.3	Hierarchical Density-Based Spatial Clustering	40
2.3.2.4	Agglomerative Hierarchical Clustering	41
2.3.2.5	Spectral Clustering	41
2.3.2.6	Mean Shift Clustering	42
2.3.3	Model Selection and Performance Evaluation	42
2.3.3.1	Synthetic Minority Oversampling Technique	42
2.3.3.2	Uniform Manifold Approximation and Projection	43
2.3.3.3	Efficient Hyperparameter Optimization	43

2.3.3.4	Stratified K-Fold Cross-Validation	44
2.3.3.5	Shapley Additive Explanations	45
2.3.3.6	Evaluation Metrics for Supervised Learning	45
2.3.3.7	Evaluation Metrics for Unsupervised Learning	46
3	METHODOLOGY: PROPOSED PIPELINE	48
3.1	Nonstationarity Handling Process	48
3.2	Koopman Feature Extraction Pipeline	49
3.3	Fourier vs Koopman Spectrum & Fault Detection	50
3.3.1	Dataset Description and Experimental Conditions	51
3.3.2	Supervised Classification Pipeline	52
3.4	Flow Rate Classification in Centrifugal Pump Systems	56
3.4.1	Dataset Description and Experimental Conditions	56
3.4.2	Supervised Classification Pipeline	58
3.4.3	Unsupervised Classification Pipeline	61
4	RESULTS AND DISCUSSION	63
4.1	Motor Fault Detection Results	63
4.1.1	Spectrum Comparison	63
4.1.2	Fault Detection with Unbalance Label	68
4.1.3	Fault Detection with Screws Labels	110
4.2	Classification Pipeline Results and Evaluation	153
4.2.1	Only 3 Currents (C)	153
4.2.1.1	Supervised Pipeline	153
4.2.1.2	Unsupervised Learning	170
4.2.2	Only 2 Vibrational Axes (V)	178
4.2.2.1	Supervised Pipeline	178
4.2.2.2	Unsupervised Learning	196
4.2.3	3 Currents + 2 Vibrational Axes (CV)	204
4.2.3.1	Supervised Pipeline	204
4.2.3.2	Unsupervised Learning	223
4.2.4	3 Currents + 3 Voltages + 2 Vibrational Axes (CTV)	231
4.2.4.1	Supervised Pipeline	231
4.2.4.2	Unsupervised Learning	248
5	CONCLUSION AND DISCUSSION	256
5.1	Study Limitations & Future Works	258
	BIBLIOGRAPHY	261

A	ATTACHMENTS	267
A.1	Published Papers	267
A.1.1	E. V. F. Rech, L. E. de L. de Oliveira, and W. C. Sant'Ana, "Detection of Nonstationary Events in Motor Signals Using Nonparametric Statistical Hypothesis Testing", IEEE Access, vol. 13, 29/09/2025	267
A.1.2	E. V. F. Rech, L. E. de L. de Oliveira, and W. C. Sant'Ana, "Flow Rate Classification in Industrial Pump Systems Using Koopman-Based Feature Extraction and Machine Learning- IEEE Access, accepted for publication on 26/05/2026	267
A.2	Submitted Papers	267
A.2.1	E. V. F. Rech, L. E. de L. de Oliveira, and W. C. Sant'Ana, "Electric Motor Fault Detection under Nonstationary Operation using Koopman Operator Spectral Analysis- Submitted to IEEE Access, 16/05/2026	267

1 Introduction

Industrial electric motor-driven systems constitute the backbone of modern manufacturing, energy production, water treatment, and process industries [1]. Among these systems, centrifugal pumps driven by induction motors are particularly critical due to their widespread use in fluid transport, chemical processing, cooling circuits, and hydraulic systems [1][2]. Ensuring reliable operation of such systems is essential not only for maintaining production continuity but also for preventing costly downtime, mechanical degradation, and catastrophic failure [3]. Consequently, the development of advanced monitoring and diagnostic methodologies for motor–pump assemblies has become an important research topic within electrical engineering and industrial automation. In previous work[4], the author proposed a statistical framework for early nonstationary event detection in induction motor signals based on hypothesis testing, demonstrating the importance of identifying subtle deviations in machine operating behaviour as an initial step toward reliable condition monitoring and fault diagnosis.

Traditionally, motor condition monitoring and fault detection rely on vibration analysis, temperature sensing, and direct mechanical measurements [5]. In parallel, Motor Current Signature Analysis ([MCSA \(Motor Current Signature Analysis\)](#)) has emerged as a non-intrusive alternative, enabling the detection of electrical and mechanical anomalies through spectral analysis of stator currents. [MCSA](#) is attractive because it requires no additional sensors beyond those already present in most industrial installations [6][7]. However, conventional approaches are predominantly based on Fourier-domain analysis, which assumes stationarity and focuses on fixed spectral bands associated with known fault frequencies. While effective in well-defined and steady operating conditions, such methods exhibit limitations when systems operate under variable speeds, non-linear loads, multiphase flow conditions, or coupled electromechanical interactions [4][8].

Centrifugal pumps represent a particularly challenging case. The electrical signals measured at the motor terminals are influenced not only by internal motor dynamics but also by hydraulic loading, cavitation phenomena, flow turbulence, and fluid composition [9][10][11]. These interactions produce non-stationary, amplitude-modulated, and speed-dependent spectral characteristics that are not adequately captured by purely Fourier-based descriptors. Furthermore, in coupled motor–pump systems, similar operating conditions may produce overlapping spectral signatures, complicating both fault detection and operational state identification [10][11].

These pumps’ performance and reliability directly influence production efficiency, energy consumption, and operational safety in sectors such as oil and gas, water treatment,

chemical processing, and power generation [1][2]. Because of this critical role, real-time knowledge of the pump's operating state - particularly the flow regime - is essential for ensuring stable operation, detecting anomalies, and optimizing system control [9][10].

In practice, flow information is typically obtained from intrusive sensors such as flow meters, pressure transducers, or vibration probes [10][11]. While effective, these sensors introduce several limitations: they require installation and maintenance inside the hydraulic circuit, suffer from fouling or drift in harsh fluids, increase system cost, and may fail under multiphase conditions such as gas-entrained flow [10][12]. These drawbacks motivate the study of pump's flow behaviour through indirect but robust measurements.

Recent MCSA-based fault detection methods for induction motors have explored advanced signal processing and intelligent classification techniques. Garcia-Bracamonte et al. combined Independent Component Analysis and Neural Networks for broken bar detection, achieving high accuracy but relying on a handcrafted spectral segmentation [13]. Song et al. used CEEMDAN (Complete Ensemble Empirical Mode Decomposition with Adaptive Noise) to enhance bearing fault visibility in stator currents, though the method is still sensitive to noise and requires manual IMF (Intrinsic Mode Function) selection [14]. Zamudio-Ramirez et al. proposed a novel approach using stray flux signals and dimensionality reduction for detecting gearbox wear, but it requires additional sensors not typically available in standard motor setups [15]. Yelpale et al. developed a fuzzy decision system to switch between FFT (Fast Fourier Transform) and Wavelet analysis based on motor load and signal behaviour, improving efficiency but dependent on accurate fuzzy rules [16]. Liu et al. introduced a beam-forming-based spectrum method robust to varying loads and noise, but it adds computational complexity and parameter sensitivity [17]. All these methods aim to create a robust diagnosis for nonstationary MCSA signals at the cost of computational complexity, sensitivity, and poor generalization for different operation conditions, which is especially bad if the input is a stationary MCSA signal.

In the flow-rate estimation front, recent developed techniques include the work of Jinasena et al used reduced order model of a Venturi channel and Kalman filters as state estimators to achieve a mean absolute error of 2% in the estimations, but relies on the availability of the Venturi channel to take fluid level measurements which is not always a given [12]. Shen et al made a comparison of multiple optical flow estimation algorithms, showing they can get very accurate speed estimates, but they need Optical Gas Imaging (OGI (Optical Gas Imaging)) cameras with very high capture rate and the fluid be visible to the camera which might be difficult in a lot of real-world industrial pumps [18]. Dinardo et al used laser Doppler vibrometer sensors to sense the accelerations inducted by flow to the pipe walls and showed how the flow-rate affects the amplitude of the first harmonic of the measurements, but it needs an exposed point on a pipe where is possible to sense the transverse acceleration of the flow induced on the pipe walls [19]. Li et al proposed a multi-

constraint framework for ultrasound velocity field estimation that combines Doppler and optical flow constraints with smoothness regularization, enhancing the accuracy of cross-range velocity estimation, but requires multiple ultrasound images to work [20]. Jiang et al proposed a Transformer neural network with encoder layers to extract flow rate features of a two-phase gas-liquid fluid using pressure, temperate and voltage as inputs, but relies in the black box approach for features extraction, multiple different measurements and its self-attention mechanism be correctly tuned [10]. All those methods rely on intrusive methods or certain system conditions to be met, which might not be possible to apply them in submerged centrifugal pumps in real-world industrial conditions.

To address this limitations, recent research has explored data-driven and operator-theoretic approaches capable of extracting richer dynamical information from measured signals. In particular, Koopman operator theory provides a framework for representing nonlinear dynamical systems through linear evolution in a higher-dimensional observable space [21][22][23]. Practical approximations such as Dynamic Mode Decomposition (DMD (Dynamic Mode Decomposition)) and Higher-Order Dynamic Mode Decomposition (HODMD (Higher-Order Dynamic Mode Decomposition)) enable the extraction of dominant spectral components, growth/decay rates, and coherent structures directly from time-series data [24][25][26]. Unlike traditional Fourier analysis, Koopman-based methods characterize not only oscillatory content but also modal energy distribution, temporal coherence, and dynamic stability properties [27][28][29][30].

In the electrical engineering domain, the Koopman related research has focused on power system analysis Susuki et al. presented one of the first comprehensive discussion regarding the application of Koopman operator theory to power systems, demonstrating its usefulness for coherency identification, oscillatory analysis, precursor instability detection, and stability assessment directly from measurement data [31]. Subsequent studies extended these ideas toward dynamic state estimation, nonlinear observability analysis [32], inertia estimation, oscillation localization [33], uncertainty propagation [34], and transient prediction in large-scale electrical networks as well in microgrids [35]. These works collectively demonstrated that Koopman-based representations are particularly attractive for strongly nonlinear electromechanical systems because they preserve physically meaningful modal information while avoiding the need for highly detailed first-principles models [31].

Despite these advances, comparatively little attention has been given to the application of Koopman-based methodologies to induction motors and industrial condition monitoring problems. Existing induction motor faults diagnosis methods remain predominantly based on classical MCSA, Fourier techniques, wavelet decomposition, and machine learning approaches operating on handcrafted statistical or spectral descriptors [3][5]. Traditional methodologies have shown considerable success for stationary operating condi-

ons but often experience performance degradation in the presence of nonlinear couplings, transient regimes, and variable-speed operation.

More importantly, although Rajendran et al. proposed using DMD for fault diagnosis in three-phase electrical machines to extract modulation frequencies of unbalanced in three-phase signal [36], the direct use of Higher-Order Dynamic Mode Decomposition (HODMD) and Koopman spectral features for induction motor fault diagnosis and centrifugal pump flow characterization remains largely unexplored in the literature. Consequently, there is currently a significant gap between the extensive theoretical development of Koopman operator methods in power system applications and their practical adoption in industrial condition monitoring scenarios involving rotating electrical machinery.

1.1 Motivation

The present dissertation seeks to address this gap by extending Koopman-based methodologies toward two distinct but strongly related applications: induction motor fault diagnosis and coupled electromechanical-hydraulic flow characterization. Unlike previous studies that primarily employ Koopman theory for system identification or stability assessment, the proposed framework investigates the use of Koopman-derived spectral and modal features as physically interpretable representations for machine learning tasks. Furthermore, the work demonstrates that these representations not only improve classification performance relative to conventional raw-signal and Fourier-based approaches but also preserve the natural dynamical organization of the underlying physical system, providing an important bridge between nonlinear dynamical systems theory and practical industrial monitoring applications.

The motivation of this work is therefore to investigate whether Koopman operator theory can provide a more informative and physically meaningful representation of industrial signals. By combining concepts from nonlinear dynamical systems, spectral analysis, and machine learning, this dissertation seeks to develop a feature extraction framework capable of preserving the intrinsic system dynamics while simultaneously improving fault diagnosis and operating-condition estimation performance. Ultimately, the proposed approach aims to bridge the gap between advanced data-driven methodologies and practical engineering applications contributing toward more robust, interpretable, and generalizable industrial monitoring systems.

1.2 Research Objectives

The first objective of this thesis is to systematically evaluate Koopman-derived features in the context of motor fault detection, comparing their performance against raw

time-domain signals and conventional Fourier-based representations. By assessing multiple classifiers across these three data representations, this study aims to establish whether Koopman features offer superior discriminative capability and robustness in identifying electrical or mechanical anomalies.

The second objective is to extend the Koopman-based framework to coupled motor–pump system analysis. Building upon the previously published results using three-phase currents, this work investigates the influence of additional sensing modalities, including multi-axis vibration and voltage measurements for flow rate classification. Different combinations of input channels are evaluated to determine how multimodal fusion affects classification performance and dynamical separability. Through this analysis, the dissertation explores whether Koopman spectral descriptors maintain consistency and reliability across heterogeneous sensor configurations and which combination yields the best results.

By unifying fault detection and flow regime identification under a single operator-theoretic framework, this thesis seeks to demonstrate that Koopman-based feature extraction provides a generalizable and physically interpretable methodology for monitoring complex electromechanical systems. Rather than relying solely on fixed frequency bands or handcrafted spectral indicators, the proposed approach leverages data-driven dynamical decomposition to uncover coherent structures embedded in measured signals. The contributions of this dissertation can therefore be summarized as follows:

1. A comparative study between raw signals, Fourier-based representations, and Koopman-derived features for motor fault detection;
2. An extension of Koopman spectral analysis to multimodal coupled system monitoring;
3. An evaluation of supervised and unsupervised machine learning pipelines for flow regime classification under varying operations;
4. An assessment of the robustness and generalization capabilities of Koopman-based descriptors in non-stationary industrial environments.

Through these developments, this thesis aims to bridge classical signal processing and modern operator-theoretic learning approaches, advancing the state of the art in non-intrusive monitoring of motor-driven industrial systems.

1.3 Work Structure

This study is structured as follows: [Chapter 2](#) defines the theoretical concepts required to expose and further develop the proposed work. [Chapter 3](#) discusses the feature extraction procedure, classification pipelines for both fault detection and flow rate classification, and the datasets used to evaluate these pipelines. [Chapter 4](#) presents the main results of the project and their implications. Finally, [Chapter 5](#) concludes the study with the main insights, discusses the project limitations, and proposes future research based on the contributions.

2 Theoretical Review

Understanding the methodological foundations of this work requires navigating through four domains of knowledge: motor fault detection ([MCSA](#)), nonlinear dynamical systems, data-driven Koopman spectral analysis, and statistical learning. This section introduces the theoretical principles that enable the extraction of meaningful dynamic features from motor current signals and their use in motor fault detection and flow-regime classification.

2.1 Motor Current Signature Analysis and Fault Frequencies

Motor Current Signature Analysis ([MCSA](#)) is a non-invasive technique used to detect electrical and mechanical faults in motors by analysing the frequency spectrum of the stator current. The fundamental assumption of [MCSA](#) is that specific physical faults introduce predictable frequency components or sidebands around the supply frequency [\[7\]\[37\]](#).

The most common analytical tool in [MCSA](#) is the Fast Fourier Transform ([FFT](#)), which decomposes a time-domain signal into sinusoidal components under the assumption of stationarity within the observation window [\[4\]](#). When faults are present, they cause characteristic modulations in the current, producing frequency components - known as fault signatures - that deviate from the normal supply frequency [\[14\]\[16\]](#).

Each type of fault introduces specific frequency patterns such as:

- Broken rotor bars generate sidebands around the supply frequency f_s at:

$$f_{bar} = (1 \pm 2ks)f_s \quad (2.1)$$

Where k is any positive integer, and s is the rotor slip [\[3\]\[6\]\[7\]\[17\]](#).

- Air gap eccentricity, caused by misalignment or rotor deformation, leads to components at:

$$f_{ecc} = ((kR \pm n_d) \frac{1-s}{p} \pm \nu)f_s \quad (2.2)$$

Where R is the number of rotor slots, s is the rotor slip, p is the number of pole pairs, k is any positive integer, n_d is the eccentricity order, and ν is the order of stator harmonics [3][6][7][17].

- Bearing faults result in certain vibration which influences the eccentricity in air gap between the rotor and stator. Since the bearing vibration leads to ripple in output torque of the motor, it can be found in the current spectrum as:

$$\begin{aligned}
 f_o &= \frac{N_b}{2} f_r (1 - \frac{d}{D} \cos \beta) \\
 f_i &= \frac{N_b}{2} f_r (1 + \frac{d}{D} \cos \beta) \\
 f_c &= \frac{f_r}{2} (1 - \frac{d}{D} \cos \beta) \\
 f_b &= \frac{d}{2D} f_r [1 - (\frac{d}{D} \cos \beta)^2] \\
 f_{br} &= |f_s \pm n f_c|
 \end{aligned} \tag{2.3}$$

Where f_o is the frequency of outer race, f_i is the frequency of inner race, f_c is the frequency of cage failure, f_b is the frequency of ball defect, N_b is the number of balls in bearing, β is the contact angle of the bearing, f_r is the rotor speed, d_B is the ball diameter, D is the pitch diameter, and n is an integer [3][6][7][17].

- Stator winding faults create asymmetries in the magnetic field, typically leading to negative-sequence components and harmonics of f_s and f_r , computed as:

$$f_{st} = [k \pm \frac{n(1-s)}{p}] f_s \tag{2.4}$$

Where k is an odd integer, s is the rotor slip and p is the number of pole pairs [3][6][7][17].

MCSA is widely used in industry due to its simplicity, low cost, and ability to monitor machines under load without physical intrusion. However, accurate interpretation of the spectral content is essential for reliable fault diagnosis.

2.2 Nonlinear Systems Analysis

2.2.1 Takens' Embedding Theorem

Takens' Theorem provides the theoretical basis for reconstructing the state space of a smooth deterministic dynamical system from the time-delayed observations of a single measurement [38]. Specifically, if a system evolves on a compact attractor, a stable, bounded set that the system's trajectories eventually settle onto, with dimension d_A , then a sequence of delayed samples

$$x(t), x(t - \tau), x(t - 2\tau), \dots, x(t - (m - 1)\tau) \quad (2.5)$$

embeds the attractor into $m > 2d_A$ dimensions. This embedding is diffeomorphic to the true system state, meaning that the geometric properties of the latent dynamics are preserved [38][39].

In practice, this implies that time-delayed snapshots of motor current signals contain sufficient information to recover the underlying electromechanical and fluid-induced dynamics, even though only a scalar (or low-dimensional) observable is measured. This insight is foundational for methods such as Higher-Order Dynamic Mode Decomposition (HODMD), which operate on time-delay embedded matrices.

2.2.2 Time-Delay Embedding and the Hankel Matrix

A practical realization of Takens' embedding is the Hankel Matrix, constructed by sliding a window of length d across the signal:

$$H = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_m \\ x_2 & x_3 & x_4 & \cdots & x_{m+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_d & x_{d+1} & x_{d+2} & \cdots & x_{d+m-1} \end{bmatrix} \quad (2.6)$$

Each column of H represents a time-delayed state vector, and the matrix as a whole encodes the system's temporal geometry [40][41][30]. Key properties:

- It converts a temporal signal into a pseudo-spatial structure, enabling HODMD to treat time-delay coordinates as spatial modes [40][41][30].
- It captures oscillatory coherence across delays, enhancing the visibility of persistent frequencies [40][41][30].
- It increases rank, allowing the identification of multiple dynamics regimes (motor harmonics, load oscillations, turbulence-induced modulations).

For multichannel systems, channel can be stacked or embedded jointly, producing a richer representation of cross-channel correlations.

2.2.3 Koopman Operator Theory

The Koopman operator provides a linear perspective on nonlinear systems. Consider a nonlinear dynamical system following a dynamic function $F(x)$:

$$x_{t+1} = F(x_t) \quad (2.7)$$

with observable functions $g(x)$ [21][22][23]. The Koopman operator $\mathcal{K}$ acts linearly on these observables:

$$\mathcal{K}g(x) = g(F(x)) \quad (2.8)$$

Although the system itself may be nonlinear, the Koopman operator is infinite-dimensional and linear, allowing the system to be decomposed into modes and eigenvalues [21][22][23]:

$$g(x_t) = \sum_k \langle g(x_0), \phi_k(x) \rangle e^{\lambda_k t} \phi_k(x) \quad (2.9)$$

where:

- $\lambda_k = \sigma_k + j\omega_k$ encodes growth/decay and frequency [21][22][23]
- ϕ_k are eigenfunctions describing the spatial patterns [21][22][23]
- $\langle g(x_0), \phi_k(x) \rangle$ represents the initial value/amplitude of each Koopman mode [21][22][23]

For induction motor current signals, persistent electrical phenomena such as the supply fundamental, slot harmonics, rotor sidebands, and other fault-related frequency components, may therefore be interpreted as dominant Koopman frequencies [31][42]. Additional electromechanical interactions, including load variations and flow-induced torque oscillations, can introduce structures or modify the energy distribution among existing modes, leading to changes in the Koopman spectrum. Similar interpretations have been successfully employed in power-systems oscillation analysis, where eigenfunctions have been associated with coherent electromechanical phenomena and nonlinear oscillatory behaviours [31][42].

The modal amplitudes further enable the definition of an energy distribution over the Koopman spectrum,

$$p_k = \frac{|b_k|^2}{\sum_i |b_i|^2} \quad (2.10)$$

which can be interpreted as the relative contribution of each dynamical mechanism to the measured current signal. Under steady operation conditions, the energy tends to concentrate in a limited number of persistent modes associated with deterministic electrical excitations [31][42]. Conversely, increased electromechanical complexity, stochastic flow interactions, or fault-induced perturbations may redistribute the modal energy across a larger number of modes, producing a more dispersed Koopman spectrum [31][42].

From an information-theoretic perspective, this redistribution can be quantified through the spectral entropy

$$H = - \sum_k p_k \log(p_k) \quad (2.11)$$

which measures the complexity or disorder of the modal energy distribution. Low entropy indicates that the system dynamics are dominated by a few coherent oscillatory mechanisms, whereas high entropy suggests the presence of multiscale interactions, intermittent behaviours, or broadband dynamical phenomena [43][44]. Similar entropy-based interpretations have been extensively employed in motor current signature analysis and vibration diagnostics, where increases in spectral entropy are often associated with fault progression, turbulence, and nonstationary operating conditions [43][44].

Thus Koopman spectral analysis thus provides a unified framework for decomposing complex fluid-motor interactions.

2.2.4 Dynamic Mode Decomposition

Dynamic Mode Decomposition (DMD) is a data-driven method originally developed in fluid mechanics to extract coherent spatiotemporal structures from high-dimensional flow fields. Its relevance in pump-motor systems stems from its formal connection to the Koopman spectral decomposition of nonlinear dynamical systems [24][25][26].

Let $x_t \in \mathbb{R}^n$ be the system state or observable at time t . DMD assumes that the evolution of snapshots can be approximated by a linear operator:

$$x_{t+1} = Ax_t \quad (2.12)$$

Given snapshot matrices $X = [x_1, x_2, \dots, x_{m-1}]$, $X' = [x_2, x_3, \dots, x_m]$, DMD finds the best-fit linear operator A satisfying $X' \approx AX$. A least-squares estimate is [24][25][26]:

$$A \approx X'X^\dagger \quad (2.13)$$

Where $X^\dagger$ denotes the Moore-Penrose pseudo-inverse of the snapshot matrix X .

To prevent overfitting and improve numerical conditioning, [DMD](#) applies a rank truncated Singular Vector Decomposition (SVD) and computes the reduced operator [\[24\]\[25\]\[26\]](#):

$$\begin{aligned} X &= U\Sigma V^* \\ \tilde{A} &= U^* X' V \Sigma^{-1} \end{aligned} \quad (2.14)$$

U and V contain the left and right singular vectors of X . The columns of U form an orthonormal basis of the column space of X , the columns of V are the orthonormal basis of the row space of X and Σ is a diagonal matrix formed by the singular values of the X matrix. $(\cdot)^*$ denotes a conjugate transpose.

The Koopman eigenvalues are then approximated by the eigenvalues of $\tilde{A}$ [\[24\]\[25\]\[26\]](#):

$$\tilde{A} w_k = \lambda_k w_k \quad (2.15)$$

With associated modes:

$$\phi_k = X' V \Sigma^{-1} w_k \quad (2.16)$$

2.2.5 Higher-Order Dynamic Mode Decomposition

While [DMD](#) provides a powerful linear spectral analysis of dynamical systems, its performance deteriorates when the available measurements have low spatial dimensionality but large temporal length, or when the underlying process exhibits multi-scale temporal correlations, such as modulations, slow drifts, or weakly nonlinear coupling.

Higher-Order Dynamic Mode Decomposition ([HODMD](#)) extends the classical DMD framework by incorporating time-delay embedding prior to the computation of the Koopman spectrum. This modification fundamentally reshapes how the algorithm extracts spatiotemporal structure from data, enabling it to resolve dynamical features that are inaccessible to standard [DMD](#) [\[27\]\[28\]\[29\]](#).

The [HODMD](#) has two major modifications over classical DMD, first instead of using 2 snapshots X, X' with SVD to compute the Koopman eigenvalues and modes, the HODMD uses 2 time shifted Hankel matrices H_1, H_2 with Higher-Order Singular Value Decomposition ([HOSVD \(Higher-Order Singular Value Decomposition\)](#)) [\[27\]\[28\]\[29\]](#).

$$H = S \times_{n=1}^N U^{(n)} \quad (2.17)$$

with H being the Hankel matrix, S the core tensor and $U^{(n)}$ the orthonormal mode matrices, therefore $H_{(n)} = U^{(n)} \Sigma^{(n)} V^{(n)T}$.

This process has two crucial properties, first it transforms temporal structure into spatial structure, making it possible to separate components based in both instantaneous correlations and in temporal evolution, second it reveals coherent oscillatory patterns

that persist across delay, making it easier to distinguish dynamic true physical modes from random noise [27][28][29].

The delay embedding effectively increases the dimensionality of the observable space, enabling HODMD to distinguish between closely spaced frequencies, improve its robustness to noise, enabling to capture multi-scale dynamics, and emphasizes persistent temporal structure due to small variations are amplified in the embedded space, enabling their detection even when their instantaneous amplitude is small [27][28][29].

2.2.6 Understanding HODMD Through Classical Concepts

When an electrical engineer observes a current or vibration signal, the first instinct is usually to ask: "Which frequencies compose this signal?" This naturally leads to the Fourier Series. If a signal is periodic, it can be represented as a sum of harmonic oscillations:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k f_0 t} \quad (2.18)$$

This idea has been extraordinarily successful because many electrical systems indeed behave approximately as combinations of sinusoids. Stator currents, inverter harmonics, mechanical vibrations and resonance phenomena can all often be interpreted in this manner.

However, Fourier analysis makes a strong assumption: the oscillatory components are stationary and their amplitudes remain constant in time. Real systems rarely satisfy these assumptions. Faults in induction motors, load variations, transients, and nonlinear interactions produce components that may appear, disappear, grow, or decay over time.

To account for these effects, electrical engineering introduced the Laplace Transform. Instead of representing signals through pure oscillations $e^{j\omega t}$, the basis functions become exponentially modulated oscillations $e^{st} = e^{(\sigma + j\omega)t}$.

The parameter $s = \sigma + j\omega$ simultaneously describes: oscillation frequency (ω); exponential growth or decay (σ). This immediately connects with classical systems theory. Consider a linear dynamical system $\dot{x} = Ax$, its solution is $x(t) = e^{At}x_0$.

If the system matrix is diagonalizable $A = W\Lambda W^{-1}$ then the response becomes

$$x(t) = \sum_i c_i v_i e^{\lambda_i t} \quad (2.19)$$

Every engineer recognizes this expression: the system behaviour is completely determined by its poles and modes.

At this point, an important question naturally arises: "What if the governing equations are unknown?" Suppose only measurements are available $x_1, x_2, \dots, x_m$. Can one estimate the underlying dynamics directly from data?

Dynamic Mode Decomposition was developed precisely for this purpose. DMD assumes that consecutive measurements satisfy $X_{k+1} \approx AX_k$ where A is an unknown linear operator to be estimated from the data itself.

After estimating A , its eigenvalues play exactly the same role as the poles of a state-space system. Consequently, DMD may be viewed as a data-driven counterpart to modal Laplace analysis, since it identifies the dominant poles and modal contributions of a system directly from measured data, without requiring differential equations, physical models, transfer functions, or explicit state-space representations.

Nevertheless, many practical systems cannot be adequately described by a first-order relationship. Electrical and mechanical systems frequently possess memory. Their future state depends not only on the current measurement but also on previous states. A more realistic model often resembles $A_1x_k + A_2x_{k-1} + \dots + A_dx_{k-d+1}$. This is immediately recognizable as a higher-order difference equation or an autoregressive model.

The classical control solution to this problem is state augmentation. A second-order differential equation, for example, is converted into a first-order state-space model by introducing additional state variables.

Higher-Order Dynamic Mode Decomposition follows exactly the same philosophy. Instead of considering only the present snapshot, it constructs an augmented state vector containing several delayed measurements

$$z_k = \begin{bmatrix} x_k & x_{k+1} & \vdots & x_{k+d-1} \end{bmatrix} \quad (2.20)$$

The remarkable aspect is that this procedure is not merely an engineering trick; it is deeply connected with nonlinear dynamical systems theory through Takens' Embedding Theorem.

Takens' theorem states that, under relatively mild assumptions, the complete state-space dynamics of a deterministic system can be reconstructed from delayed measurements of a single observable. In other words, even if the true internal states of the motor are inaccessible, the sequence $[x(t), x(t + \tau), x(t + 2\tau), \dots, x(t + (d - 1)\tau)]$ contains sufficient information to reconstruct a space that is topologically equivalent to the original dynamical manifold.

This result is profound. It implies that the hidden dynamics of a nonlinear system may be recovered simply by stacking delayed measurements. The practical implementation of this idea naturally leads to the construction of a Hankel matrix.

$$H = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots \\ x_2 & x_3 & x_4 & \cdots \\ x_3 & x_4 & x_5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (2.21)$$

Each column of the Hankel matrix represents one delayed state vector. Consequently, the Hankel matrix acts as a data-driven reconstruction of the system's state space.

HODMD can therefore be viewed as a synthesis of several classical ideas:

- Takens' theorem provides the theoretical justification for delay embedding;
- Hankel matrices implement the delay embedding numerically;
- State augmentation transforms higher-order dynamics into first-order dynamics;
- DMD estimates the hidden linear evolution operator;
- Koopman theory justifies why even nonlinear systems may exhibit approximately linear evolution in an appropriately constructed space.

Once the delayed state space has been constructed, DMD is applied to the Hankel snapshots. The resulting decomposition assumes the form

$$\sum_{j=1}^r b_j \phi_j e^{\lambda_j t} \quad (2.22)$$

where ϕ_j are the dynamic modes, $\lambda_j = \sigma_j + j\omega_j$ are continuous-time eigenvalues, ω_j represent modal frequencies, and σ_j describe modal growth or decay.

This expression should appear extremely familiar. It is mathematically almost identical to modal decompositions encountered in Laplace analysis, pole-residual representations, Prony methods, state-space and modal analysis. The essential difference is

that HODMD estimates these quantities directly from measured data and can do so even when the underlying governing equations are unknown or nonlinear.

Consequently, HODMD may be interpreted as a data-driven modal analysis framework that combines ideas from Fourier analysis, Laplace transforms, system identification, delay embeddings, and nonlinear dynamical systems theory.

In a single sentence: HODMD reconstructs a hidden state space from delayed measurements, estimates its underlying evolution operator, and extracts the dominant oscillatory mods and poles of the system directly from data.

2.3 Classification Algorithms

While the Koopman Operator and its associated decomposition methods provides powerful tools for analysing temporal evolution and modal structure of dynamical systems, classification algorithms offer a systematic way to assign observed system states or trajectories into predefined categories, thereby enabling the identification of characteristic behaviours, anomalies, or operating regimes. Classification algorithms can generally be divided into two broad paradigms: supervised and unsupervised learning.

Supervised learning relies on labelled data, where each observation in the dataset is associated with a known class or outcome. The goal of supervised algorithms is to learn a mapping from input space to these known classes, allowing the model to predict the class of new, unseen observations. In the context of dynamical systems, supervised methods can, for example, classify regimes of operation, detect faults, or distinguish between periodic and chaotic behaviours.

In contrast, unsupervised learning does not rely on labelled data. Instead, these methods aim to discover intrinsic structures or patterns within the data by grouping similar observations together. When applied to dynamical systems, unsupervised classification algorithms can reveal emergent behaviours, identify coherent structures, or detect transients between different dynamical regimes without prior knowledge of the system's classes.

By combining modal decomposition techniques with classification algorithms, one can transform complex, high-dimensional system data into a structured framework where distinct behaviours can be systematically recognized and analysed. This approach facilitates both predictive monitoring and interpretation of complex dynamical phenomena, complementing the insights provided by the spectral methods discussed before.

2.3.1 Supervised Learning Methods

2.3.1.1 k-Nearest Neighbours

The k-Nearest Neighbours ([k-NN \(k-Nearest Neighbours\)](#)) algorithm is one of the foundational methods in statistical learning, operating under the principle that nearby points in features space tend to belong to the same class. Formally, given a test point x , the algorithm identifies the k points in the training set that minimize the Euclidean distance $\|x - x_i\|$, and assigns to x the majority class among this neighbourhood [45][46].

Despite its simplicity [k-NN](#) can be extremely powerful when the feature space is well-structured. Each point in the feature space retains a dynamical interpretation, meaning that samples with similar dynamical regimes naturally cluster together.

However, [k-NN](#) is sensitive to feature scaling, the curse of dimensionality, and local density variations. When features possess different magnitudes, states that should be similar may appear distant, thus proper data standardization is essential. Furthermore, [k-NN](#) is a lazy learner: it stores training data verbatim, making predictions compute intensive for large datasets [45][46].

2.3.1.2 Support Vector Machines

Support Vector Machines ([SVM \(Support Vector Machine\)](#)) are maximum-margin classifiers that construct an optimal separating hyperplane between two or more classes. The central idea is to maximize the geometric margin:

$$margin = \frac{2}{\|w\|} \quad (2.23)$$

where w is the normal vector to the separating hyperplane. A large margin is desirable because it yields better generalization capacity and ensures robustness against noise and small fluctuations in the data [47][48].

When the features of the dataset are not linearly separable, SVM introduces kernel functions that implicitly project inputs into high-dimensional feature spaces. The most important kernels in this project include:

- Sigmoidal (tanh) kernel can capture S-shaped, saturating nonlinearities and works well when the decision boundaries resemble those in shallow neural networks [48].
- Radial Basis Function ([RBF \(Radial Basis Function\)](#))/Gaussian kernel introduces highly flexible nonlinear boundaries and excels at modelling smooth, continuous decision surfaces [48].

2.3.1.3 Random Forests

Random Forests ([RF \(Random Forest\)](#)) are an ensemble learning model consisting of multiple decision trees trained on randomly selected subsets of data and features. Each tree partitions the feature space using axis-aligned thresholds to minimize impurity criteria such as the Gini index [49][50].

[RF](#) improves upon a single decision tree by reducing variance through averaging. Each tree is weakly correlated with the others, thanks to feature subsampling and bootstrap dataset resampling. When aggregated, the ensemble is dramatically more robust to noise and overfitting [49][50].

2.3.1.4 Logistic Regression

Logistic Regression models the log-probability of a data point belonging to a class as linear function of the features:

$$\log \frac{P(y = a|x)}{1 - P(y = a|x)} = w_a^T x + b_a \quad (2.24)$$

Where $P(y = a|x)$ being the probability of x belonging to the class a , w_a is the weight vector for class a , and b_a is the bias of the class a .

Although intrinsically linear, it remains highly informative and serves as a baseline for linear separability. Regularization is essential to avoid overfitting and to encourage model stability which can be done with the L^1 norm penalty, the L^2 norm penalty, or the elastic-net regularization, that linearly combines both norms for a more accurate regression [51][52].

2.3.1.5 Gaussian Naive Bayes

Gaussian Naive Bayes ([GNB \(Gaussian Naive Bayes\)](#)) assumes feature-wise conditional independence and models each feature distribution per class as a Gaussian. Despite the independence assumption being violated in most real-world situations it often performs well because it provides a probabilistic decision rule, it is robust to small training sets, and it handles noisy data gracefully [53][54].

GNB is particularly valuable as a baseline, comparing a simple generative model against discriminative or ensemble approaches.

2.3.1.6 Extreme Gradient Boosting

Extreme Gradient Boosting ([XGBoost \(Extreme Gradient Boosting\)](#)) is a gradient-boosted tree algorithm that builds decision trees in sequence, each new tree attempting to

correct the residual error of the ensemble built so far. The key idea is to optimize a differentiable loss function using boosting techniques, incorporating second-order information (gradient + Hessian) [55][56].

The main strengths of XGBoost include:

- Regularized tree construction, preventing overfitting via penalization of tree complexity.
- Ability to model subtle nonlinearities, making it highly competitive for tabular data.
- Intrinsic handling of interactions between multiple features, such as cross-effects between spectral energy distributions and mode coherence.

Given the nonlinear and multi-physical nature of the data, XGBoost frequently provides among the strongest predictive performances [55][56].

2.3.1.7 Multi-Layer Perceptron

Multi-Layer Perceptron ([MLP \(Multi-Layer Perceptron\)](#)) is a universal approximator capable of modelling arbitrary decision boundaries through nonlinear transformations. It consists of layers of artificial neurons, each applying linear transformations followed by nonlinear activation functions (e.g. ReLU, tanh, sigmoid, etc) [57][58].

Key theoretical aspects:

- Universal Approximation Theorem: a sufficiently wide MLP with at least one hidden layer can approximate any continuous function with arbitrary precision.
- Backpropagation: gradients propagate backward through the network to update parameters.
- Regularization: dropout, batch normalization, and weight decay help control overfitting.

2.3.2 Unsupervised Learning Methods

2.3.2.1 k-Means Clustering

k-Means is one of the most widely used clustering algorithms due to its simplicity and its grounding in classical optimization theory. The algorithm seeks a partition of the dataset into k clusters such that the total within-cluster variance is minimized [59][60].

Formally, it solves:

$$\min_{\{C_i\}} \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (2.25)$$

where μ_i denotes the centroid of cluster C_i . The algorithm alternates between two steps:

- Assignment step: each point is assigned to the closest centroid.
- Update step: each centroid is recomputed as the mean of its assigned points.

Since this is equivalent to a coordinate descent procedure, convergence is guaranteed but only to a local minimum. However, k-Means cannot model elongated or overlapping distributions, and it assumes equal cluster variance, which may not hold in complex operating conditions [59][60].

2.3.2.2 Gaussian Mixture Models

Gaussian Mixture Models ([GMM \(Gaussian Mixture Model\)](#)) generalize k-Means by modelling each cluster as a multivariate Gaussian, rather than as a spherical region around a centroid. A GMM defines the data distribution as:

$$p(x) = \sum_{i=1}^K \pi_i \mathcal{N}(x | \mu_i, \Sigma_i) \quad (2.26)$$

where each component has its own mean μ_i , covariance Σ_i , and mixing weight π_i [61][62]. The Expectation-Maximization (EM) algorithm iteratively estimates these parameters:

- E-step: compute posterior probabilities $p(z = i | x)$, providing soft cluster assignments.
- M-step: update means, covariances, and weights to maximize likelihood.

[GMM](#) is highly expressive because it can capture elliptical clusters, clusters with distinct variances, overlapping regions, and probabilistic membership, allowing nuanced interpretation of ambiguous samples [61][62].

2.3.2.3 Hierarchical Density-Based Spatial Clustering

Hierarchical Density-Based Spatial Clustering ([HDBSCAN \(Hierarchical Density-Based Spatial Clustering of Applications with Noise\)](#)) is a density-based clustering algorithm derived from Density-Based Spatial Clustering ([DBSCAN \(Density-Based Spatial Clustering\)](#)) but extended to handle variable-density regions and hierarchical cluster structures. Rather than specifying a global density threshold, HDBSCAN constructs a density-based spanning tree of the data and extract clusters as regions of persistent high-density connectivity [63][64].

Key theoretical features:

- Variable-density clustering: unlike k-Means or [GMM](#), [HDBSCAN](#) can detect clusters of very different densities, a property essential when the system produces sparse or uneven feature distributions.
- Hierarchical cluster condensation: HDBSCAN generates a dendrogram of density levels and selects clusters based on stability - how long a cluster persists across density thresholds.
- Noise identification: the algorithm explicitly classifies low-density points as "noise", which is particularly useful for detecting outliers or ambiguous operating states.

HDBSCAN provides a principled way to discover natural groupings and outliers without imposing any geometric assumptions [\[63\]](#)[\[64\]](#).

2.3.2.4 Agglomerative Hierarchical Clustering

Agglomerative Clustering builds a hierarchy of cluster merges starting from individual data points. At each step, the two most similar clusters are merged according to a linkage criterion:

- Single linkage: Measures the distance between closest points
- Complete linkage: Measures the distance between farthest points
- Average linkage: Computes the average distance between all pair of points
- Ward's linkage: Aims to minimize the increase in total variance within clusters

Agglomerative Clustering yields a dendrogram, a tree that visually illustrates the multi-scale cluster structure. It does not assume specific cluster shapes, but it can be sensitive to outliers and to the chosen linkage method [\[65\]](#)[\[66\]](#).

2.3.2.5 Spectral Clustering

Spectral clustering operates by interpreting the dataset as a graph. Each data point becomes a node, and edges encode similarity between pairs of points. From this graph, a graph Laplacian matrix is constructed:

$$L = D - W \tag{2.27}$$

where W is the similarity matrix and D is the degree matrix. The eigenvectors of the Laplacian encode the geometric structure of the data. The algorithm embeds the points into a low-dimensional space using the first few Laplacian eigenvectors and then applies k-Means in this spectral embedding [\[67\]](#)[\[68\]](#).

The power of spectral clustering lies in its ability to:

- Detect non-convex clusters
- Capture manifold structure
- Separate data linked by graph connectivity rather than geometric proximity

Spectral clustering is especially relevant because the Laplacian embedding can untangle nonlinear manifolds, revealing class structure that is not visible in the raw feature space [67][68].

2.3.2.6 Mean Shift Clustering

Mean Shift is a non-parametric clustering method that interprets data as samples from an underlying probability density function. The algorithm performs gradient-ascent on the estimated density surface: each point iteratively shifts toward that local maxima of the Kernel Density Estimate ([KDE \(Kernel Density Estimate\)](#)). Clusters emerge naturally as basins of attraction of density peaks.

Properties that make Mean Shift valuable:

- No need to predefine the number of clusters, unlike k-Means or [GMM](#)
- Detection of arbitrarily shaped clusters
- Ability to identify multiple operational modes in the data distribution

The main challenge is the choice of bandwidth, too small it yields many small clusters, too large it flattens important structure, making it harder to distinguish the clusters.

2.3.3 Model Selection and Performance Evaluation

2.3.3.1 Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique ([SMOTE \(Synthetic Minority Oversampling Technique\)](#)) is a widely adopted strategy to mitigate the issue of imbalanced datasets and model biases towards the majority classes. Instead of duplicating minority samples (which increases overfitting), SMOTE synthesizes new, plausible samples by interpolating between existing ones [69][70]. For a minority-class sample x , SMOTE selects one of its k nearest minority neighbours x_{nn} and generates a synthetic point:

$$x_{new} = x + \lambda(x_{nn} - x), \lambda \sim \mathcal{U}(0, 1) \quad (2.28)$$

This effectively expands the minority regions of feature space in a smooth, data-consistent manner. This reduces model bias and ensures that classification performance reflects genuine discriminative structures [69][70].

2.3.3.2 Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP (Uniform Manifold Approximation and Projection)) is a modern nonlinear dimensionality-reduction method grounded in Riemannian geometry and fuzzy simplicial set theory. Unlike linear reduction methods such as Principal Component Analysis (PCA (Principal Component Analysis)) or Linear Discriminant Analysis (LDA (Linear Discriminant Analysis)), UMAP models the dataset as a low-dimensional manifold embedded in a higher-dimensional space and seeks to preserve both local geometry (neighbourhood relationships) and global topology (connected components and cluster separation) [71][72].

UMAP constructs a weighted graph of local neighbourhoods in the high-dimensional space using:

- a smooth k-nearest-neighbour kernel
- distance-to-probability transformation transformations that encode fuzzy connectivity.

It then optimizes a low-dimensional representation whose fuzzy-simplicial structure best matches the high-dimensional graph [71][72]. UMAP projections thus provide an intuitive way to inspect whether physically meaning cluster separation emerges from the Koopman spectral descriptors.

For each sample x_i , the similarity between two points is defined as:

$$p_{ij} = \exp\left(-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}\right) \quad (2.29)$$

where $d(x_i, x_j)$ is the distance between samples, ρ_i is the distance to the nearest neighbour, and σ_i the local scaling factor controlling the neighbourhood size. Next a embedded space is constructed using $q_{ij} = (1 + a||y_i - y_j||^{2b})^{-1}$ with y_i is low-dimensional coordinates while a, b are parameters controlling the embedding shape.

So UMAP minimizes the cross-entropy between the two fuzzy simplicial sets

$$\mathbb{L} = \sum_{i,j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) + (1 - p_{ij}) \log\left(\frac{1 - p_{ij}}{1 - q_{ij}}\right) \quad (2.30)$$

2.3.3.3 Efficient Hyperparameter Optimization

Machine learning models often contain hyperparameters, such as regularization strengths, kernel widths, number of neighbours, and embedding parameters that strongly influence performance. Exhaustive search becomes impractical when dealing with high dimensional feature spaces and multiple classifiers, particularly in pipelines that include nonlinear transformations and data resampling.

To address these limitations, we employ an adaptive hyperparameter optimization strategy based on a sequential model-based approach using a Tree-structured Parzen Estimator ([TPE \(Tree-structured Parzen Estimator\)](#)) [73][74]. Instead of sampling configurations uniformly at random, the method iteratively refines the search by learning from previous evaluated configurations. Candidate hyperparameter sets are sampled using a probabilistic model that prioritizes regions of the search space associated with higher performance, enabling a more efficient exploration compared to purely random strategies.

The next hyperparameters are selected by maximizing

$$EI(x) \propto \frac{l(x)}{g(x)} \quad (2.31)$$

where $l(x) = p(x|y < y^*)$, $g(x) = p(x|y \geq y^*)$, and $y^* = \gamma - \text{quantile}(y)$. New configurations are then sampled preferentially from regions exhibiting a higher probability of producing improved validation performance, resulting in a significantly more efficient exploration of large hyperparameter spaces.

For each sampled configuration, performance is estimated using a stratified cross-validation scheme. Additionally, a multi-fidelity evaluation strategy is adopted where configurations are first assessed on reduced subsets of the data and progressively refined only if they demonstrate promising performance [75][76]. An early pruning mechanism is also applied during cross-validation, allowing poorly performing configurations to be discarded before completing all validations folds. And, to help the algorithm don't get stuck in local minima, a restart system is deployed, redoing the search from a new start point so the algorithm is able to explore the search grid efficiently and reach a true global minimum. The restarting mechanism works by detecting the search converged to a point, then it picks a new random starting point and redo the search, doing this loop for a maximum of n times. If the best CV score isn't improved in $threshold_1$ iterations, the search converges to the same point $threshold_2$ times or the mean CV score of the last k iterations converges ($std \leq \epsilon$), the algorithm interrupts the search to avoid wasting resources when the algorithm has a clear convergence and returns the best model found. This combined approach reduces computational cost by avoiding full evaluation of poor configurations, focuses the search on promising regions of the hyperparameter space, efficiently handles continuous and discrete parameter domains, does multiple searches to assure global minimum convergence, and seamlessly integrates with stratified K-fold cross-validation.

2.3.3.4 Stratified K-Fold Cross-Validation

Cross-validation ([CV \(Cross-Validation\)](#)) is essential for assessing a model's ability to generalize to unseen data. However, when classes are imbalanced naive K-fold methods may yield folds lacking sufficient representation of minority classes.

Stratified K-fold cross-validation ensures that each fold maintains approximately the same class distribution as the full dataset. This prevents biased model evaluation and reduces variance in performance estimates [77][78].

Mathematically, the dataset D is partitioned into K disjoint fold $D_1, \dots, D_K$ such that:

$$P(y = c|x \in D_i) \approx P(y = c|x \in D), \forall c, i \quad (2.32)$$

Each folds is used once as a validation set while the remaining $K - 1$ folds serve as training data [77][78].

2.3.3.5 Shapley Additive Explanations

Shapley Additive Explanations (SHAP (Shapley Additive Explanations)) values are a method used to interpret the predictions of machine learning models by breaking them down into contributions from each input feature by comparing it to a baseline, which is typically the average prediction of the model across the dataset. Each feature is assigned a value that represents how much it pushes the predictions higher or lower relative to the baseline. When you add up all these individual contributions, they reconstruct the model's final output, making the explanation both intuitive and consistent [79][80].

The central idea is that each feature's contribution is computed by considering all possible subsets of features and measuring how much adding that feature changes the model's prediction. Formally, the SHAP values for a features i is given by:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} \cdot [f(S \cup \{i\}) - f(S)] \quad (2.33)$$

Where N is the set of all features, S is a subset of all features not containing i , $f(S)$ represents the model's prediction when only the features in S are known, and $|\cdot|$ denotes the cardinality of a set. The weighting term ensures that each ordering of features inclusion is treated fairly [79][80].

2.3.3.6 Evaluation Metrics for Supervised Learning

Supervised metrics quantify how well the classifier predicts known labels. In this work, standard metrics are used that highlight complementary aspects of performance, especially under class imbalance.

The following metrics break down the classifier's behaviour in terms of type of mistakes. Precision measures how many predicted samples of a class truly belong to that class, recall (sensitivity) measures how many true samples of a class the model is able to

retrieve and F1-Score is the harmonic mean of precision and recall [81].

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1 &= 2 \frac{Precision \cdot Recall}{Precision + Recall} \end{aligned} \quad (2.34)$$

Balanced accuracy is preferred over plain accuracy when class distributions are skewed. Instead of counting raw correctness, it averages the recall obtained on each class [82].

$$Balanced_Accuracy = \frac{1}{C} \sum_{i=1}^C Recall_i \quad (2.35)$$

Receiver Operating Characteristic-Area Under Curve (ROC (Receiver Operating Characteristic)-AUC (Area Under the Curve)) evaluates how well the classifier separates classes based on its decision scores rather than hard labels. For binary classification, the ROC curve plots the True Positive Rates against the False Positive Rate, while AUC summarizes the entire ROC curve into a single number, representing the probability that the classifier assigns a higher score to a randomly chosen positive example than to a negative one [81].

2.3.3.7 Evaluation Metrics for Unsupervised Learning

Unsupervised metrics assess the structure and quality of clusters without relying on ground-truth labels. If the feature space forms compact and well-separated clusters, downstream supervised models become easier, more stable, and more interpretable.

The Silhouette Score measures how similar a sample is to its own cluster compared to other clusters. For each point:

- a is the mean distance to all other points in the same cluster (cohesion)
- b is the minimum mean distance to points in the nearest different cluster (separation)

Then:

$$s = \frac{b - a}{\max(a, b)} \quad (2.36)$$

- $s \approx 1$: point is well matched to its cluster
- $s \approx 0$: point lies between clusters
- $s < 0$: point may be misassigned

The overall Silhouette Score is the average of s over all points. It simultaneously captures cluster compactness and inter-cluster separation, making it a good general-purpose quality index [83].

Calinski-Harabasz Index ([CH \(Calinski-Harabasz Index\)](#)) evaluates the ratio between between-cluster dispersion with within-cluster dispersion. Formally:

$$CH = \frac{\text{between-cluster dispersion}}{\text{within-cluster dispersion}} \cdot \frac{N - C}{C - 1} \quad (2.37)$$

Where C is the number of clusters and N is the number of samples. High [CH](#) values indicate that cluster are well separated, each cluster is internally coherent and the feature space has strong natural grouping structure [84].

Davies-Bouldin Index ([DBI \(Davies-Bouldin Index\)](#)) measures the average similarity between each cluster and its most similar neighbour, using a ratio of cluster scatter over distance between cluster centroids:

$$DB = \frac{1}{C} \sum_{i=1}^C \max_{j \neq i} \left(\frac{S_i + S_j}{M_{ij}} \right) \quad (2.38)$$

where S_i is the average distance of points in cluster i to its centroid and M_{ij} is the distance between centroids of clusters i and j . Lower DB values are better because it means that cluster are compact, far apart and has no dangerously similar neighbour [85].

Adjusted Rand Index ([ARI \(Adjusted Rand Index\)](#)), differently from previous metrics, does use data labels when available. It measures the agreement between the obtained clustering and the true classes, correcting for random label assignments [86].

$$ARI = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} \quad (2.39)$$

High [ARI](#) values indicates good cluster alignment with the true classes while low values tells random clustering [86].

3 Methodology: Proposed Pipeline

This chapter presents the complete methodological framework developed in this dissertation. The structure follows a progressive validation strategy. First, the Koopman-based feature extraction pipeline is described in detail, including preprocessing, time-delay embedding, HODMD decomposition, and feature construction. Second, these Koopman-derived features are evaluated in the context of motor fault detection and compared against raw time-domain signals and classical Fourier-based representations along with a signal spectrum generated by the FFT algorithm compared against the spectrum generated via the Koopman Operator. Finally, the methodology is extended to the flow rate classification problem in coupled motor–pump systems, incorporating both supervised and unsupervised learning pipelines.

The objective of this staged structure is to demonstrate that Koopman spectral features are not only effective in classical motor fault detection but also generalize to more complex coupled dynamical inference problems.

3.1 Nonstationarity Handling Process

In a previous work [4], the author proposes a lightweight and interpretable framework for detecting nonstationary behaviour in induction motor current signals without relying on complex signal processing techniques such as wavelets, empirical mode decomposition, or deep learning. The central idea is transient phenomena, such as load variations, inverter disturbances, or incipient faults, modify the statistical distribution of the motor current over time, and these changes can be detected directly through statistical analysis.

To achieve this, the current signal is divided into non-overlapping windows, and simple statistical descriptors, namely mean, variance, skewness, and kurtosis, are extracted from each segment. Rather than analysing spectral components, the method compares the distributions of these feature vectors using nonparametric two-sample hypothesis tests, specifically the Anderson-Darling, Kolmogorov-Smirnov, and Cram  r-von Mises tests. The evidence from these tests is then combined through p-value fusion strategies, including Fisher’s method and the Harmonic Mean P-value, allowing for a robust decision regarding the presence of nonstationary events. An adaptive reference buffer, whose size is controlled through the Wasserstein distance between consecutive windows, enables the algorithm to adjust to different levels of variability.

Initially, the method was created to filter out the nonstationary signals that can

introduce spectral leakage, frequency smearing, and biased feature representations, and potentially leading to misleading machine learning models. However, such preprocessing turn out unnecessary when employing the proposed Koopman-based framework. By leveraging Takens' embedding theorem and [HODMD](#), the proposed methodology reconstructs the underlying dynamical system in an augmented state space, enabling the extraction of coherent spatiotemporal modes even in the presence of nonstationary behaviour. Unlike traditional spectral methods, the Koopman operator formulation does not rely on strict stationarity assumptions and is capable of representing transient dynamics as additional modes rather than as distortions of the spectral content. Consequently, nonstationary operating conditions don't significantly corrupt the extracted Koopman spectra, allowing the framework to preserve diagnostically relevant information without requiring prior filtering of transient signals.

3.2 Koopman Feature Extraction Pipeline

Before Koopman analysis, each signal undergoes minimal preprocessing to guarantee numerical stability while avoiding transformations that could bias the classification. All selected channels, consisting of the chosen input signals are vertically concatenated into a matrix:

$$X = \begin{bmatrix} Channel_1(t) \\ Channel_2(t) \\ \vdots \\ Channel_N(t) \end{bmatrix} \in \mathbb{R}^{N \times T} \quad (3.1)$$

where N is the number of input channels and T is the number of time samples. Each channel is then RMS-normalized to remove scale differences caused by voltage/RPM fluctuations:

$$X_i(t) \leftarrow \frac{X_i(t)}{RMS(X_i)} \quad (3.2)$$

This preserves the physical structure (shape, dynamics, modulations) while making comparisons across recordings meaningful.

Then a multichannel Hankel matrix with $d = 25$ delays is constructed using the normalized input signals and it is decomposed into Koopman eigenvalues, amplitudes and modes via the HODMD algorithm with a 95% energy threshold rank r truncation. From which the dominant frequencies, spectral energy ratios and entropy, and growth/decay rates are extracted from the eigenfunctions that have high energy concentration and persistence, calculated from high to low energy modes that are present for at least 75% of the measured signal until the accumulated mode energy reaches at least 95% of the energy of the signal, as the Koopman features due to their encoding of the fundamental frequency, load-related oscillations, spectral distribution, consistency of oscillatory patterns across

channels, and identification of stationary or transient operation. These features were selected due to their ability to describe the measured system, its main dynamic modes, and their capability to tell apart multiple operating regime conditions. The physical relevance of these features is further supported by their statistical behaviour across operating regimes, for instance, the spectral entropy and number of significant eigenfunctions systematically increase with flow rate and fluid heterogeneity, consistent with the increased turbulence and load-induced current modulations.

What these Koopman features physically represent and how can be calculated is as follows:

- Frequencies represent the dominant temporal scales through which information propagates in the system dynamics: $f_k = \frac{\arg(\lambda_k) \cdot f_s}{2\pi}$
- Growth/decay rates quantify the persistence and memory of each dynamical component, indicating how long information remains relevant: $\sigma_k = \mathcal{R}(\ln(\lambda_k)) \cdot f_s$
- Spectral energy ratios describe how information is distributed among eigenfunctions, enabling the identification of dynamical concentration, dispersion and multiscale interactions: $E_k = \frac{\|\phi_k\|_2^2}{\sum_i \|\phi_i\|_2^2}$
- Spectral entropy corresponds to the Shannon entropy of the normalized Koopman energy distribution and measures the uncertainty, complexity and heterogeneity of the underlying dynamics: $p_k = -\sum_k E_k \cdot \ln(E_k)$

Where f_s is the sample frequency and $\mathcal{R}$ picks the real part of a complex number.

The Koopman feature set is then stratified split into a training dataset with 80% of the original dataset and a test data with the remaining 20%. Both the training and test dataset with their labels are fed as input to both supervised and unsupervised classification pipelines to evaluate how well the selected features can capture the dynamics of the measured system, how well they encode the flow rate information, how well they can distinguish multiple motor operation conditions and detect faults, and how they perform multiple against different classification approaches.

3.3 Fourier vs Koopman Spectrum & Fault Detection

To validate the discriminative capability of Koopman-derived features, a motor fault detection study is conducted. The objective is to compare three distinct signal re-

presentations: the raw time-domain stator current, its Fourier spectrum, and the extracted Koopman feature vector.

The raw representation consists of directly using sampled signal segments as high-dimensional input vectors. While this approach preserves all waveform information, it introduces substantial dimensionality and may contain redundant or noisy components.

The Fourier representation is obtained by computing the Fast Fourier Transform of each signal segment. The magnitude spectrum is used as the feature representation. This approach corresponds to classical MCSA methodology and captures frequency-domain energy distribution but does not explicitly encode temporal coherence or modal growth characteristics.

The Koopman representation follows the pipeline described previously, producing a lower-dimensional vector of physically interpretable dynamical features.

3.3.1 Dataset Description and Experimental Conditions

The dataset consists of 106 multichannel recordings acquired from a squirrel cage induction motor testbench operating under diverse fault conditions, load levels, and rotational speeds. The test machine is a Marathon Electric induction motor rated at 0.5 HP (0.373 kW), 220V, 2.1A, and 60Hz, with a rated efficiency of 66%, rated power factor of 0.697, rated speed of 3450 RPM, and rated torque of 1.03 N·m. The motor has two poles, 34 rotor bars, 36 stator slots, and was operated either directly from the utility supply or through a Variable-Frequency Drive (VFD), as illustrated in [Figure 1](#).

- 3 Motor stator currents (I_a, I_b, I_c)
- 3 Motor voltages (V_a, V_b, V_c)
- 3 Motor vibrational axes (horizontal, vertical and axial)

The signals span a wide range of operational conditions and supply frequency, such as 60 Hz directly from the electrical grid, 30 Hz, 50 Hz, and 60 Hz connected on the VFD, and motor conditions such as "Baseline", "Short-Circuit", "Broken Bars", "Unbalance (1 Screw)" and "Unbalance (2 Screws)" with distributions demonstrated in [Figure 2](#). To create the mechanical unbalance a uniform disc with screw slots was coupled to the motor to be used as load, and 1 or 2 screws were fixed to one side of the disc to generate an unbalanced load. While the original dataset had labels with all this information, the labels used in this project only mentions the fault the motor has so the frequency label don't interfere in the classification algorithms performance, 2 copies of this dataset were trained and tested on the first has labels "Baseline", "Broken Bars", "Short-Circuit" and "Unbalance" and the

second has the same labels except for the "Unbalance" which was split into "1 Screw" and "2 Screws".

In this analysis, to both avoid the extra chance of overfitting and adding redundant information, the raw and Fourier representations is focused primarily on one of the current phases (I_a) in steady state operation, while the Koopman representation uses all three phases, so we can have the most fair comparison between the 3 representations, so more "advanced" techniques such as Park's Vector and its Extended approach, Envelope Analysis, Negative Sequence Component Analysis, and Startup Transient Analysis won't be discussed.

While the focus here is only on current signals since they are strongly tied to electromechanical loading and motor operation conditions, the feature extraction pipeline is explicitly designed to support any combination of channels to better support the detection of motor faults. While larger datasets are always desirable, the collected data provides sufficient coverage of the fault conditions of the system, with each class containing more than enough samples for its representation. This ensures statistical representativeness and supports reliable training, validation, and testing of the proposed learning pipelines.

To better analyse the proposed Koopman feature pipeline a comparison between the raw signal, Fourier and Koopman Spectrum will also be made for all analysed motor conditions to know if the Koopman Operator can capture the expected phenomena already established in the Fourier based MCSA.

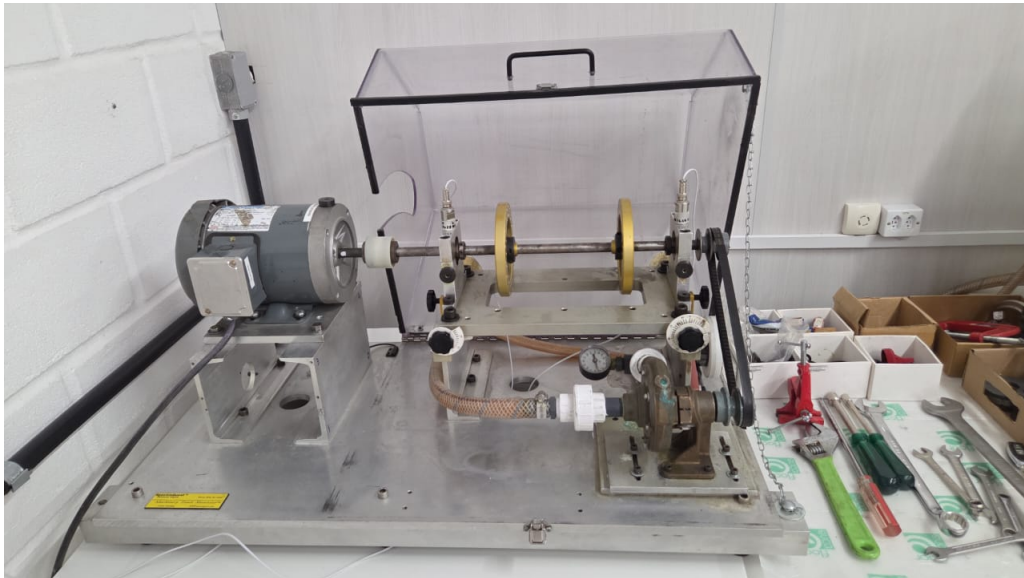


Figure 1 – Motor System Testbench

3.3.2 Supervised Classification Pipeline

Each supervised classification algorithm (k-NN, SVM, RF, Logistic Regression, GNB, XGBoost, and MLP) is trained using an identical internal pipeline that includes

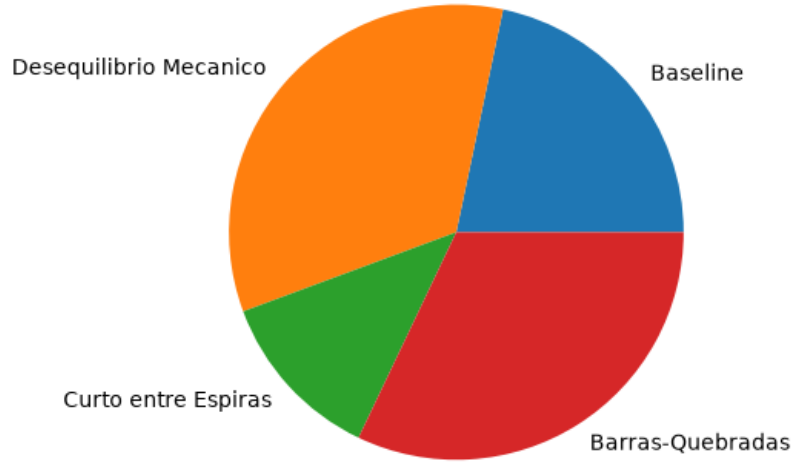


Figure 2 – Motor Fault Dataset Distribution

class balancing via SMOTE, dimensionality reduction via UMAP, and the classification algorithm itself. Although the Koopman features create a globally nonlinear manifold, that manifold is locally structured and, with the UMAP approximation enforcing a local metric linearity, the linear interpolated data created by the SMOTE algorithm can be correctly used to balance class priors, stabilize decision boundaries and reduce bias towards majority regimes.

The optimal hyperparameters for all elements of the pipeline are obtained using TPESearchCV, with a maximum of 1000 sampled configurations per restart evaluated under Stratified K-Fold cross-validation with 5 folds using balanced accuracy as its scoring metric. The optimization search jointly optimizes all SMOTE, UMAP, and classifier parameters, ensuring that interactions between these components are captured holistically. Each iteration of the TPESearchCV picks a set of hyperparameters for the classification algorithm, UMAP, and SMOTE. Then each CV fold fits all 3 algorithms to train and validate the results while avoiding data leakage.

Once the optimal pipeline is found, the model is retrained on the resampled and UMAP transformed training data and evaluated on the transformed test set. Then each global metric is evaluated together with a per-class evaluation, a confusion matrix, a SHAP summary plot showing how much each project features impacts the learning capability of each algorithm, and learning & validation curves showing the effect of sample size and key hyperparameters.

For a better visualization of the proposed induction motor fault classification pipeline, a flowchart is presented in Figure 3.

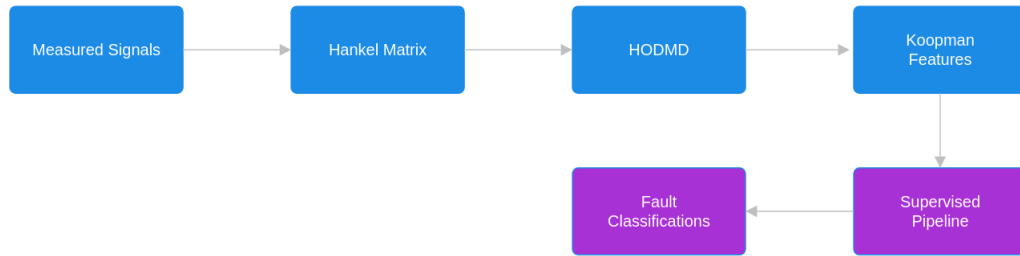


Figure 3 – Motor Fault Classification Flowchart

The hyperparameters evaluated in each CV fold of this pipeline are presented in Table 1.

Table 1 – Hyperparameter Search Space Evaluated in the Motor Supervised Pipeline

Model	Hyperparameter Search Space
kNN	$n_{neighbours} \in [2, 10]$; weights $\in \{\text{uniform, distance}\}$; Minkowski $p \in [1, 6]$
SVM	$C \in [10^{-2}, 10^2]$; kernel $\in \{\text{RBF, sigmoid}\}$; $\gamma \in \{\frac{1}{d \text{Var}X}, \frac{1}{d}, 0.01, 0.1\}$; shrinking $\in \{\text{True, False}\}$
Random Forest	$n_{estimators} \in \{100, 200, 300, 400, 500\}$; max depth $\in \{5, 10, 15, 20, 25, 30, \infty\}$
Logistic Regression	L^1 ratio $\in [10^{-2}, \dots, 1]$; tol $\in \{10^{-8}, \dots, 10^{-2}\}$; fit intercept $\in \{\text{True, False}\}$; solver = saga; intercept scaling $\in [0.5, \dots, 5.0]$; warm start $\in \{\text{True, False}\}$; class weight $\in \{\text{balanced, uniform}\}$
Gaussian NB	$\epsilon \in \{10^{-12}, \dots, 10^{-6}\}$
XGBoost	$n_{estimators} \in \{100, \dots, 500\}$; max depth $\in \{3, 5, 7, 9\}$; $\eta \in [0.05, 0.20]$; learning rate $\in \{0.05, 0.1, 0.15, 0.2\}$
MLP	hidden layer sizes $\in [2, \dots, 20]$; activation function $\in \{\text{logistic, tanh, ReLU}\}$; L^2 regularization $\in [10^{-6}, \dots, 10^{-3}]$; learning rate $\in \{\text{constant, invscaling, adaptive}\}$; learning rate init $\in \{10^{-6}, \dots, 10^{-3}\}$; batch size $\in \{32, 64, 128, 256, \min(200, \text{samples})\}$; shuffle $\in \{\text{True, False}\}$; tol $\in \{10^{-6}, \dots, 10^{-3}\}$; warm start $\in \{\text{True, False}\}$; iterations with no change $\in [5, 20]$; early stopping = True; solver = adam
UMAP	$n_{neighbours} \in \{2, 5, 10, \dots, 50\}$; min distance $\in [0.1, 0.9]$; metric $\in \{\text{euclidean, manhattan}\}$; $d \in [2, 5]$; $n_{components} \in [2, 10]$
SMOTE	$k_{neighbours} \in [2, 10]$

3.4 Flow Rate Classification in Centrifugal Pump Systems

In this context, the system exhibits strong electromechanical-hydraulic coupling. Variations in flow rate modify hydraulic torque demand, which alters rotor slip and modulates stator currents. Unlike discrete faults, flow rate represents a continuous physical quantity; however, due to the overlapping nature of electrical signatures across operating ranges, the problem is formulated as classification of discrete flow regimes.

3.4.1 Dataset Description and Experimental Conditions

The dataset consists of 1304 multichannel recordings acquired from a industrial centrifugal pump operating under diverse hydraulic and electromechanical conditions. The signals were collected from a test bench with a Submersible Centrifugal Pump model P100LS, series 538, a 50 HP induction motor and a CFW-09 (VFD) sampled at 11.718 kHz for 30 seconds from the test bench presented in [Figure 5](#), with its complete system represented in the schematic in [Figure 6](#). Each recording includes:

- 3 Motor stator currents (I_a, I_b, I_c)
- 3 Motor voltages (V_a, V_b, V_c)
- 3 Motor vibrational axes (horizontal, vertical and axial)

The signals span a wide range of operational conditions, such as flow rates varying between 0-100 m^3/h under different gas injection levels (0, 2, 5, and 8 kg/h), and motor rotational speeds of 1800, 2400, 3000, and 3500 RPM. While the original dataset was labelled with discrete values (e.g. 67.3 m^3/h), there would be a lot of different classes with few signals each so, as a preprocessing step, the dataset was relabelled into 11 flow rate ranges (i.e. 0, 1-10, 11-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90, and 91-100 m^3/h), making more signals per class with fewer classes at the cost of unbalancing the dataset as shown in [Figure 4](#). This diversity introduces strong nonlinearities and class overlap, which significantly increases the difficulty of the classification problem.

The injection of gas into the centrifugal pumping system significantly alters the hydraulic and electromechanical behaviour of the coupled motor-pump assembly. Under single-phase liquid operation, the flow through the pump is relatively stable, and the hydraulic load impose on the motor remains approximately steady for a given operating condition. However, as the gas injection increases, the flow becomes progressively more heterogeneous due to the formation of gas bubble, intermittent flow structures, and changes in the effective density and compressibility of the fluid mixture. These phenomena modify the pressure distribution inside the pump, affecting the impeller loading and introducing additional fluctuations in the torque required to maintain rotation.

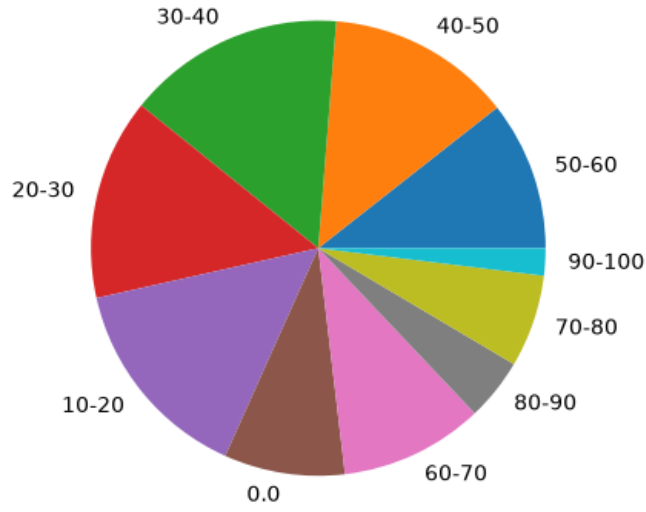


Figure 4 – Flow Rate Dataset Distribution

From a dynamical perspective, increasing gas injection rates tend to increase the nonstationary and nonlinear characteristics of the system. The resulting hydraulic instabilities may induce oscillatory behaviour associated with bubble formation, coalescence, and collapse, as well as flow recirculation and partial blockage phenomena. Consequently, the mechanical loading applied to the induction motor becomes increasingly time-varying, producing corresponding modulations in stator currents, vibrations, and other measurable quantities. Since the motor and pump form a strongly coupled electromechanical system, these hydraulic disturbances propagate through multiple physical domains, becoming indirectly observable in the electrical measurements and directly observable in the mechanical response.

As a result, different gas injection rates produce distinct dynamical regimes characterized by varying levels of temporal complexity, spectral content, and modal interactions. Such variations are expected to modify the Koopman spectral representation of the signals, influencing the distribution of modal energies, the emergence of new oscillatory components, and the relative importance of coherent dynamical structure. Therefore, the analysis of gas injection effects provides an important physical interpretation of the extracted Koopman features and helps explain their discriminative capability for operating-condition identification.



Figure 5 – Pump System Testbench - Source: Teste de desempenho de BCS e monitoramento de parâmetros elétricos - CEPETRO (2025)

3.4.2 Supervised Classification Pipeline

Each supervised classification algorithm (k-NN, SVM, RF, Logistic Regression, GNB, XGBoost, MLP) is trained using an identical internal pipeline that includes class balancing via SMOTE, dimensionality reduction via UMAP, and the classification algorithm itself. Although the Koopman features create a globally nonlinear manifold, that manifold is locally structured and, with the UMAP approximation enforcing a local metric linearity, the linear interpolated data created by the SMOTE algorithm can be correctly used to balance class priors, stabilize decision boundaries and reduce bias towards majority regimes.

The optimal hyperparameters for all elements of the pipeline are obtained using TPESearchCV, with a maximum of 1000 sampled configurations per restart evaluated under Stratified K-Fold cross-validation with 5 folds using balanced accuracy as its scoring

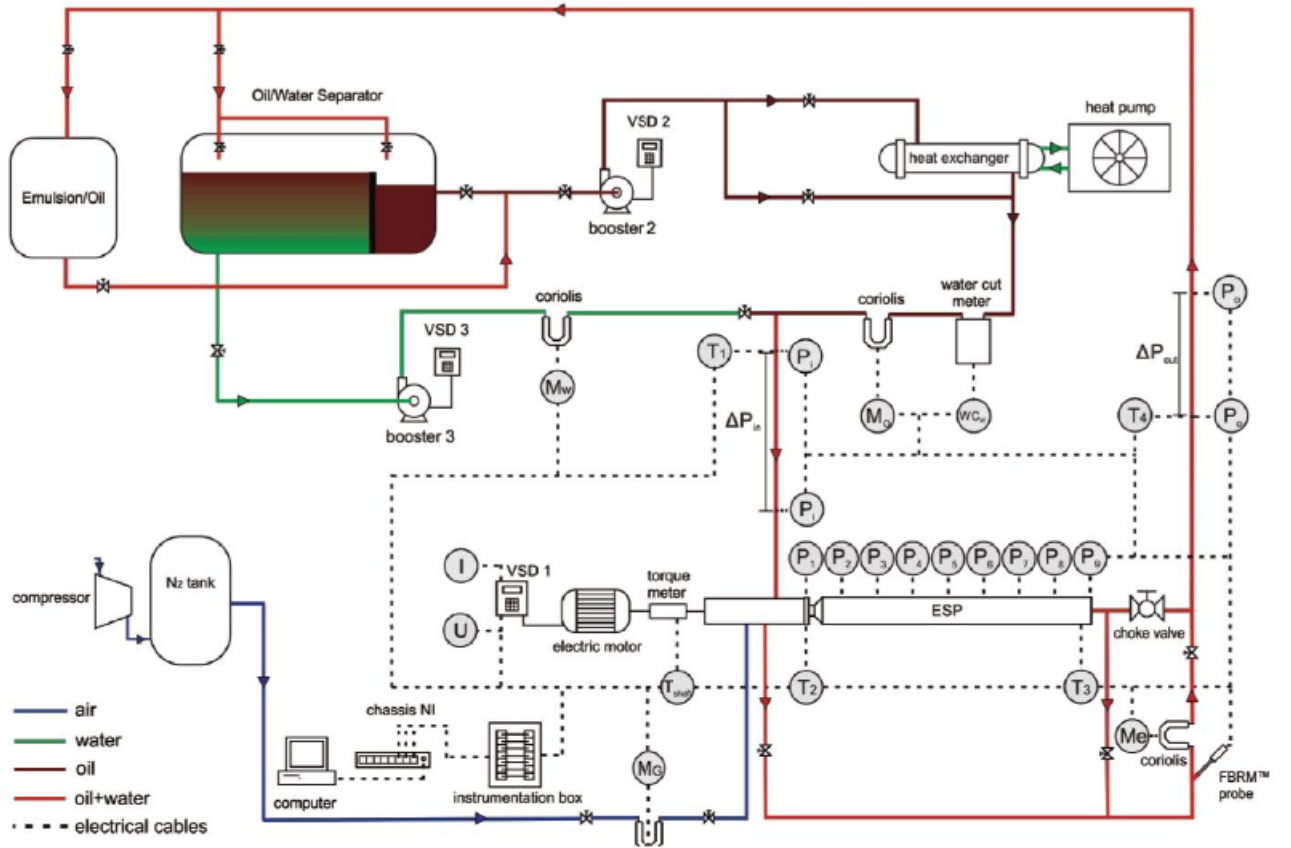


Figure 6 – Pump System Layout - Source: Teste de desempenho de BCS e monitoramento de parâmetros elétricos - CEPETRO (2025)

metric. The optimization search jointly optimizes all SMOTE, UMAP, and classifier parameters, ensuring that interactions between these components are captured holistically. Each iteration of the TPESearchCV picks a set of hyperparameters for the classification algorithm, UMAP, and SMOTE. Then each CV fold fits all 3 algorithms to train and validate the results while avoiding data leakage.

Once the optimal pipeline is found, the model is retrained on the resampled and UMAP transformed training data and evaluated on the transformed test set. Then each global metric is evaluated together with a per-class evaluation, a confusion matrix, a SHAP summary plot showing how much each project features impacts the learning capability of each algorithm, and learning & validation curves showing the effect of sample size and key hyperparameters.

The hyperparameters evaluated in each CV fold of this pipeline are presented in [Table 2](#).

Table 2 – Hyperparameter Search Space Evaluated in the Flow Rate Supervised Pipeline

Model	Hyperparameter Search Space
kNN	$n_{neighbours} \in [2, 10]$; weights $\in \{\text{uniform, distance}\}$; Minkowski $p \in [1, 6]$
SVM	$C \in [10^{-2}, 10^2]$; kernel $\in \{\text{RBF, sigmoid}\}$; $\gamma \in \{\frac{1}{d \text{Var}X}, \frac{1}{d}, 0.01, 0.1\}$; shrinking $\in \{\text{True, False}\}$
Random Forest	$n_{estimators} \in \{100, 200, 300, 400, 500\}$; max depth $\in \{5, 10, 15, 20, 25, 30, \infty\}$
Logistic Regression	L^1 ratio $\in [10^{-2}, \dots, 1]$; tol $\in \{10^{-8}, \dots, 10^{-2}\}$; fit intercept $\in \{\text{True, False}\}$; solver = saga; intercept scaling $\in [0.5, \dots, 5.0]$; warm start $\in \{\text{True, False}\}$; class weight $\in \{\text{balanced, uniform}\}$
Gaussian NB	$\epsilon \in \{10^{-12}, \dots, 10^{-6}\}$
XGBoost	$n_{estimators} \in \{100, \dots, 500\}$; max depth $\in \{3, 5, 7, 9\}$; $\eta \in [0.05, 0.20]$; learning rate $\in \{0.05, 0.1, 0.15, 0.2\}$
MLP	hidden layer sizes $\in [2, \dots, 20]$; activation function $\in \{\text{logistic, tanh, ReLU}\}$; L^2 regularization $\in [10^{-6}, \dots, 10^{-3}]$; learning rate $\in \{\text{constant, invscaling, adaptive}\}$; learning rate init $\in \{10^{-6}, \dots, 10^{-3}\}$; batch size $\in \{32, 64, 128, 256, \min(200, \text{samples})\}$; shuffle $\in \{\text{True, False}\}$; tol $\in \{10^{-6}, \dots, 10^{-3}\}$; warm start $\in \{\text{True, False}\}$; iterations with no change $\in [5, 20]$; early stopping = True; solver = adam
UMAP	$n_{neighbours} \in \{2, 5, 10, \dots, 50\}$; min distance $\in [0.1, 0.9]$; metric $\in \{\text{euclidean, manhattan}\}$; $d \in [2, 5]$; $n_{components} \in [2, 10]$
SMOTE	$k_{neighbours} \in [2, 10]$

3.4.3 Unsupervised Classification Pipeline

The unsupervised learning pipeline mirrors the structure of the supervised pipeline but removes all label-dependent components, replacing the classification with clustering. The goal is to assess how naturally the Koopman feature space organizes itself into coherent clusters that correspond to different flow-rate regimes, without explicit supervision.

The pipeline operates on the same training and test partitions but does not use class labels during model fitting, except when computing the ARI metric. For each unsupervised algorithm, the pipeline consists of the UMAP projection and the clustering model itself, unlike supervised models, no SMOTE is applied, since clustering performance does not depend on class balance.

A TPESearchCV wrapper is used to explore combinations of UMAP and clustering hyperparameters with a maximum of 1000 sampled configurations per restart evaluated under a Stratified K-fold cross-validation with 5 folds using Silhouette score as its scoring metric, due to no labels being available during fitting process. Each iteration of the TPESearchCV picks a set of hyperparameters for the classification algorithm, and UMAP. Then each CV fold fits all algorithms to train and validate the results while avoiding data leakage.

After model selection, the best UMAP-clustering pipeline is retrained and applied to the test set. Then all unsupervised metrics are calculated, the SHAP summary plot and confusion matrix (after label alignment) are plotted to further evaluation. This pipeline quantifies how strongly the Koopman feature embedding captures the inherent structure of different operational regimes, independent of classifier assumptions.

For a better visualization of the proposed flow rate classification pipeline, a flowchart is presented in [Figure 7](#).

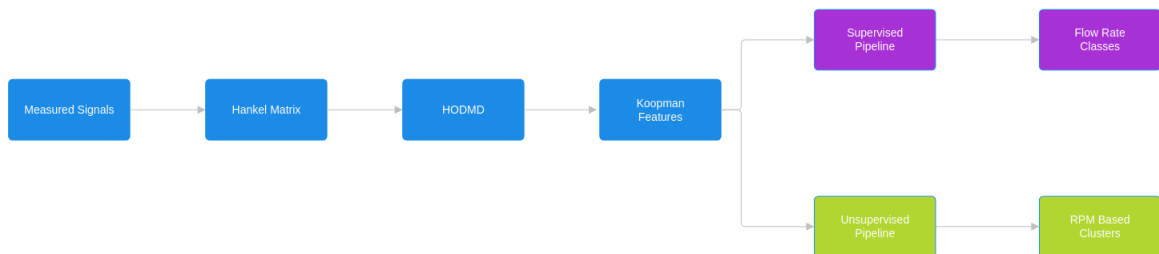


Figure 7 – Flow Rate Classification Flowchart

The hyperparameters evaluated in each CV fold of this pipeline are presented in Table 3.

Table 3 – Hyperparameter Search Space Evaluated in the Flow Rate Unsupervised Pipeline

Model	Hyperparameter Search Space
GMM	$n_{components} \in [2, 10]$; covariance type $\in \{\text{full, tied, diag, spherical}\}$; EM iters $\in \{100 \dots, 500\}$; tol $\in [10^{-2}, \dots, 10^{-6}]$; init $\in \{\text{kmeans, kmeans++}, \text{random, random from data}\}$
KMeans	$n_{clusters} \in [2, 10]$; $n_{init} \in [10, \dots, 50]$; max iter $\in \{200, \dots, 1000\}$; algorithm $\in \{\text{lloyd, elkan}\}$
HDBSCAN	min samples $\in \{100, 200, 300, 400, 500\}$; min cluster size $\in \{5, 10, 20, 50\}$; metric $\in \{\text{euclidean, manhattan, cosine}\}$; cluster selection method $\in \{\text{eom, leaf}\}$; alpha $\in \{0.5, 1.0, 1.5\}$; allow single cluster $\in \{\text{True, False}\}$
Agglomerative Clustering	$n_{clusters} \in [2, 10]$; metric $\in \{\text{euclidean, manhattan, cosine}\}$; compute full tree $\in \{\text{True, False, auto}\}$; linkage $\in \{\text{ward, average, complete, single}\}$
Spectral Clustering	$n_{clusters} \in [2, 10]$; eigen solver $\in \{\text{arpack, lobpcg}\}$; affinity $\in \{\text{nearest neighbours, rbf, laplacian, sigmoid, } \chi^2\}$; assign labels $\in \{\text{kmeans, discretize, cluster QR}\}$; $n_{neighbours} \in [5, 20]$
Mean Shift	band width $\in [0.5, 2.5]$; bin seeding $\in \{\text{True, False}\}$; cluster all $\in \{\text{True, False}\}$
UMAP	$n_{neighbours} \in \{2, 5, 10, \dots, 50\}$; min distance $\in [0.1, 0.9]$; metric $\in \{\text{euclidean, manhattan}\}$; $d \in [2, 5]$; $n_{components} \in [2, 10]$

4 Results and Discussion

This chapter presents and interprets the experimental results obtained from the proposed Koopman-based framework. The evaluation is structured progressively. First, the motor fault detection results are analysed to validate the effectiveness of Koopman-derived features relative to raw and Fourier-based representations together with a comparison between the Fourier and Koopman estimated spectra. Next, evaluate the flow rate classification across different sensing configurations with both supervised and unsupervised clustering performance to evaluate its ability to correctly classify the system's flow rate and the intrinsic structure of the extracted feature space.

4.1 Motor Fault Detection Results

4.1.1 Spectrum Comparison

From [Figure 8](#) to [Figure 11](#) shows the Fourier and the estimated Koopman Spectrum of each motor operation state. Both measured spectra show around the same identified frequencies present in the signal, but they don't share the same amplitudes due to Fourier's amplitude measures the overall RMS/energy of frequency component across the entire time window, while Koopman's amplitude measures how strongly a dynamic mode (that oscillates at that frequency) is excited at the beginning of the observation window. Despite this, the ratio between its frequencies is preserved, therefore the proposed method can be used to identify motor faults just the as the current established Fourier method. It is important to notice that although both methods identify the same frequencies, the proposed Koopman-based features uses it and other dynamic information to characterize the system, making it more robust and richer than simple frequency identification that is used in Fourier-based methods.

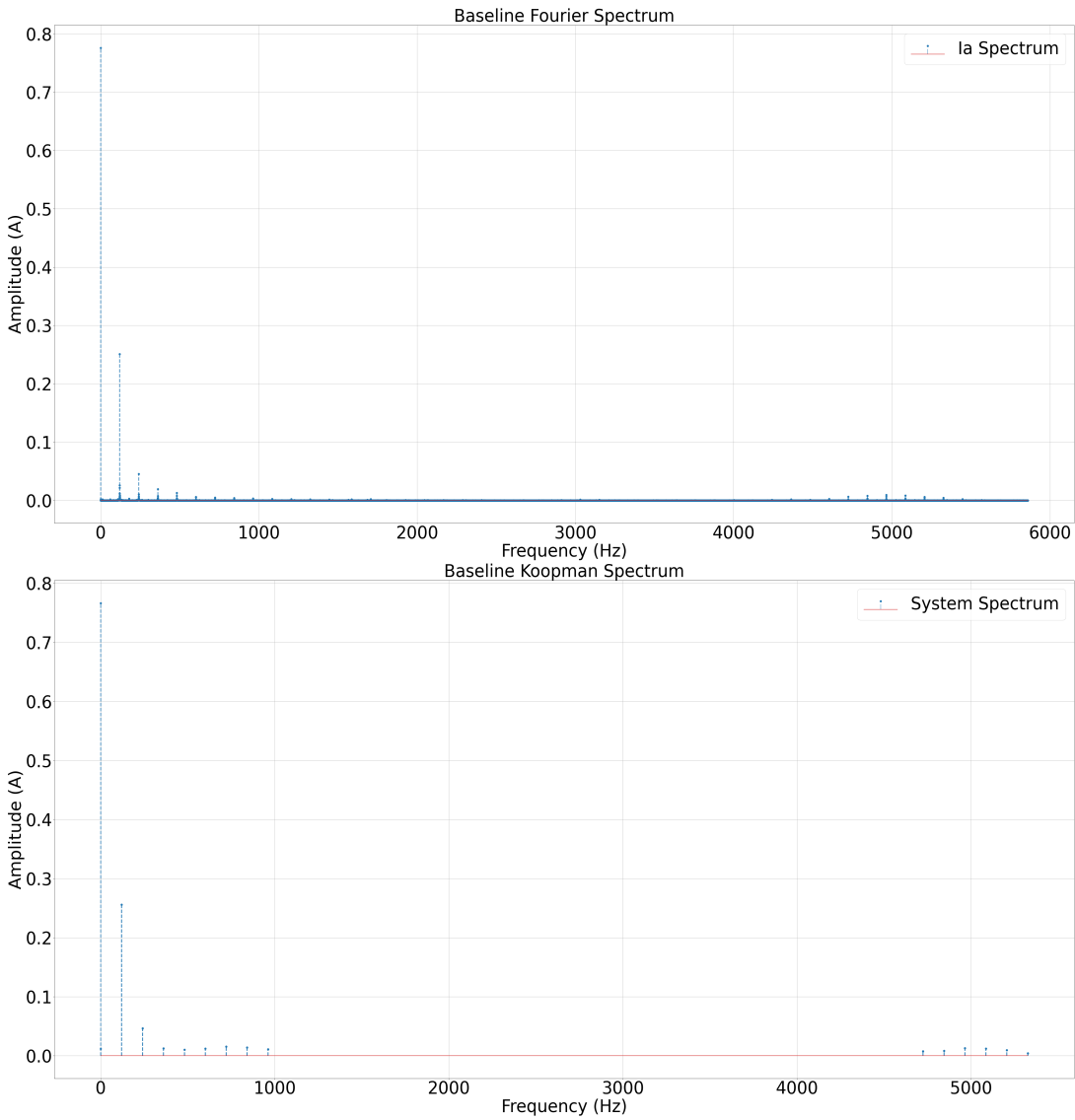


Figure 8 – Healthy Spectrum

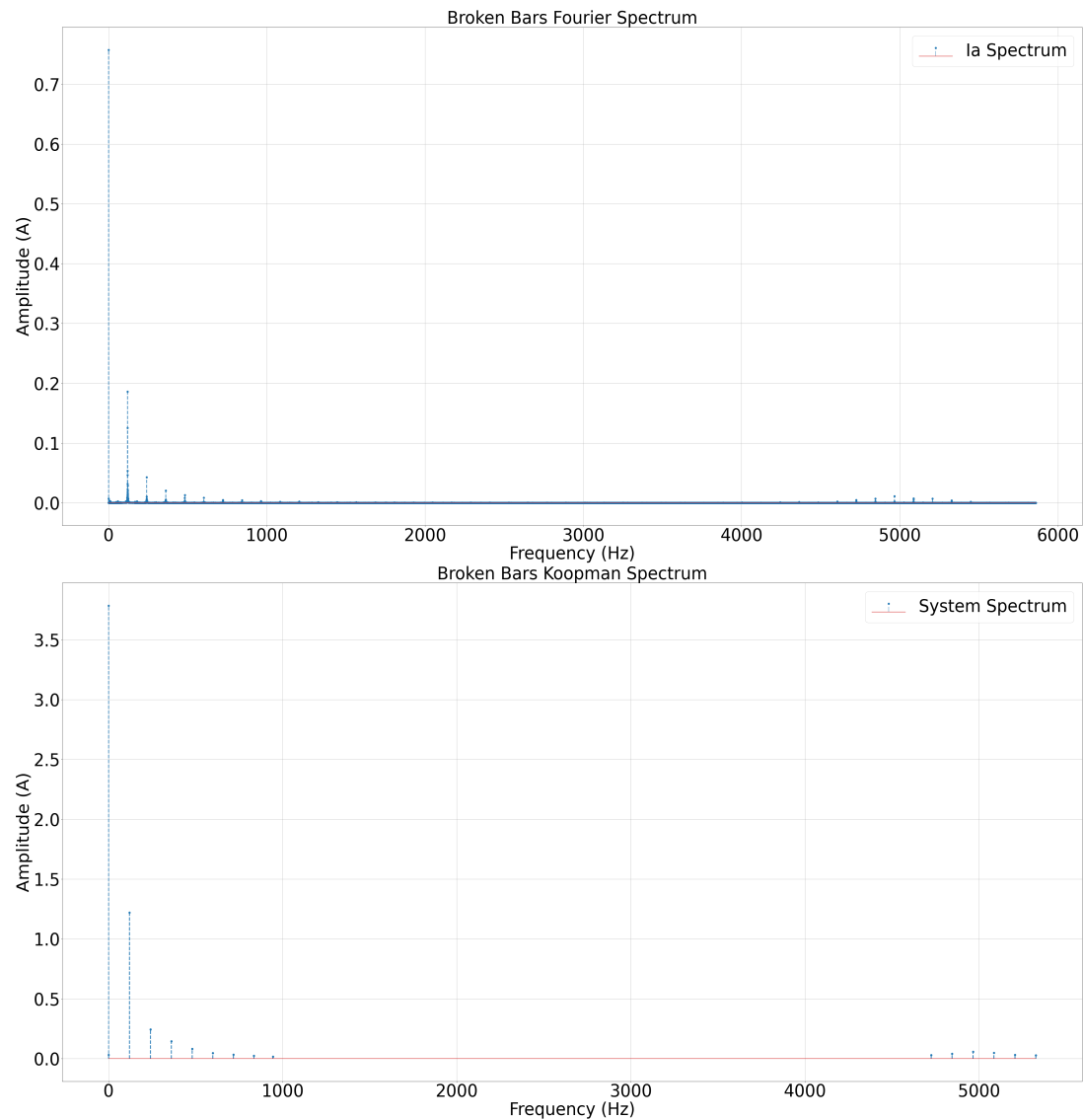


Figure 9 – Broken Bars Spectrum

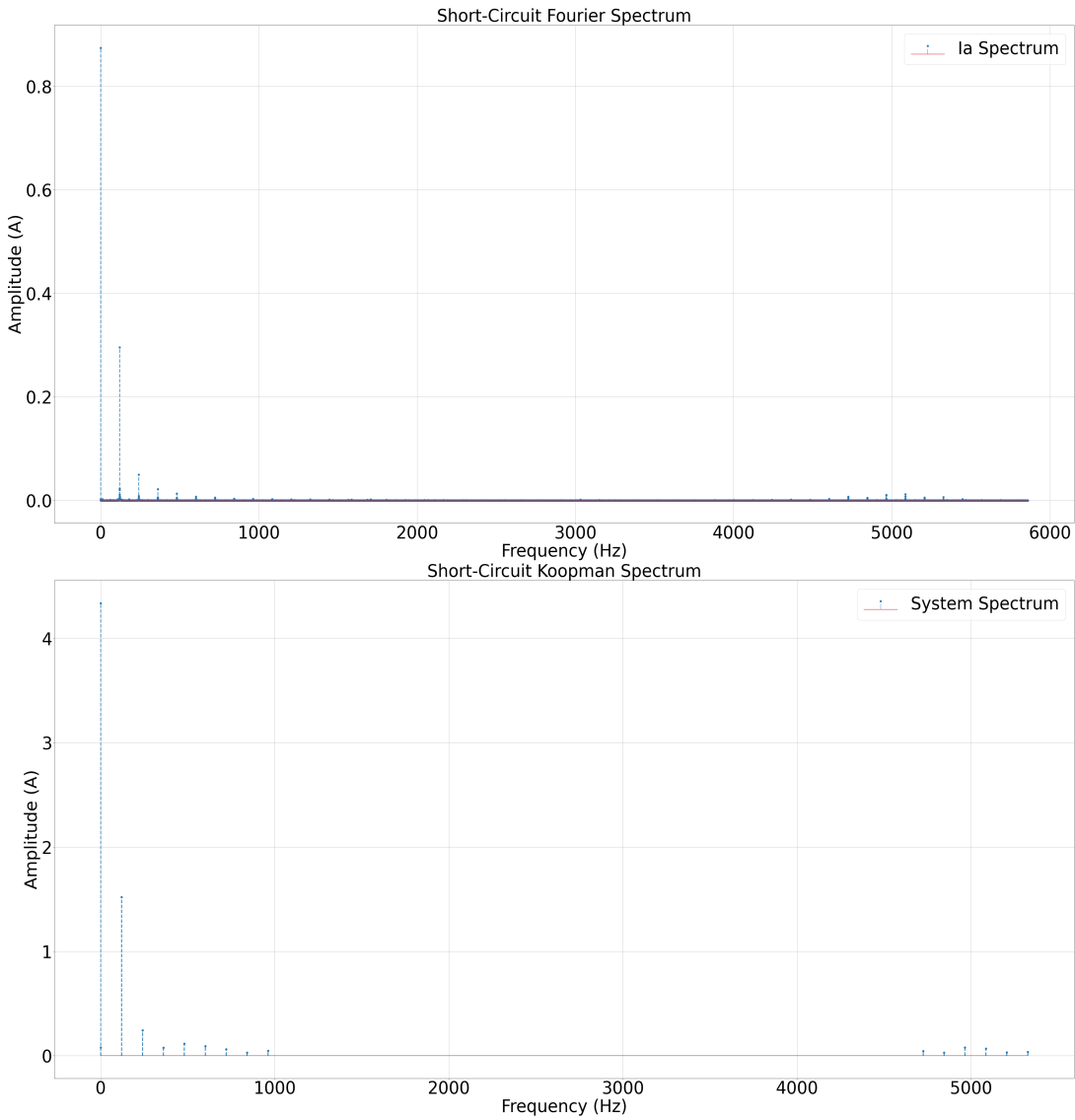


Figure 10 – Inner-turn Short-Circuit Spectrum

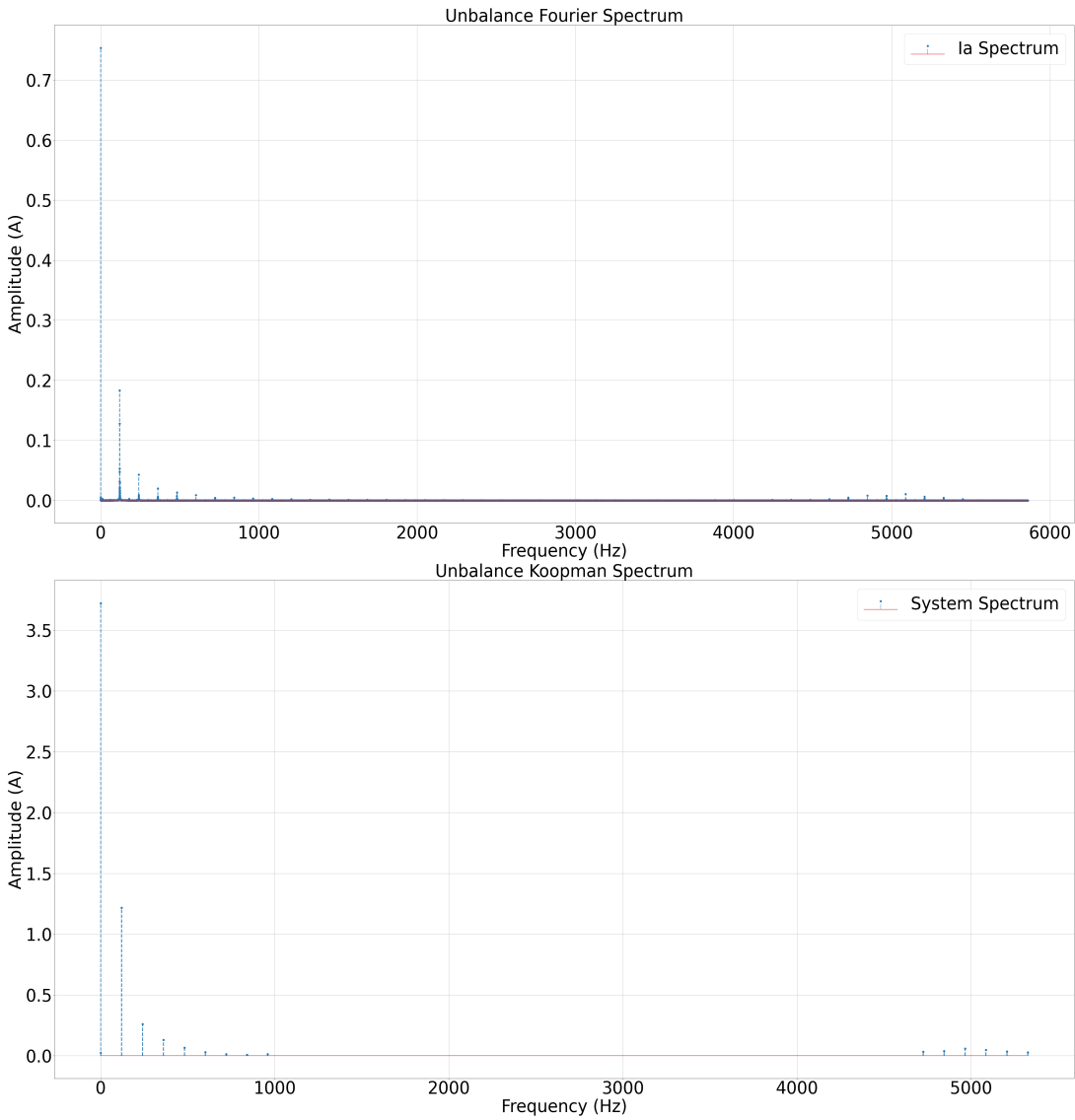


Figure 11 – Unbalance Spectrum

4.1.2 Fault Detection with Unbalance Label

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	93.2%	98.4%	92.2%	39.8%	54%	83.9%	65.6%
Test Accuracy	14.3%	14.3%	14.3%	26.2%	24.1%	14.3%	30.7%
F1 Score	0.16	0.16	0.16	0.26	0.21	0.15	0.22
ROC-AUC	0.45	0.43	0.44	0.40	0.44	0.35	0.48

Table 4 – Raw Signal Input - Unbalance

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	84.4%	72.3%	81.8%	38.3%	100%	97.6%	94.8%
Test Accuracy	68.8%	41.9%	65.2%	34.3%	22.6%	22.6%	27.4%
F1 Score	0.68	0.48	0.64	0.27	0.22	0.21	0.20
ROC-AUC	0.82	0.71	0.82	0.72	0.45	0.48	0.55

Table 5 – Fourier Signal Input - Unbalance

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	79.5%	95.1%	100%	95.1%	92.4%	96.8%	91.4%
Test Accuracy	73.6%	77.1%	80.7%	73.6%	70%	73.6%	73.6%
F1 Score	0.69	0.72	0.78	0.69	0.64	0.68	0.68
ROC-AUC	0.88	0.89	0.88	0.89	0.78	0.94	0.79

Table 6 – Koopman Signal Input - Unbalance

Tables [Table 4](#) to [Table 6](#) summarize the overall performance of all tested supervised algorithms for raw, Fourier Spectrum, and Koopman selected features as inputs for motor fault detection with the "Unbalance" label, showing that, even with high cross-validation balanced accuracy, both raw and Fourier's test balanced accuracy are quite low, indicating a model overfit and an inability to extrapolate to untrained data. The Koopman features had more promising results, with Random Forest as the best performing algorithm followed by SVM, XGBoost and k-NN all of them had cross-validation accuracy near or higher than 80% and test accuracy above 70%. The worst performing model was the Gaussian Naive Bayes algorithm due to its assumption of Gaussian distributed classes being broken, lowering its performance. However, the UMAP + SMOTE preprocessing step partially mitigates this limitation. By introducing neighbourhood-based interpolations, the resulting feature space becomes more compatible with Gaussian class-conditional modelling, allowing this algorithm to achieve relatively competitive accuracy despite its theoretical limitation.

The confusion matrix obtained using the raw time-domain inputs (Figure 12) reveals that the k-NN classifier struggled to properly distinguish the "Broken Bars" condition. Most of the samples belonging to the class were instead assigned to the "Unbalance" category, indicating that the raw signal representation does not sufficiently separate the fault signatures in the feature space. The SHAP analysis in Figure 13 shows that the fifth projected feature plays the most influential role in the decision process, followed by the sixth and seventh components. In contrast, the first, second and fourth components contribute less strongly, although they still provide relevant information to the classifier. The learning & validation curves in Figure 14 indicate stable training behaviour and show no clear signs of overfitting, while also illustrating the sensitivity of the method to the selected hyperparameters.

When the FFT representation is used (Figure 15), the classifier exhibits a different type of confusion. In this case, the "Broken Bars" condition is predominantly misclassified as the "Baseline" class, suggesting that the spectral representation alone does not fully capture the fault-specific modulation patterns. According to the SHAP interpretation in Figure 16, the first feature becomes the most relevant contributor, followed by the third and fifth components. The remaining features present smaller yet non-negligible contributions to the classification outcome. As observed in Figure 17, the learning behaviour again suggests adequate generalization capability and confirms that the chosen hyperparameter range meaningfully influences model performance.

The results obtained using the Koopman-based features (Figure 18) show a different misclassification pattern. In this case, the "Baseline" condition is frequently mistaken for "Broken Bars", which indicates that the Koopman embedding strongly emphasizes dynamical structures associated with rotor faults. The SHAP analysis in Figure 19 highlights the first projected feature as the dominant contributor to the classifier decisions, followed by the third component, while the second and fourth features exhibit comparatively smaller but still relevant effects. Finally, the learning and validation curves presented in Figure 20 confirm that the model maintains consistent generalization behaviour across the explored hyperparameter space.

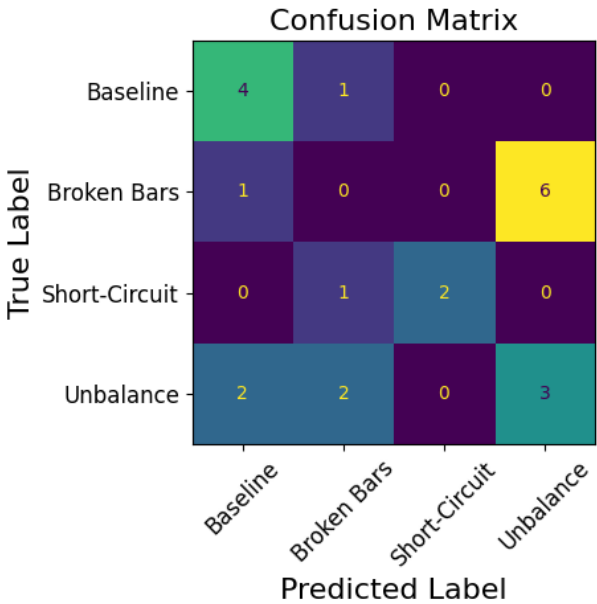


Figure 12 – k-NN Confusion Matrix - Raw

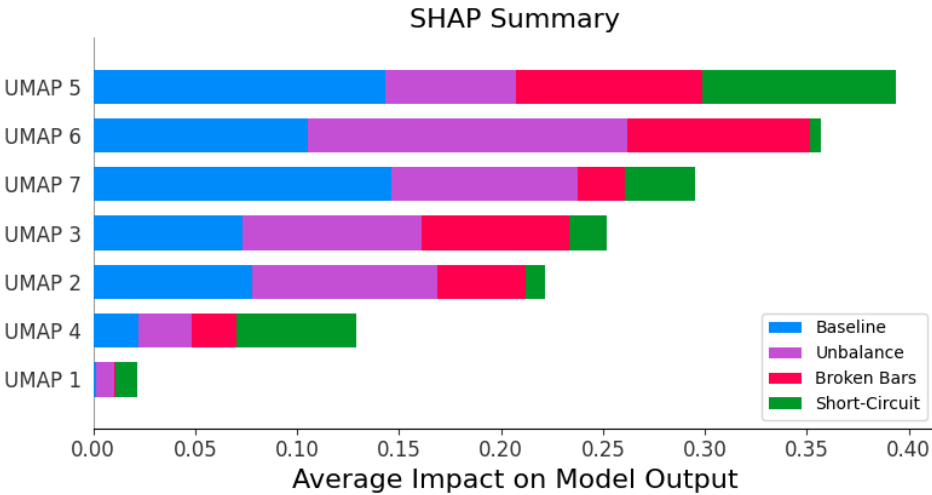


Figure 13 – k-NN SHAP Values Summary - Raw

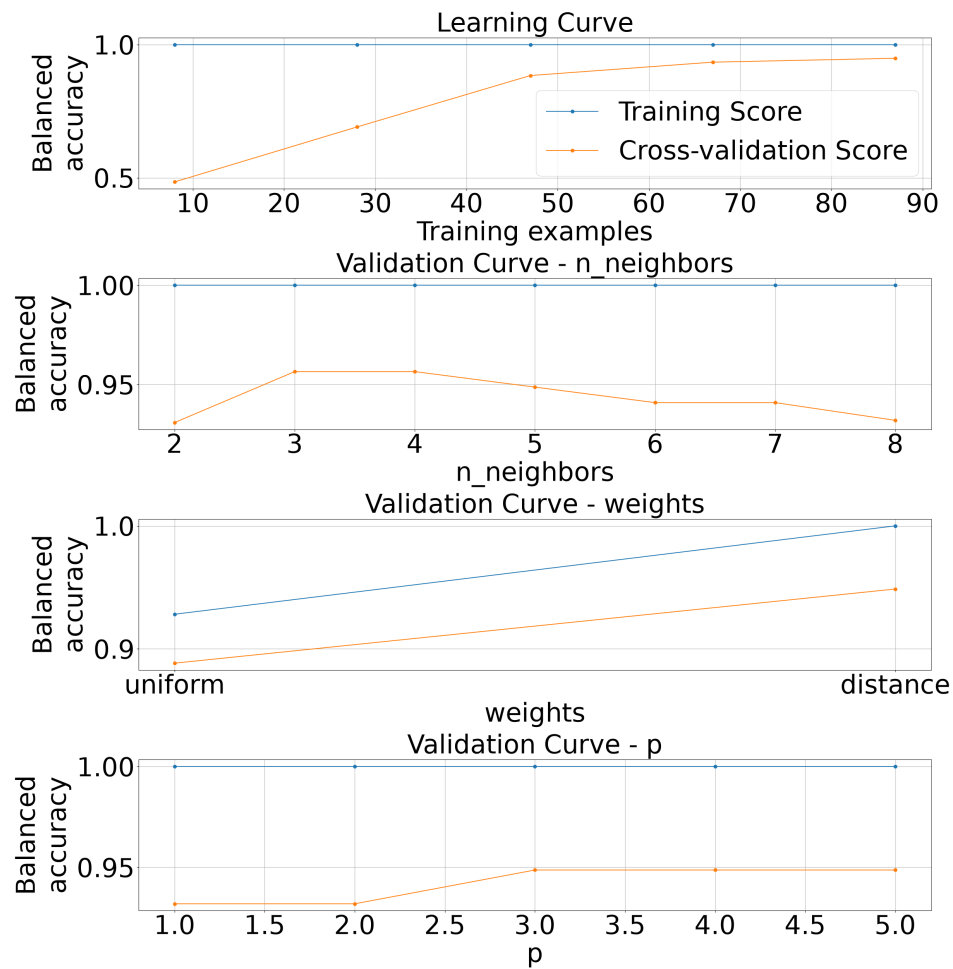


Figure 14 – k-NN Learning & Validation Curves - Raw

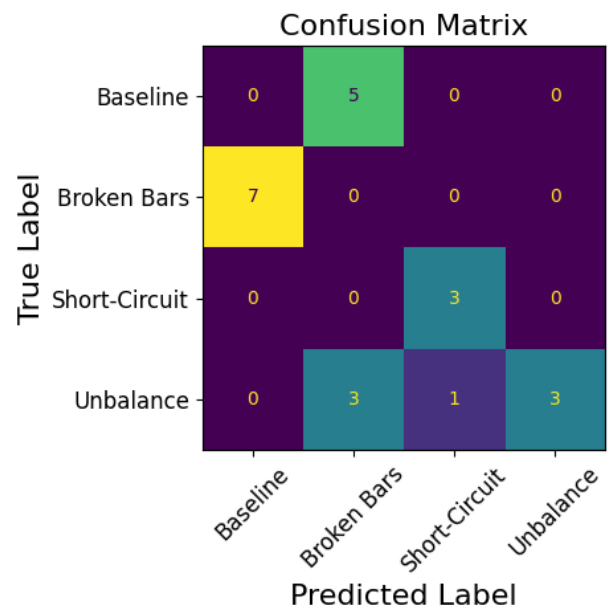


Figure 15 – k-NN Confusion Matrix - FFT

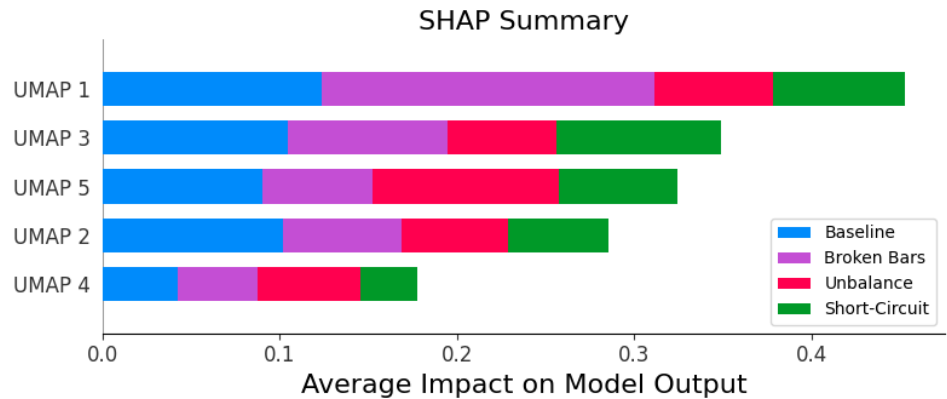


Figure 16 – k-NN SHAP Values Summary - FFT

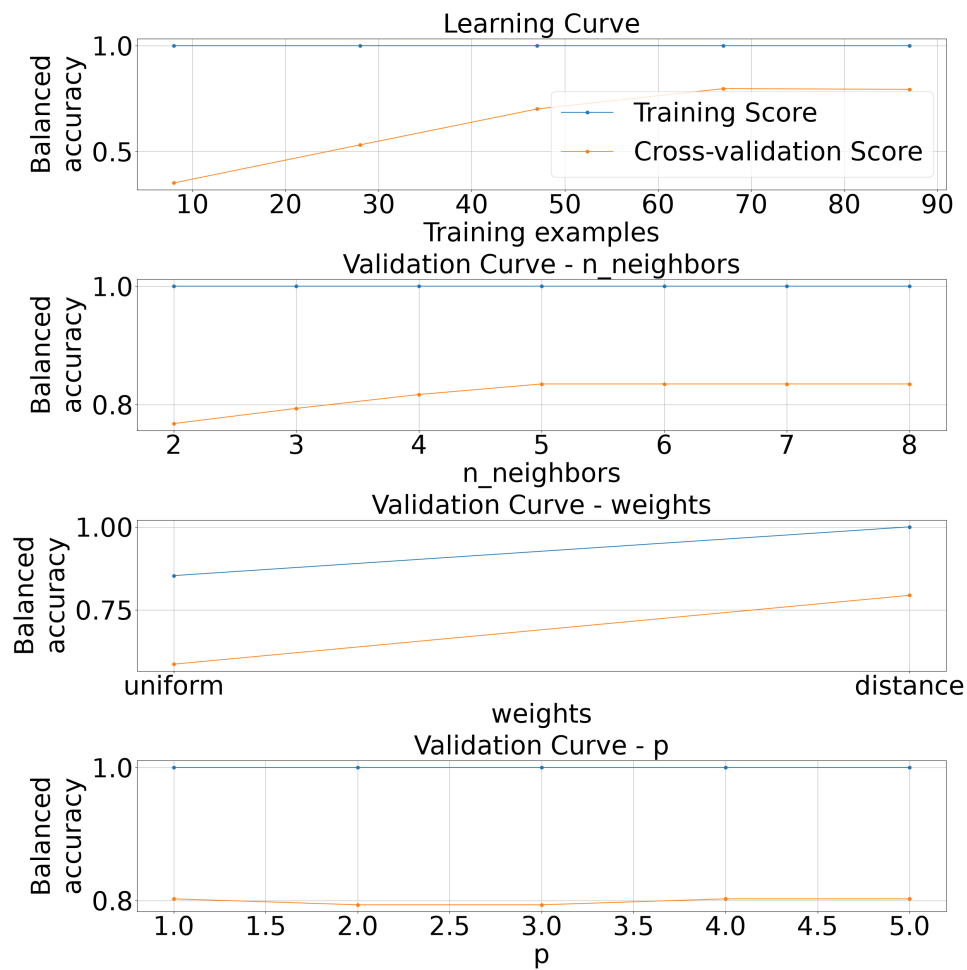


Figure 17 – k-NN Learning & Validation Curves - FFT

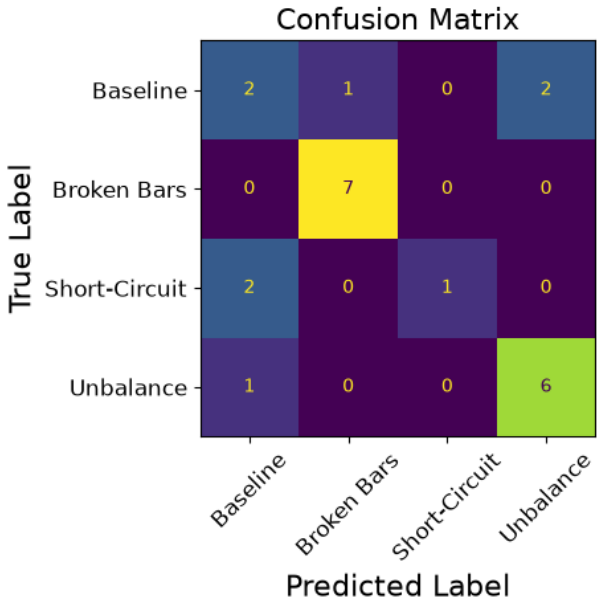


Figure 18 – k-NN Confusion Matrix - Koopman

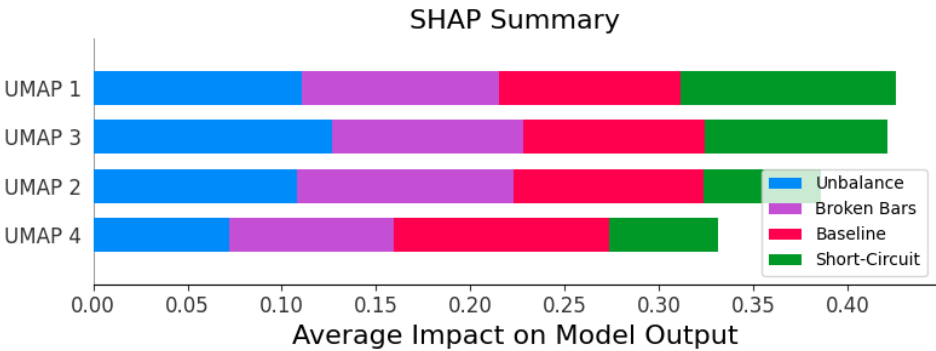


Figure 19 – k-NN SHAP Values Summary - Koopman

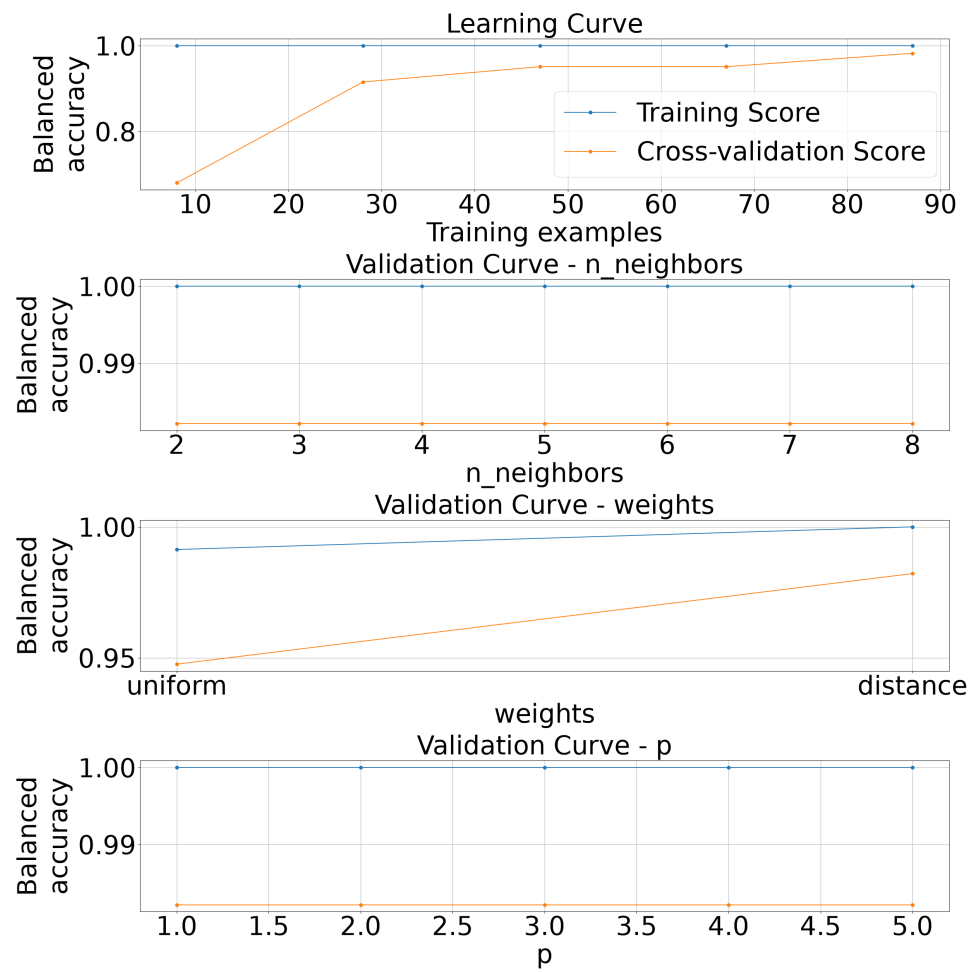


Figure 20 – k-NN Learning & Validation Curves - Koopman

The confusion matrix obtained using the raw time-domain signals (Figure 21) reveals that the SVM struggled to form a clear decision boundary for the "Broken Bars" condition. A large portion of these samples were instead assigned to the "Unbalance" class, suggesting that the raw waveform representation does not sufficiently highlight discriminative signatures of rotor asymmetry. The SHAP analysis in Figure 22 indicates that the classifier relied primarily on the second projected feature, followed by the fifth and fourth, while the remaining features contributed more modestly but still played a non-negligible role. The corresponding learning and validation curves (Figure 23) suggest the presence of mild overfitting and demonstrate a noticeable sensitivity to hyperparameter selection.

When the same classifier was trained using FFT-based inputs, the misclassification pattern changed considerably. As shown in Figure 24, most "Broken Bars" samples were reassigned to either the "Short-Circuit" or Unbalance classes, with a clear bias toward the former. This behaviour indicates that although spectral features provide richer frequency information than raw signals, they still fail to isolate rotor-bar damage effectively. The SHAP interpretation (Figure 25) shows a different feature importance hierarchy, where the fifth and eighth projected features dominate the decision process, followed by the sixth and second. The learning and validation curves in Figure 26 exhibit stronger divergence than in the raw-signal case, pointing to a more pronounced overfitting tendency. Using Koopman-based features produced a distinct behaviour pattern.

According to the confusion matrix in Figure 27, the classifier continued to experience difficulty isolating the "Broken Bars" condition; however, the dominant confusion shifted toward the "Short-Circuit" class rather than "Unbalance". The SHAP results in Figure 28 indicate that the classification decision is largely driven by the first projected feature, followed by the fourth and third, while the second feature contributes comparatively little. The associated learning and validation curves (Figure 29) again show signs of moderate overfitting and confirm the influence of hyperparameter tuning on generalization performance.

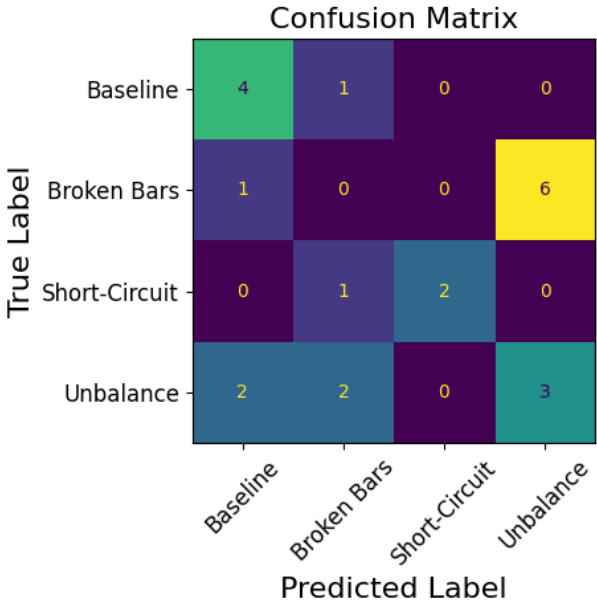


Figure 21 – SVM Confusion Matrix - Raw

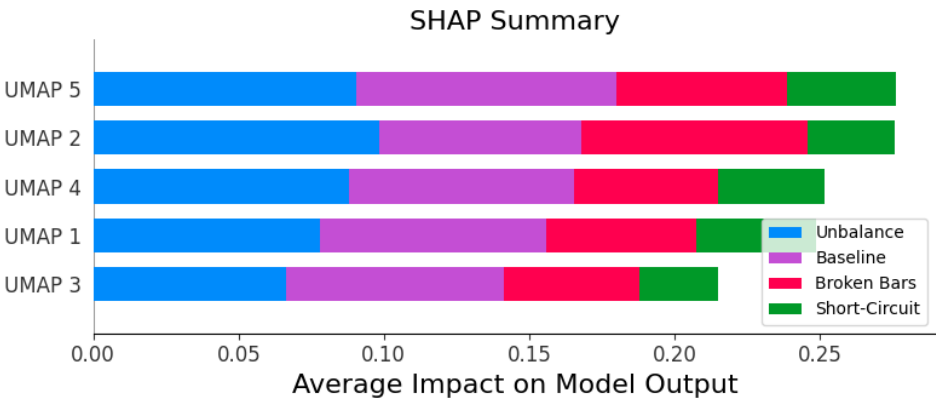


Figure 22 – SVM SHAP Values Summary - Raw

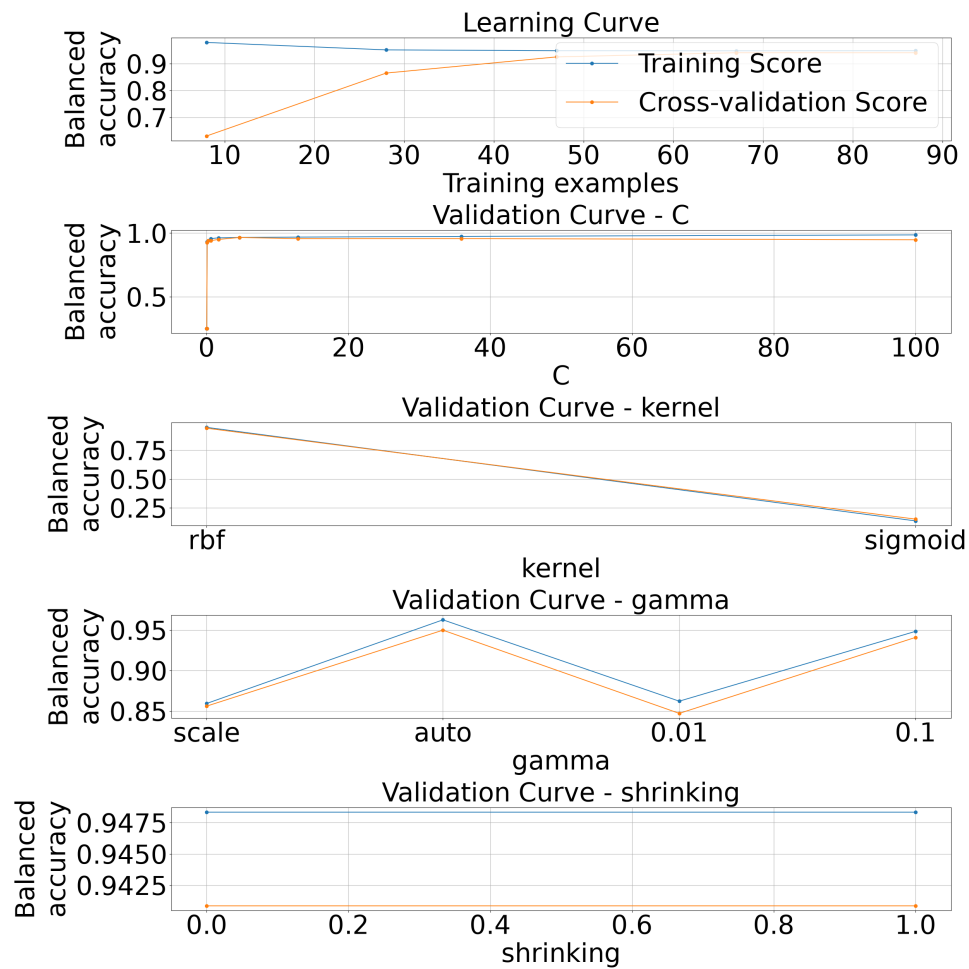


Figure 23 – SVM Learning & Validation Curves - Raw

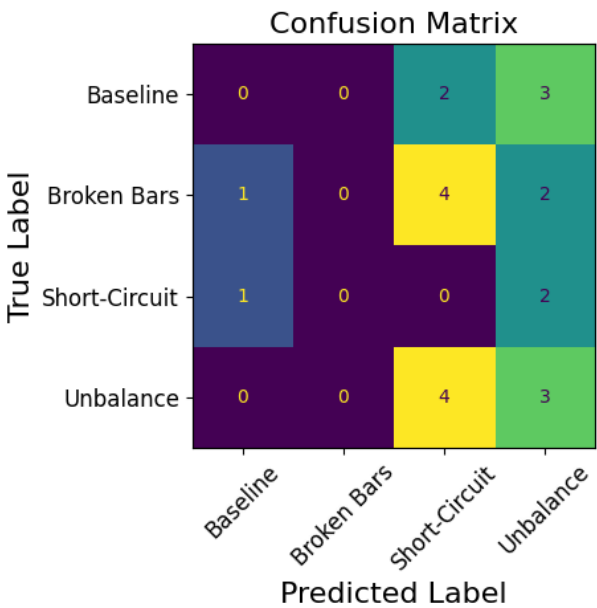


Figure 24 – SVM Confusion Matrix - FFT

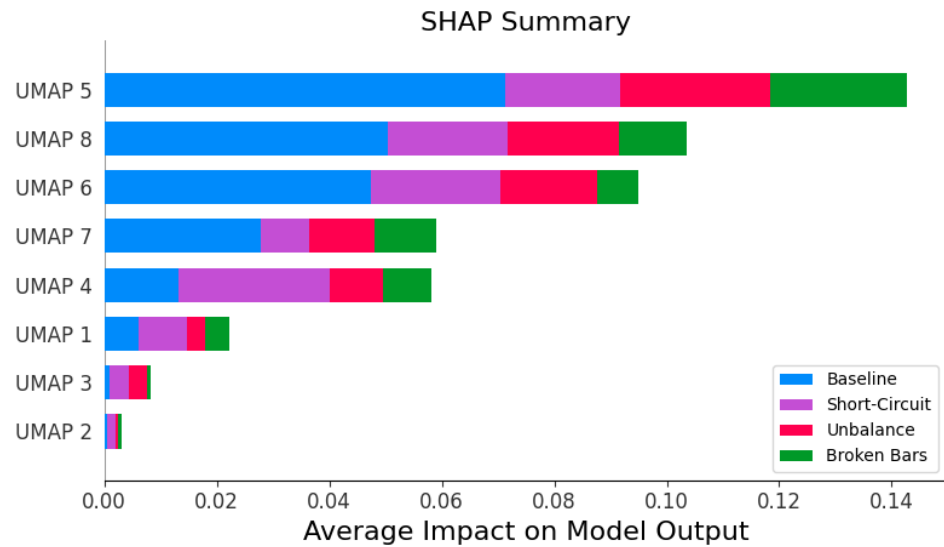


Figure 25 – SVM SHAP Values Summary - FFT

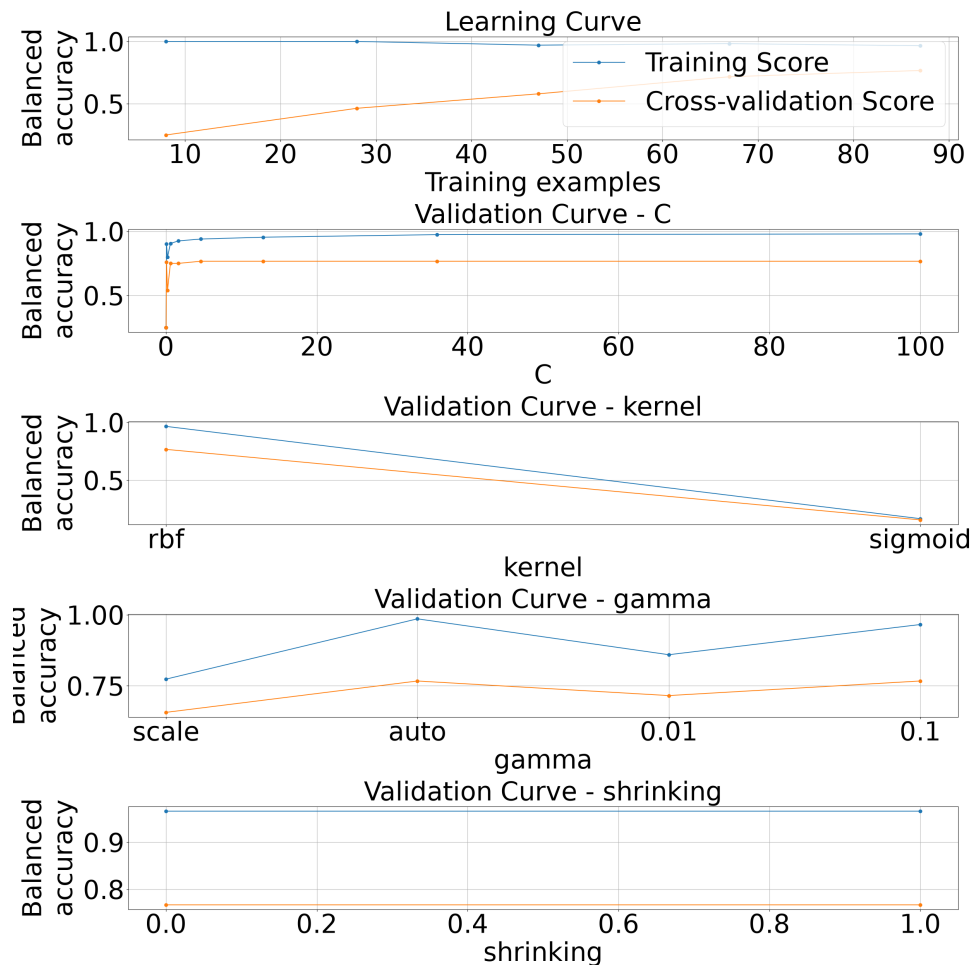


Figure 26 – SVM Learning & Validation Curves - FFT

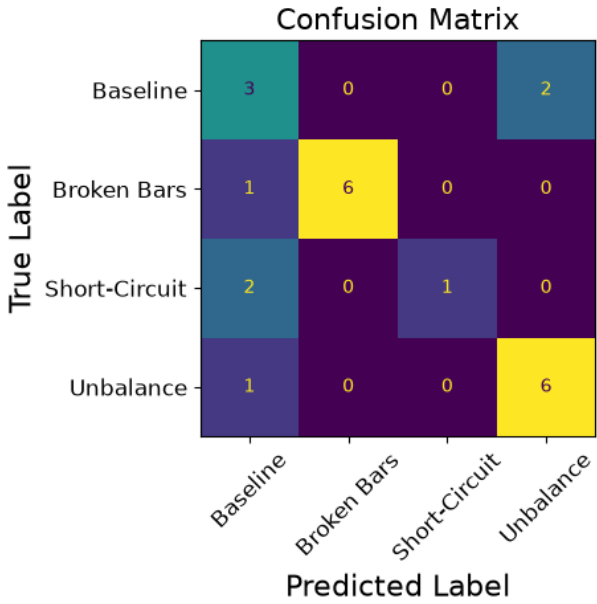


Figure 27 – SVM Confusion Matrix - Koopman

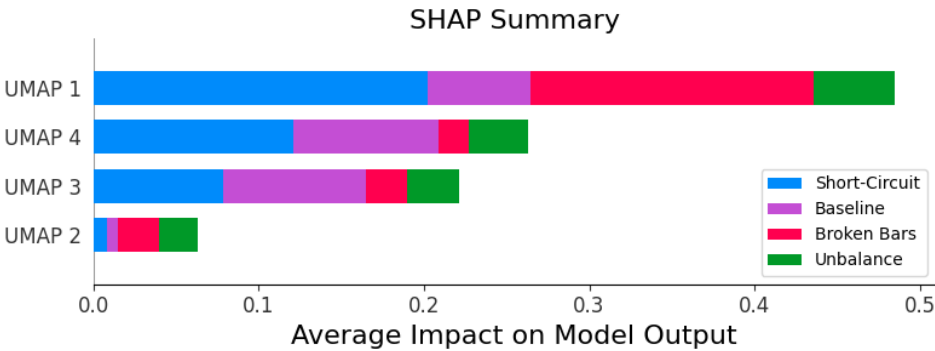


Figure 28 – SVM SHAP Values Summary - Koopman

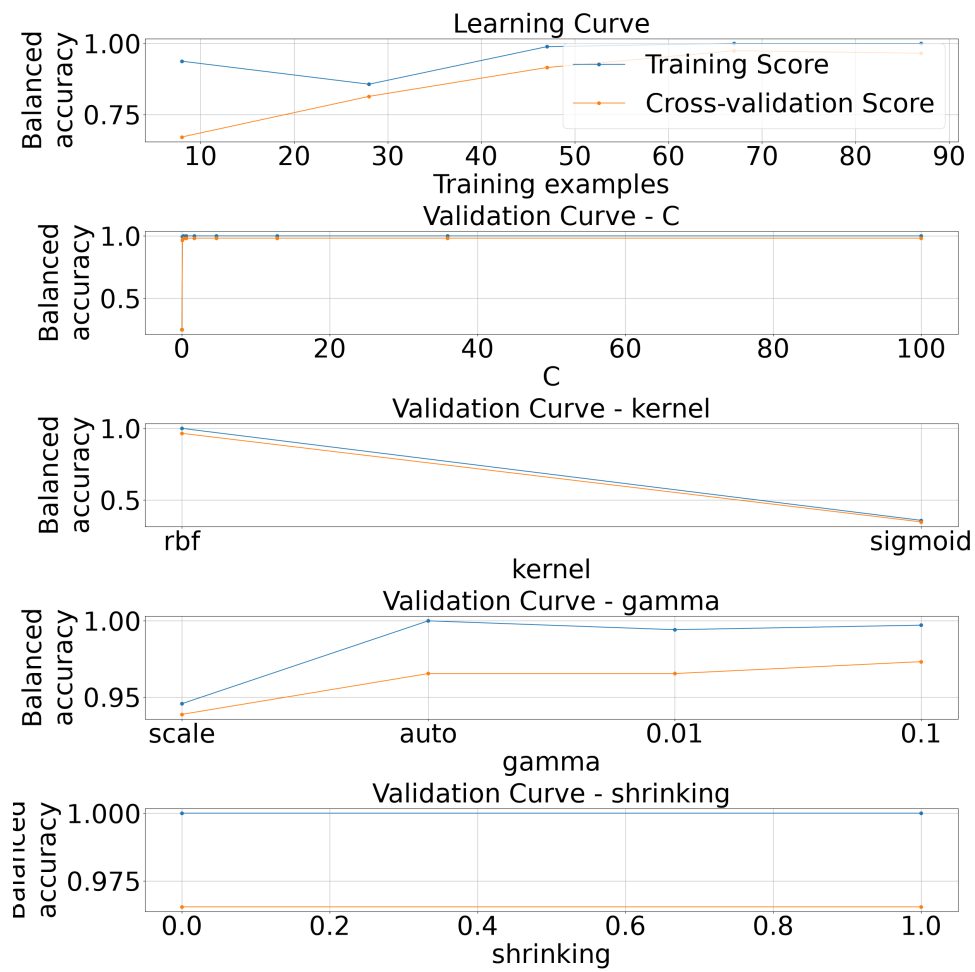


Figure 29 – SVM Learning & Validation Curves - Koopman

Figure 30 illustrates the class-wise performance obtained when Random Forest was trained directly on the raw current signals. The confusion matrix reveals a clear difficulty in separating the "Broken Bars" condition, which is predominantly misidentified as "Unbalance". This suggests that the raw time-domain representation does not sufficiently highlight the distinguishing characteristics of rotor asymmetry. The feature-importance analysis shown in Figure 31 indicates that the third projected feature plays the most prominent role in the decision process, followed by the first and second components, while the fourth and fifth features contribute less strongly but remain relevant. The learning and validation curves in Figure 32 show a noticeable gap between training and validation performance, pointing to moderate overfitting and a strong dependence on hyperparameter selection.

When the classifier was trained using Fourier-domain features, the behaviour changed significantly. As presented in Figure 33, the model collapsed to predicting nearly all samples as "Broken Bars", indicating a severe loss of discriminative capability across classes. The SHAP analysis in Figure 34 shows a redistribution of feature relevance, with the first and third components dominating the model decisions, followed by the fifth component, while the remaining features have comparatively smaller influence. The corresponding learning curves in Figure 35 reveal pronounced overfitting, confirming that the spectral representation alone is insufficiently robust for reliable fault separation in this scenario. The results obtained with Koopman-based features exhibit a distinct behaviour.

According to the confusion matrix in Figure 36, the classifier shows improved generalization compared to the Fourier case, although confusion remains between the "Baseline", "Broken Bars", and "Short-Circuit" classes, with many samples from the first two being predicted as "Short-Circuit". The feature-importance visualization in Figure 37 indicates that the third Koopman feature again dominates the decision process, followed by the first, while the remaining features provide secondary yet meaningful contributions. Importantly, the learning and validation curves in Figure 38 do not exhibit clear signs of overfitting, suggesting that the Koopman representation provides a more stable and generalizable feature space for the Random Forest classifier.

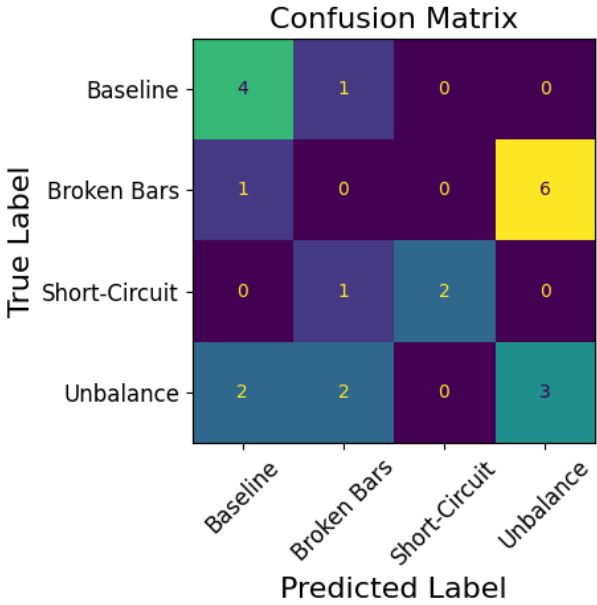


Figure 30 – Random Forest Confusion Matrix - Raw

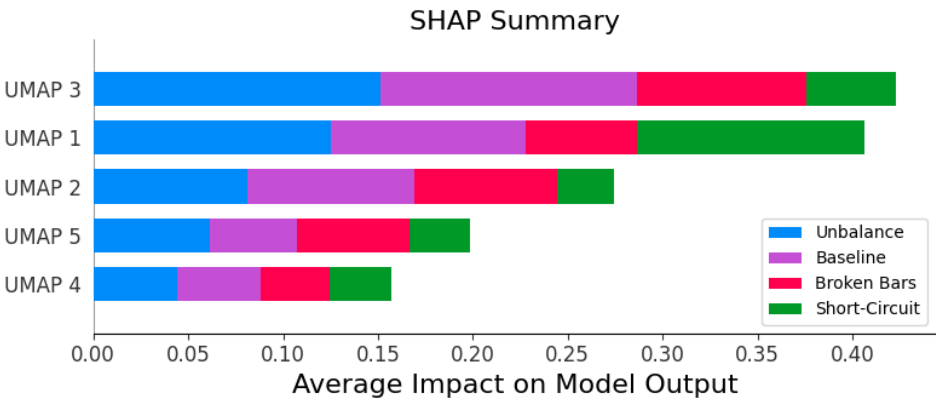


Figure 31 – Random Forest SHAP Values Summary - Raw

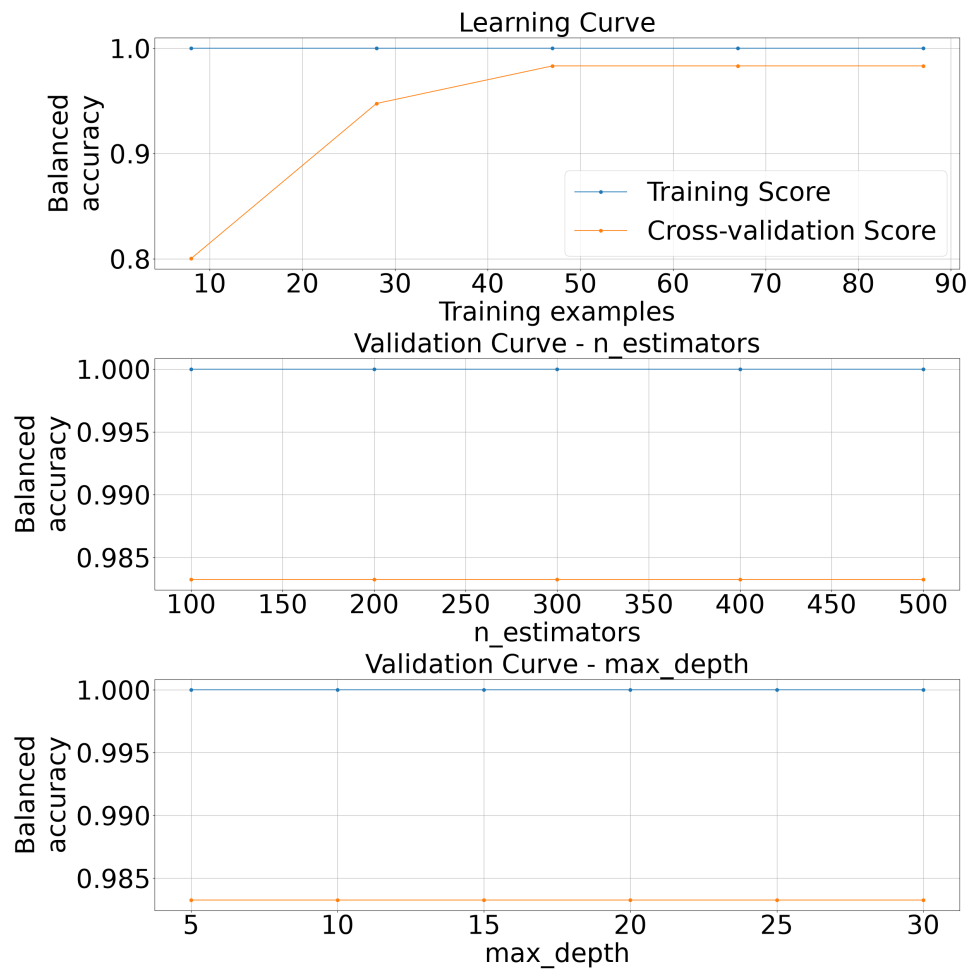


Figure 32 – Random Forest Learning & Validation Curves - Raw

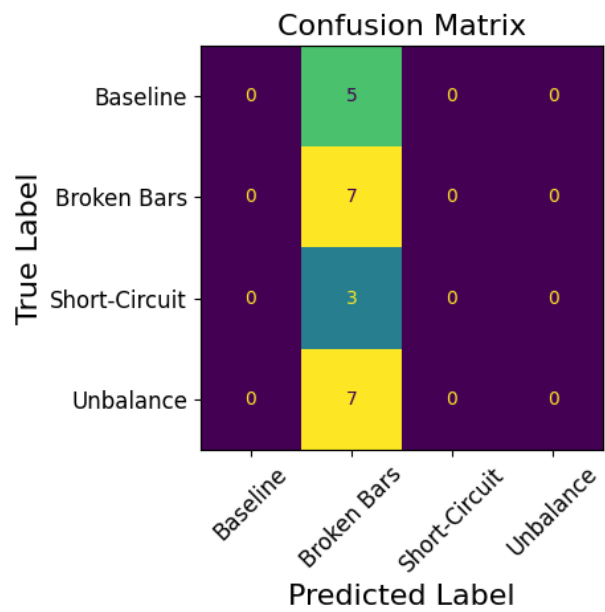


Figure 33 – Random Forest Confusion Matrix - FFT

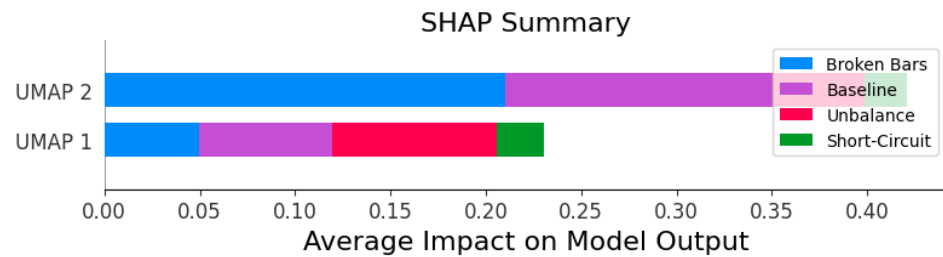


Figure 34 – Random Forest SHAP Values Summary - FFT



Figure 35 – Random Forest Learning & Validation Curves - FFT

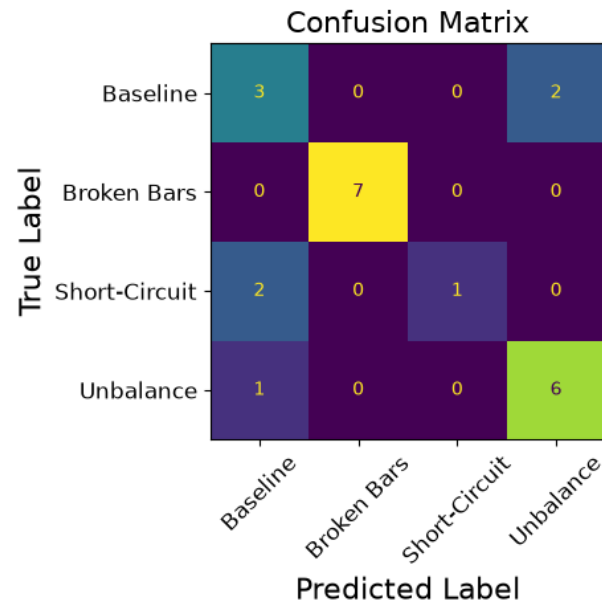


Figure 36 – Random Forest Confusion Matrix - Koopman

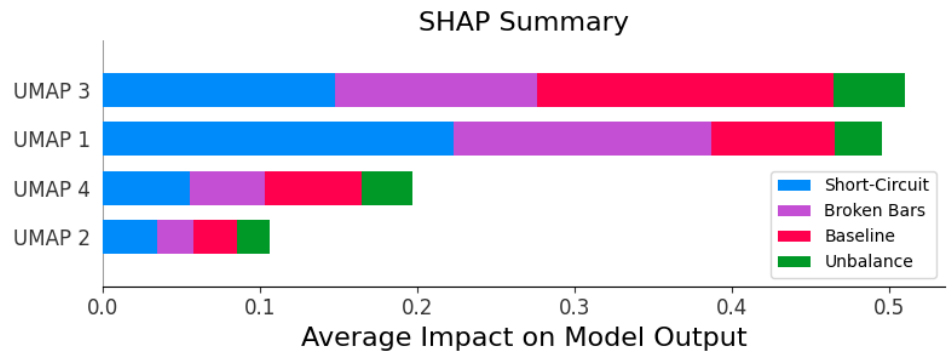


Figure 37 – Random Forest SHAP Values Summary - Koopman

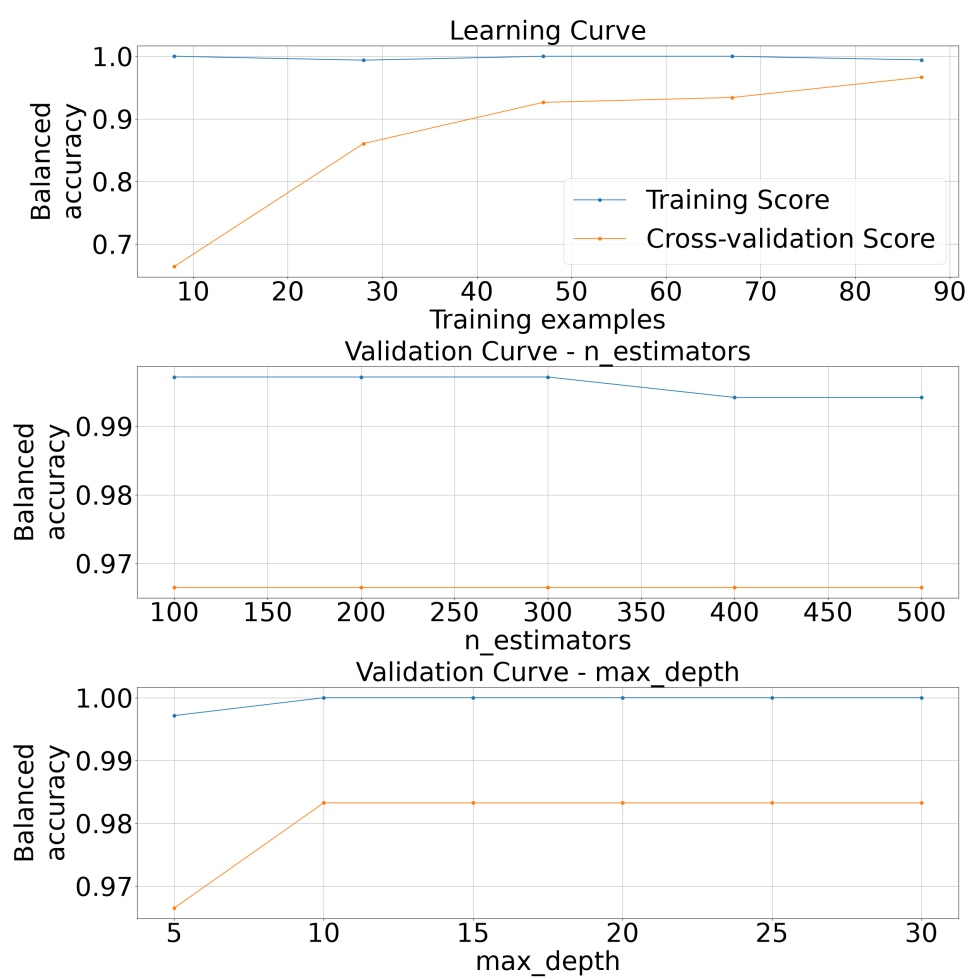


Figure 38 – Random Forest Learning & Validation Curves - Koopman

Figure 39 illustrates the class-wise performance of Logistic Regression when trained using the raw signal inputs. The model struggled to generalize across most categories, achieving reasonable recognition primarily for the "Unbalance" condition while exhibiting significant confusion among the remaining classes. The feature-importance analysis in Figure 40 indicates that the fifth latent feature contributed most strongly to the decision process, followed by the second and fourth components. The first and third features played a comparatively smaller yet still non-negligible role. The learning and validation curves shown in Figure 41 suggest the presence of mild overfitting and reveal a noticeable sensitivity of the model to hyperparameter selection.

When Fourier-based features were used, the confusion matrix in Figure 42 shows a different failure mode. In this case, the classifier had particular difficulty identifying the "Broken Bars" condition, with a strong bias toward predicting the "Baseline" class. The SHAP analysis in Figure 43 reveals that the first and third features dominated the classification decisions, while the remaining components had smaller but still measurable influence. The training curves in Figure 44 display clearer signs of overfitting, indicating that the model struggled to achieve stable generalization with this representation.

In contrast, the results obtained using Koopman-based features present a different behaviour. As seen in Figure 45, the classifier's main difficulty shifted toward the "Unbalance" class, which was frequently confused with the "Baseline" condition. The feature-importance analysis in Figure 46 shows that the first Koopman feature dominated the decision process, followed by the second, while the remaining features contributed more modestly. Notably, the learning and validation curves in Figure 47 do not exhibit clear signs of overfitting, suggesting improved stability of the learning process compared with the raw and Fourier representations.

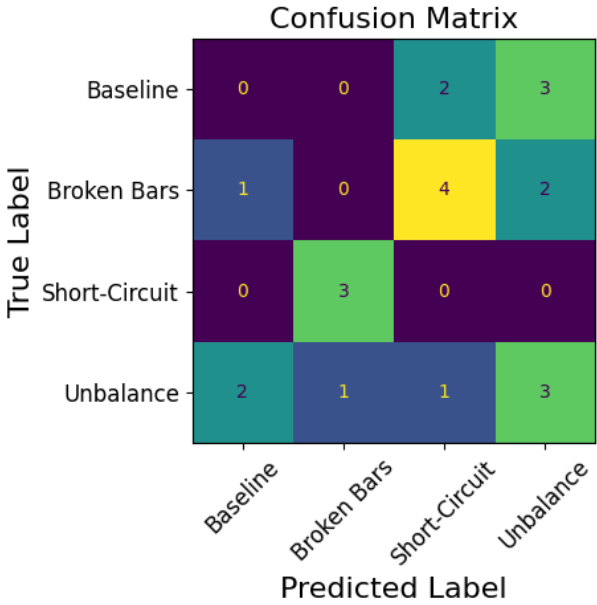


Figure 39 – Logistic Regression Confusion Matrix - Raw

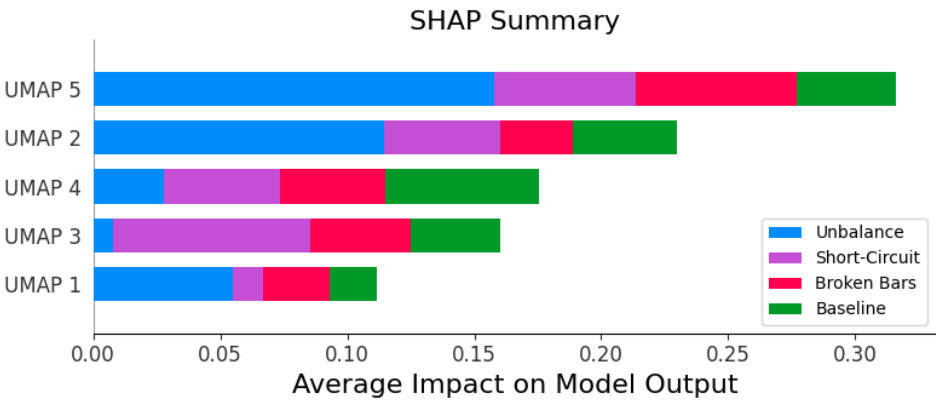


Figure 40 – Logistic Regression SHAP Values Summary - Raw

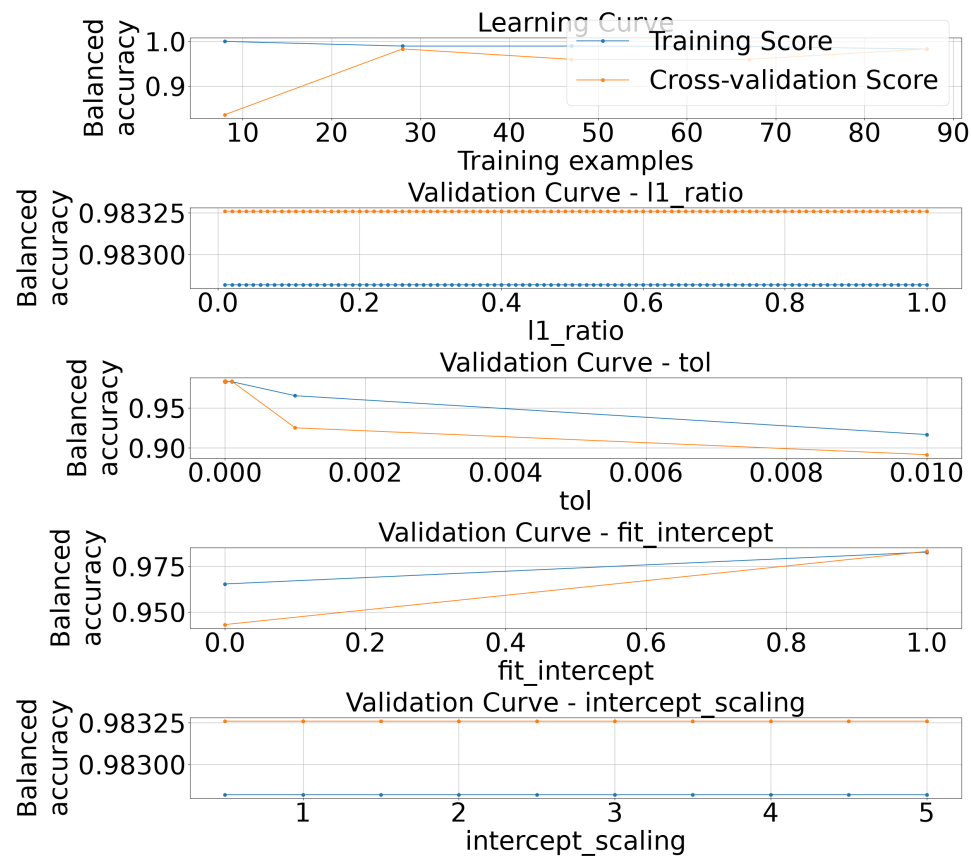


Figure 41 – Logistic Regression Learning & Validation Curves - Raw

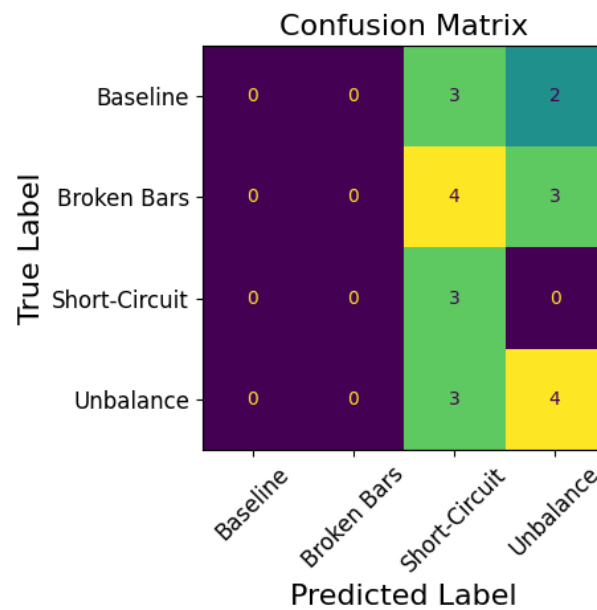


Figure 42 – Logistic Regression Confusion Matrix - FFT

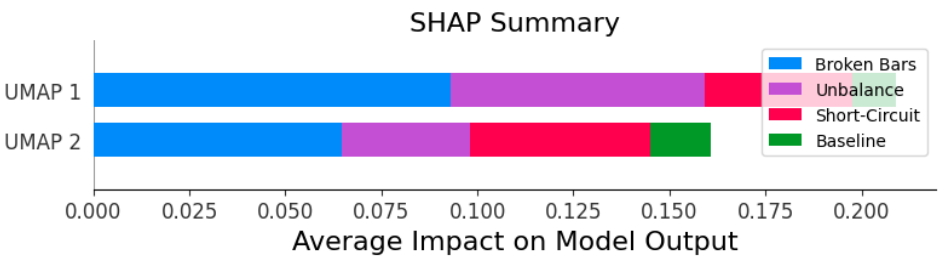


Figure 43 – Logistic Regression SHAP Values Summary - FFT

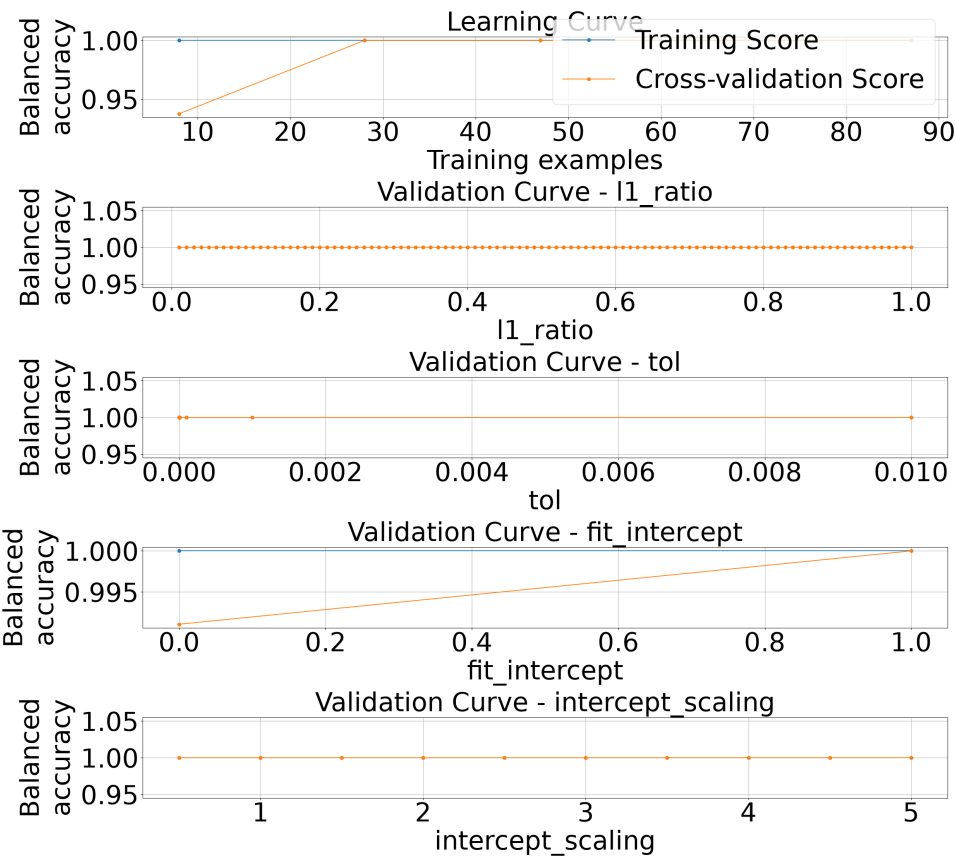


Figure 44 – Logistic Regression Learning & Validation Curves - FFT

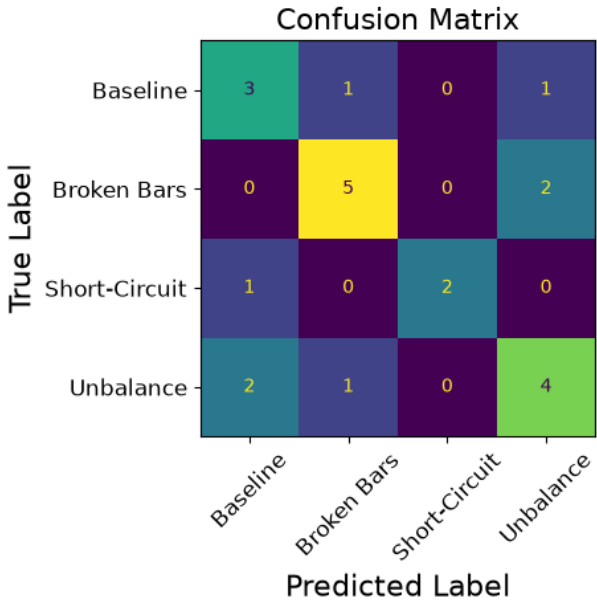


Figure 45 – Logistic Regression Confusion Matrix - Koopman

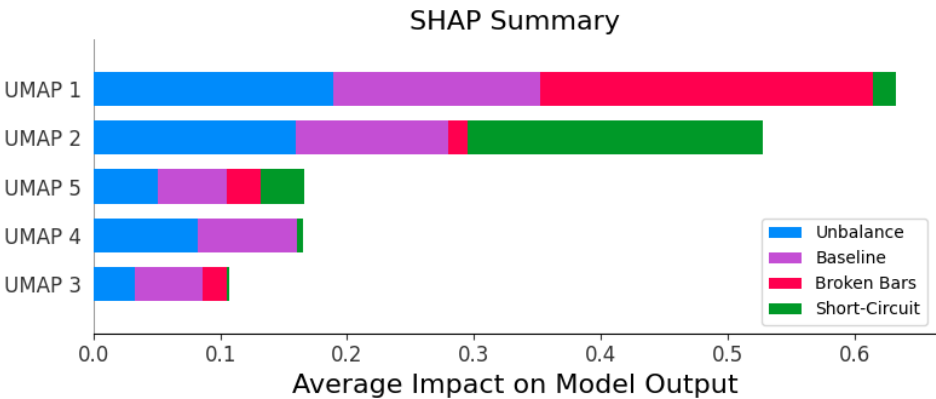


Figure 46 – Logistic Regression SHAP Values Summary - Koopman

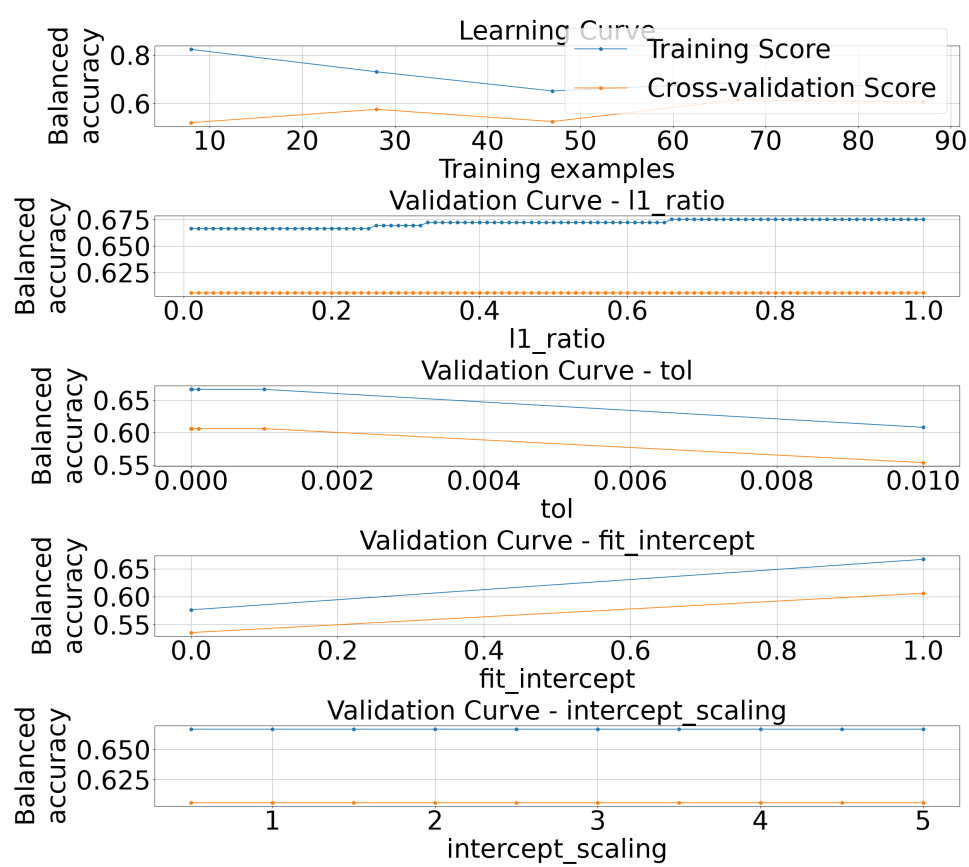


Figure 47 – Logistic Regression Learning & Validation Curves - Koopman

Figure 48 illustrates the confusion matrix obtained when Gaussian Naive Bayes is trained using the raw current signals. The results indicate a pronounced difficulty in correctly identifying the "Baseline" condition, which is frequently misclassified as the "Short-Circuit" fault. The feature attribution analysis in Figure 49 reveals that the first projected component dominates the decision process, followed by the third component, while the second contributes only marginally. The corresponding learning and validation curves in Figure 50 suggest the presence of overfitting and highlight the sensitivity of model performance to the selected hyperparameters.

When Fourier-domain inputs are employed, the classifier behaviour changes significantly, as shown in Figure 51. In this scenario, the model collapses toward predicting the "Baseline" class for most samples, indicating poor class separability in the FFT feature space. The SHAP analysis in Figure 52 confirms that only the first projected feature meaningfully influences the predictions, while the second has minimal impact. The learning and validation curves in Figure 53 again reveal strong overfitting, reinforcing the limited discriminative power of these features for this task.

In contrast, the results obtained with Koopman-based features (Figure 54) demonstrate a markedly different behaviour. The classifier is able to separate most fault classes more effectively, although some confusion remains between the "Broken Bars" and "Unbalance" conditions. The feature attribution analysis in Figure 55 shows a more balanced contribution among the projected components, with the first feature still being the most influential, followed by the third, and the second playing a negligible role. Importantly, the learning and validation curves in Figure 56 no longer indicate significant overfitting, suggesting improved generalization when Koopman features are used.

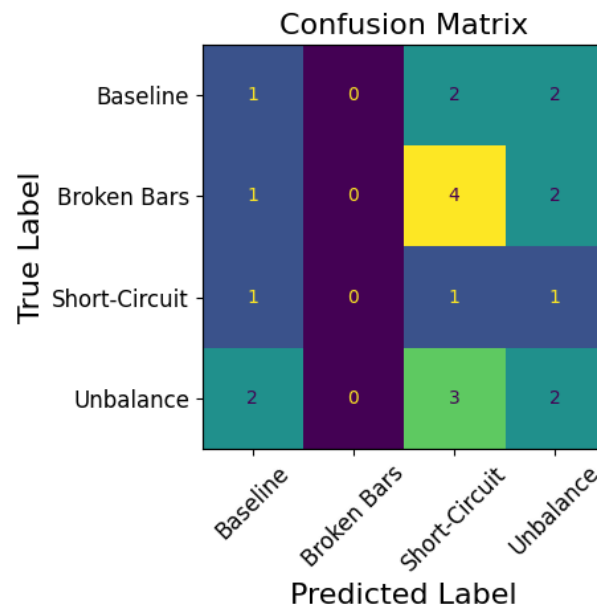


Figure 48 – Gaussian Naive Bayes Confusion Matrix - Raw

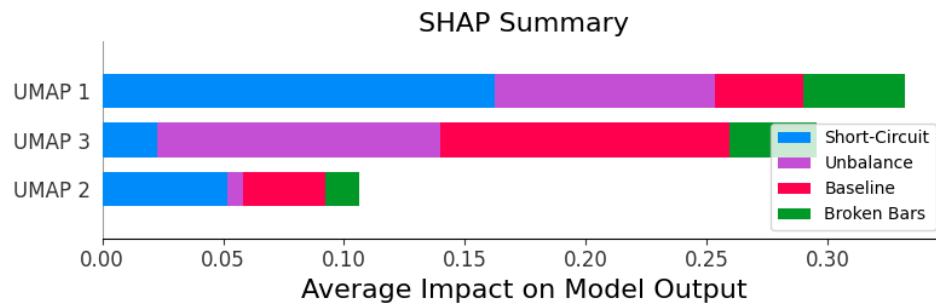


Figure 49 – Gaussian Naive Bayes SHAP Values Summary - Raw

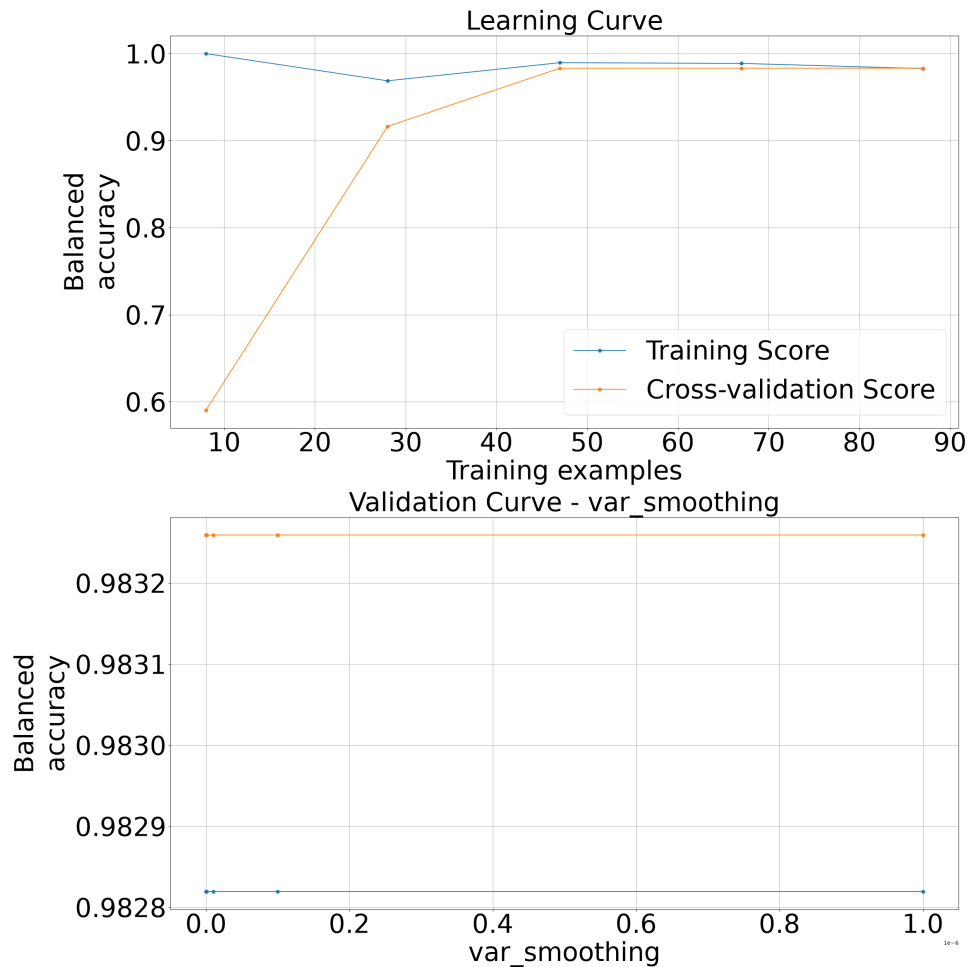


Figure 50 – Gaussian Naive Bayes Learning & Validation Curves - Raw

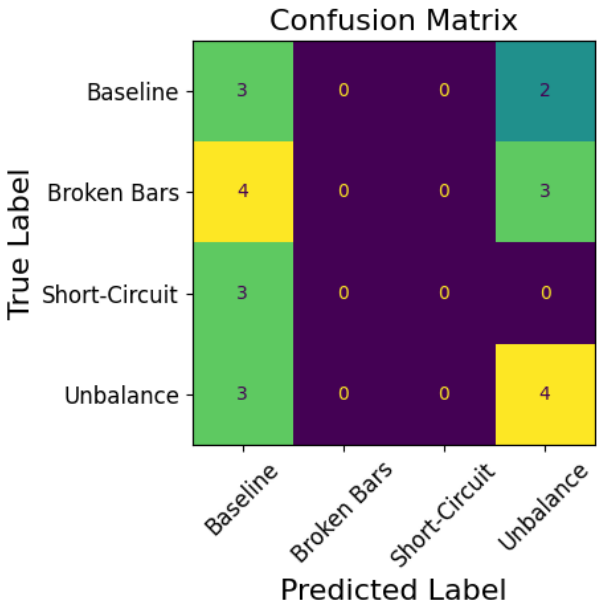


Figure 51 – Gaussian Naive Bayes Confusion Matrix - FFT

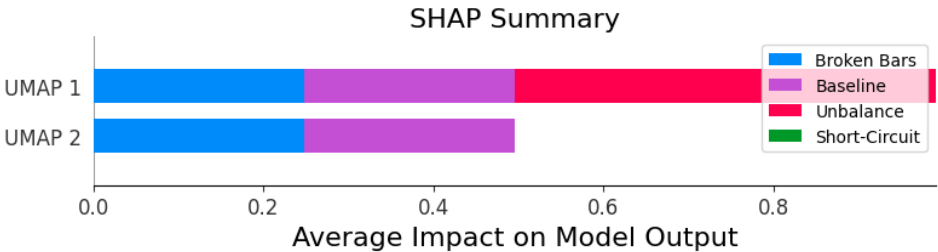


Figure 52 – Gaussian Naive Bayes SHAP Values Summary - FFT

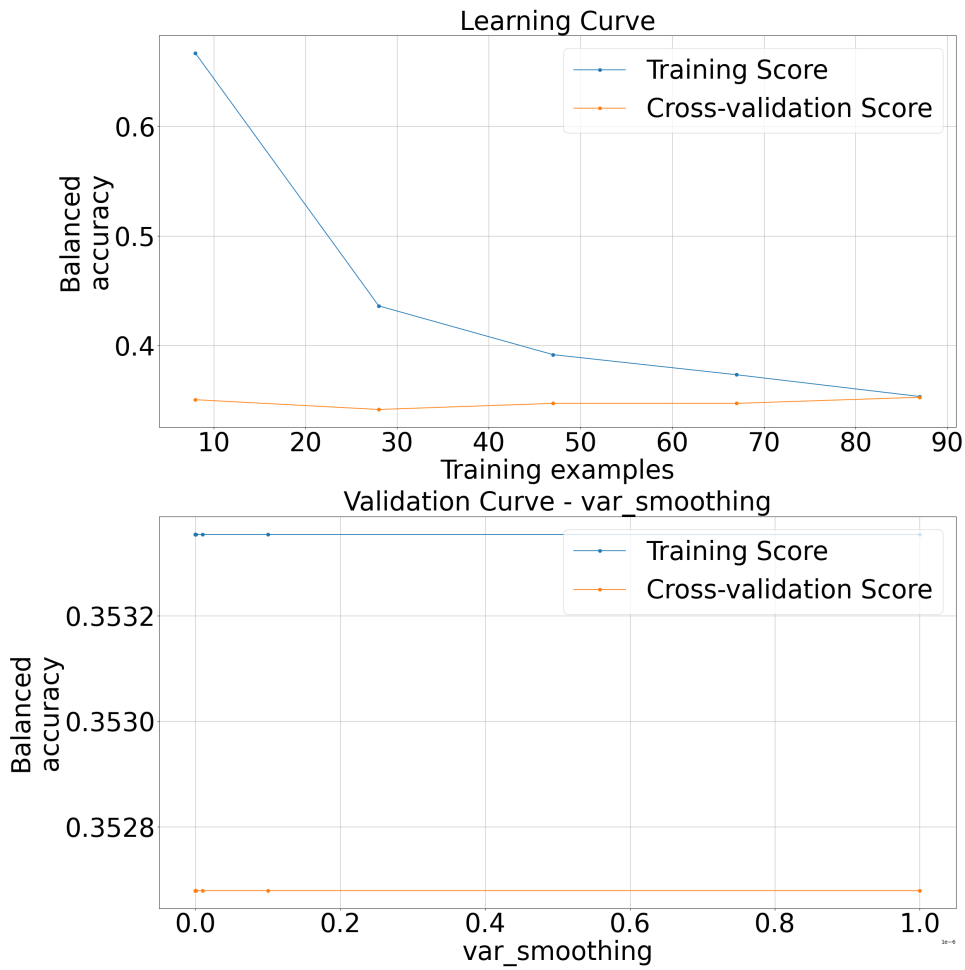


Figure 53 – Gaussian Naive Bayes Learning & Validation Curves - FFT

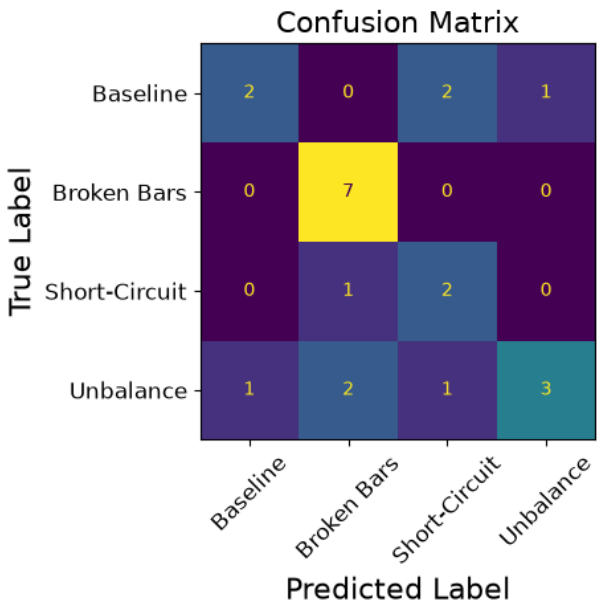


Figure 54 – Gaussian Naive Bayes Confusion Matrix - Koopman

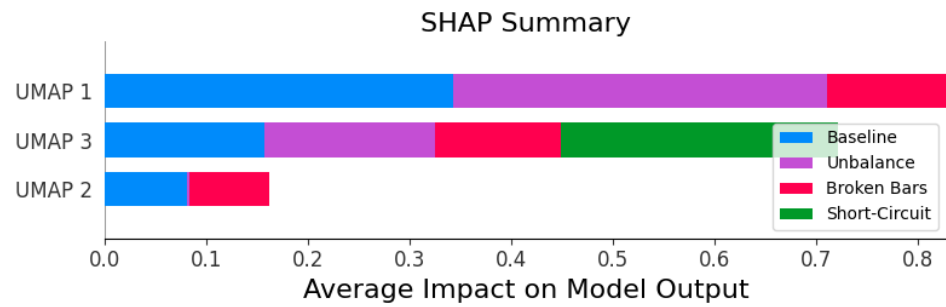


Figure 55 – Gaussian Naive Bayes SHAP Values Summary - Koopman

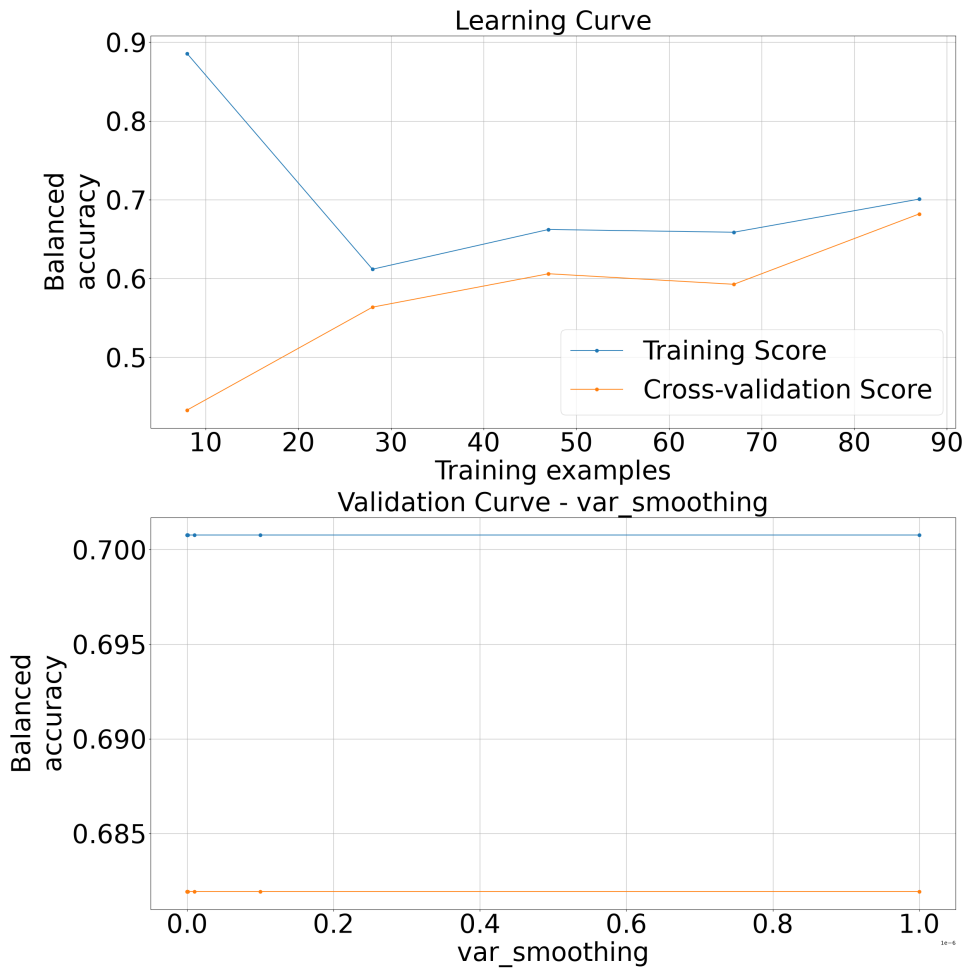


Figure 56 – Gaussian Naive Bayes Learning & Validation Curves - Koopman

The confusion matrix obtained using the raw current features (Figure 57) reveals a strong class bias in the learned model. In particular, the classifier struggled to correctly recognize the "Broken Bars" condition, with most of those samples being misidentified as "Unbalance". The SHAP analysis shown in Figure 58 indicates that the optimized model relied on a very small subset of the projected feature space: only two features were selected during hyperparameter tuning, yet both played a dominant role in the decision process. The associated learning and validation curves (Figure 59) suggest a mild tendency toward overfitting and illustrate the sensitivity of the model performance to the chosen hyperparameters.

When the FFT representation was used (Figure 60), the classifier exhibited a similar misclassification pattern. The "Broken Bars" class again proved difficult to separate, with most instances incorrectly assigned to the "Unbalance" category. The SHAP interpretation (Figure 61) confirms that the optimized configuration again converged to a minimal feature subset composed of two dominant components. In contrast to the raw-signal case, the learning and validation curves in Figure 62 show stable convergence with no clear evidence of overfitting, indicating a more regularized training behaviour.

The results obtained using the Koopman features present a different misclassification trend. As shown in Figure 63, the "Unbalance" class became the most challenging to identify, with many samples being predicted as "Baseline". The SHAP summary in Figure 64 reveals a more distributed feature importance profile compared with the previous representations. Although the second Koopman feature exerted the strongest influence, several additional features contributed meaningfully to the classification process, indicating a richer and more informative representation. The learning and validation curves in Figure 65 confirm a stable training process without signs of overfitting.

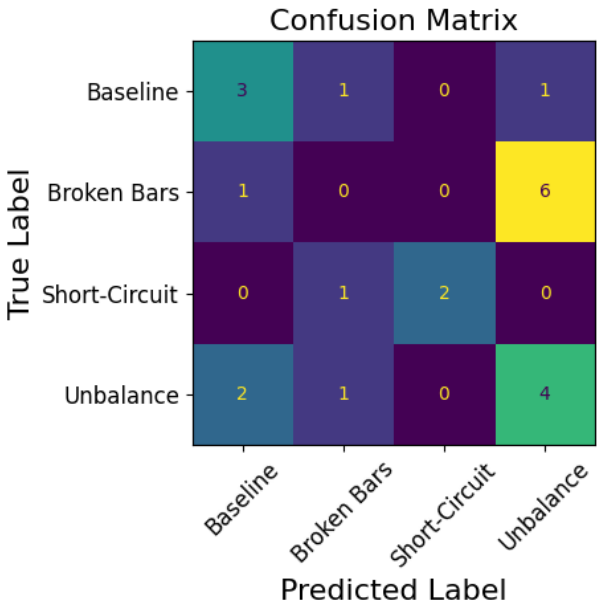


Figure 57 – XGBoost Confusion Matrix - Raw

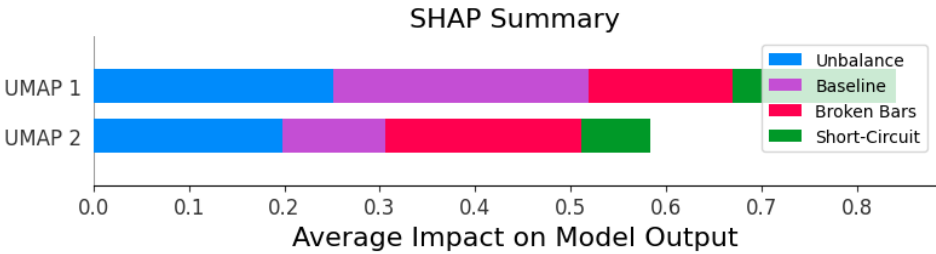


Figure 58 – XGBoost SHAP Values Summary - Raw

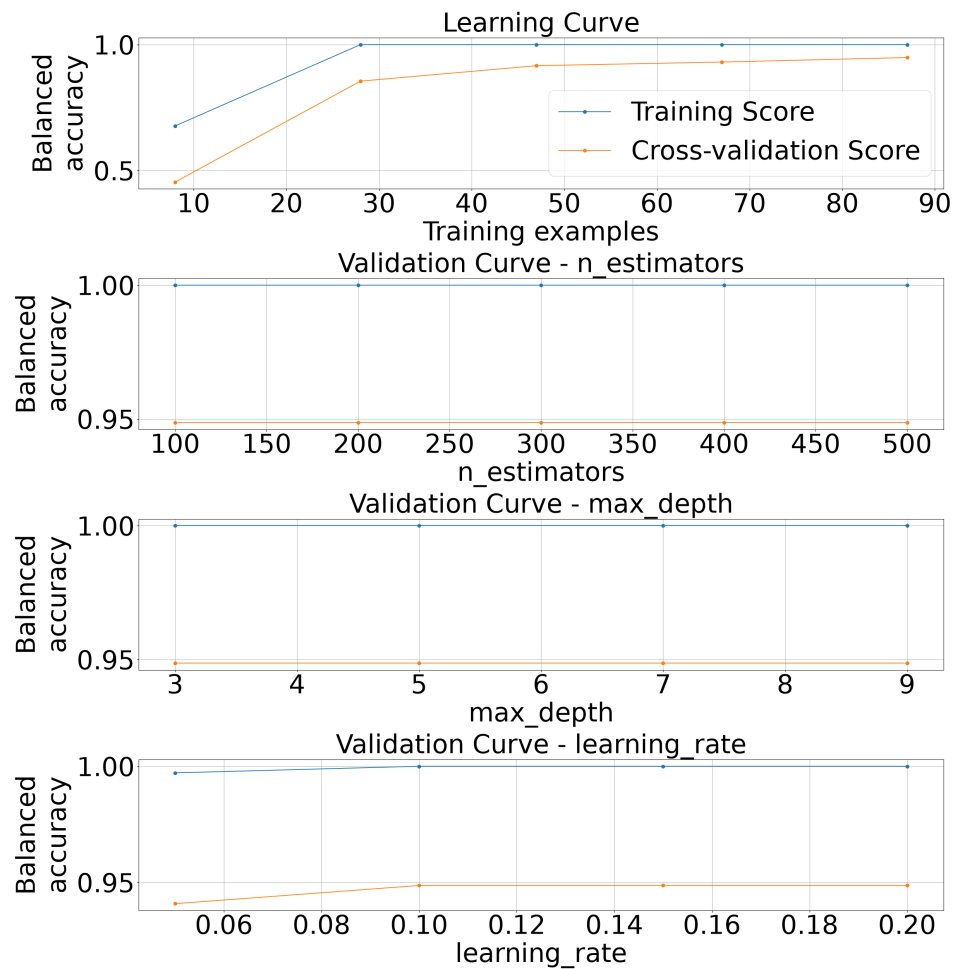


Figure 59 – XGBoost Learning & Validation Curves - Raw

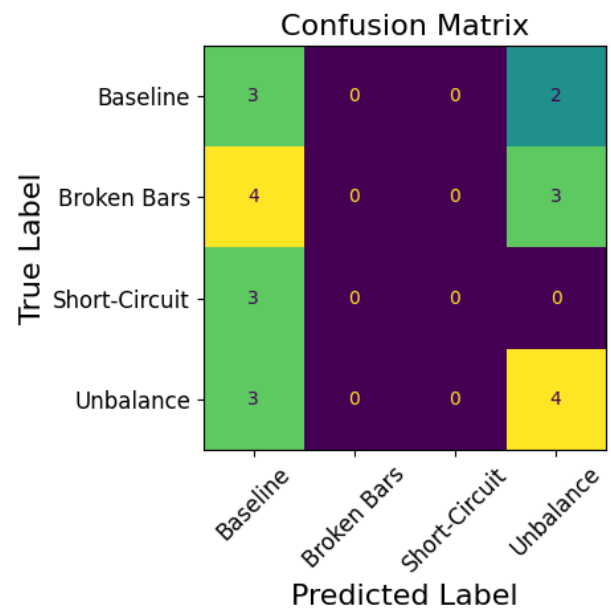


Figure 60 – XGBoost Confusion Matrix - FFT

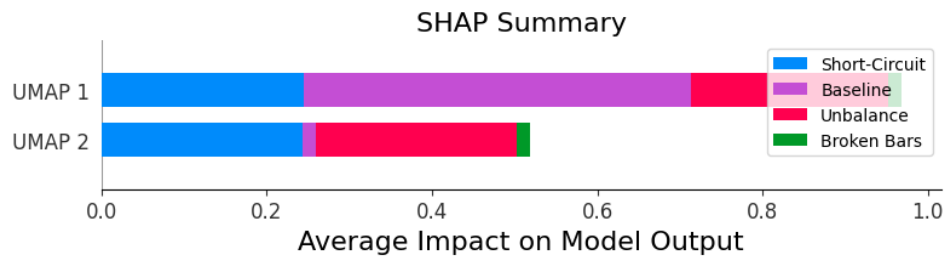


Figure 61 – XGBoost SHAP Values Summary - FFT

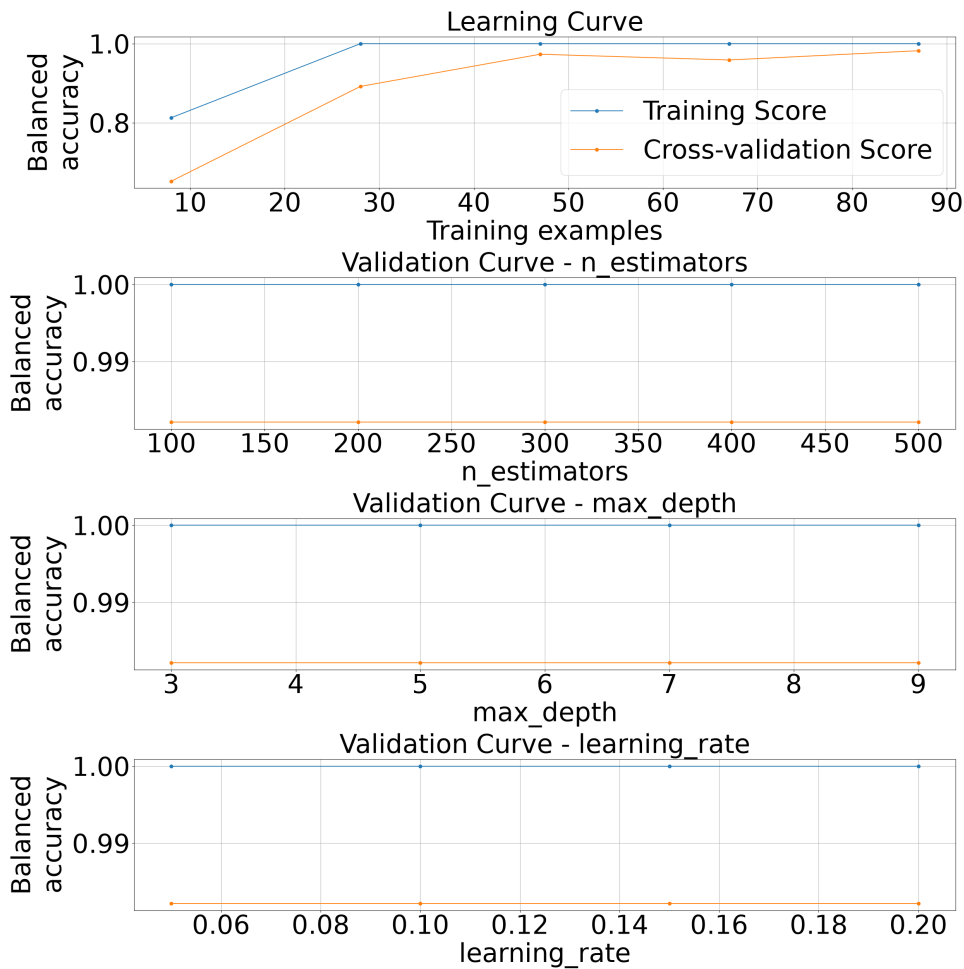


Figure 62 – XGBoost Learning & Validation Curves - FFT

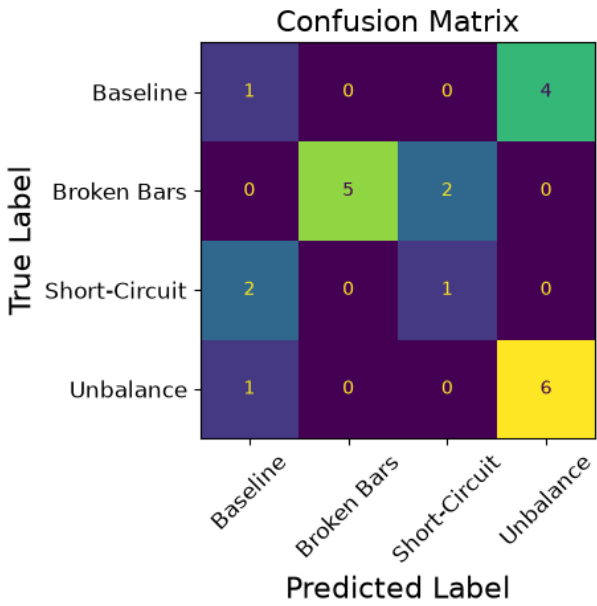


Figure 63 – XGBoost Confusion Matrix - Koopman

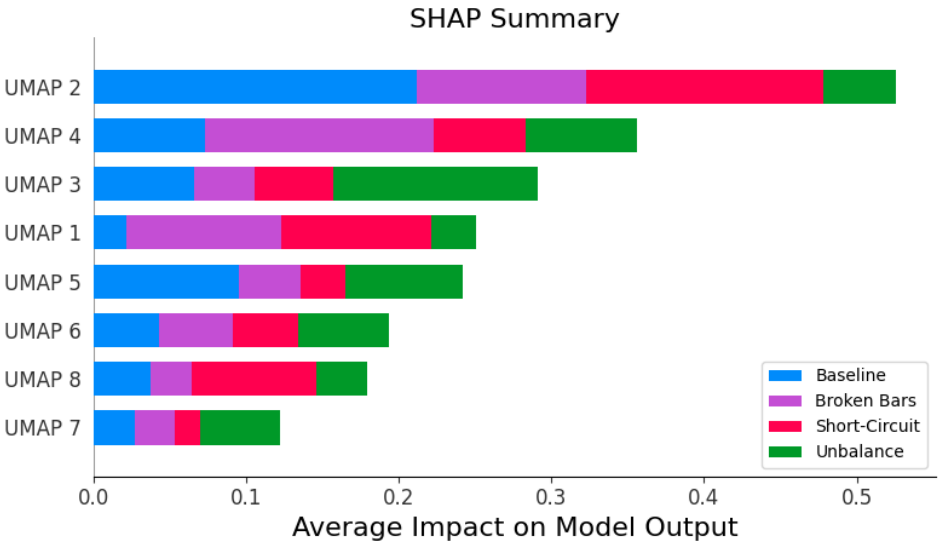


Figure 64 – XGBoost SHAP Values Summary - Koopman

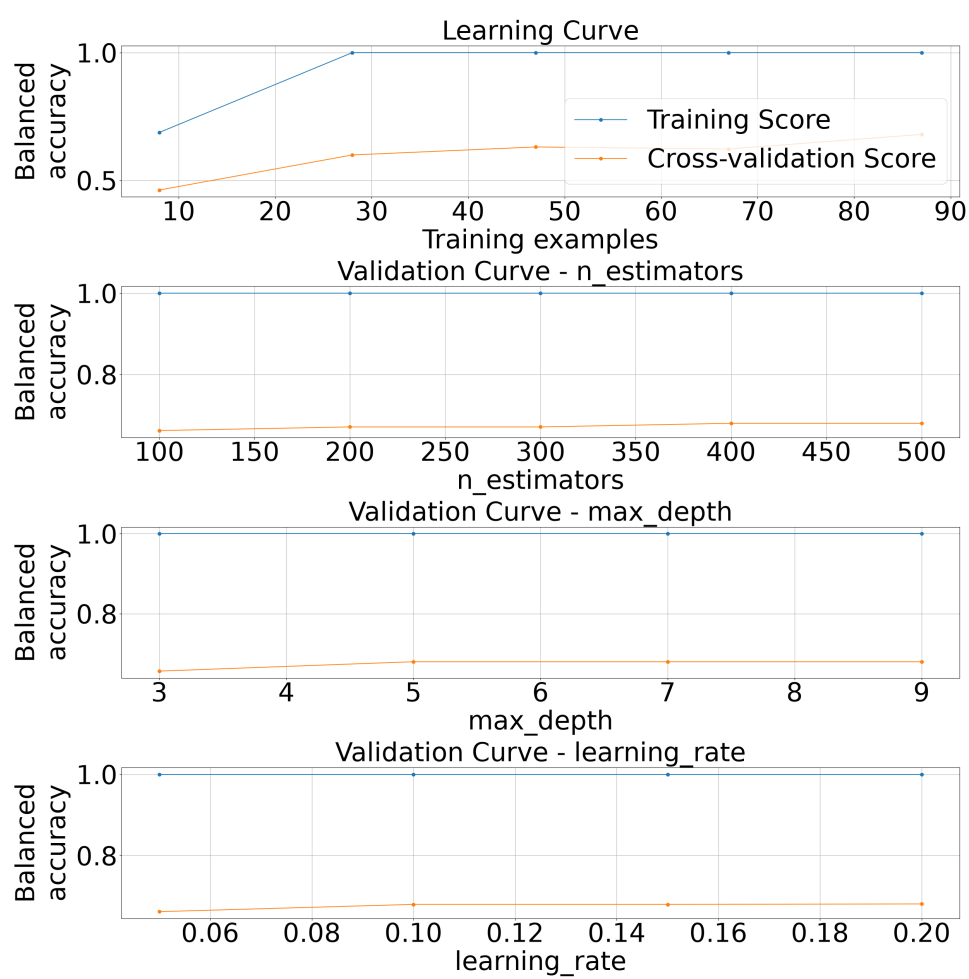


Figure 65 – XGBoost Learning & Validation Curves - Koopman

The confusion matrix obtained using the raw current signals (Figure 66) reveals that the MLP struggled to correctly recognize the "Unbalance" condition, with the majority of samples being incorrectly assigned to the "Baseline" class. The SHAP analysis in Figure 67 indicates that the final optimized model relied on only two latent features, both of which showed strong influence on the classification outcome. The corresponding learning and validation curves in Figure 68 suggest stable generalization behaviour, with no clear evidence of overfitting and a consistent response to hyperparameter tuning.

When Fourier-based features were employed (Figure 69), the classification performance deteriorated noticeably. The confusion matrix shows that most samples were misclassified as "Short-Circuit", indicating poor class separability in the Fourier feature space. The SHAP interpretation in Figure 70 demonstrates that the fourth and third latent features dominated the model decisions, followed by the second and fifth, while the first and sixth features contributed less but still provided relevant information. The learning and validation curves in Figure 71 reveal a pronounced gap between training and validation performance, providing strong evidence of overfitting when using FFT-derived inputs.

In contrast, the Koopman feature representation led to a more balanced learning behaviour (Figure 72). The confusion matrix shows improved discrimination across classes, although the "Broken Bars" condition remained challenging, with many samples being predicted as "Unbalance". The SHAP analysis in Figure 73 highlights the sixth and fourth latent features as the most influential, while the first and second features contributed more modestly to the decision process. The learning and validation curves in Figure 74 indicate stable convergence without clear signs of overfitting, suggesting that the Koopman representation provides a more robust and generalizable feature space for fault classification.

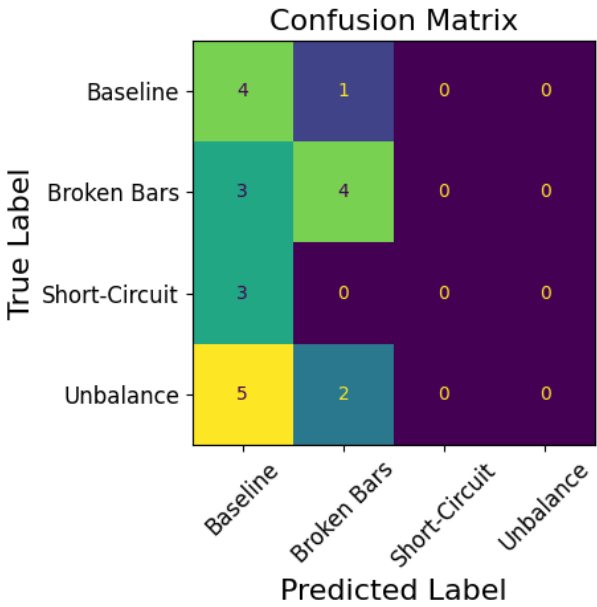


Figure 66 – MLP Confusion Matrix - Raw

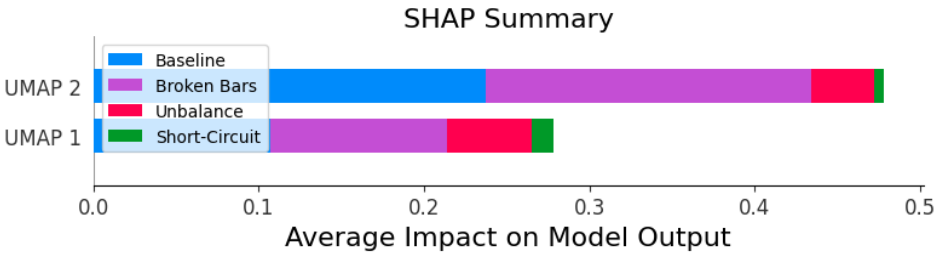


Figure 67 – MLP SHAP Values Summary - Raw

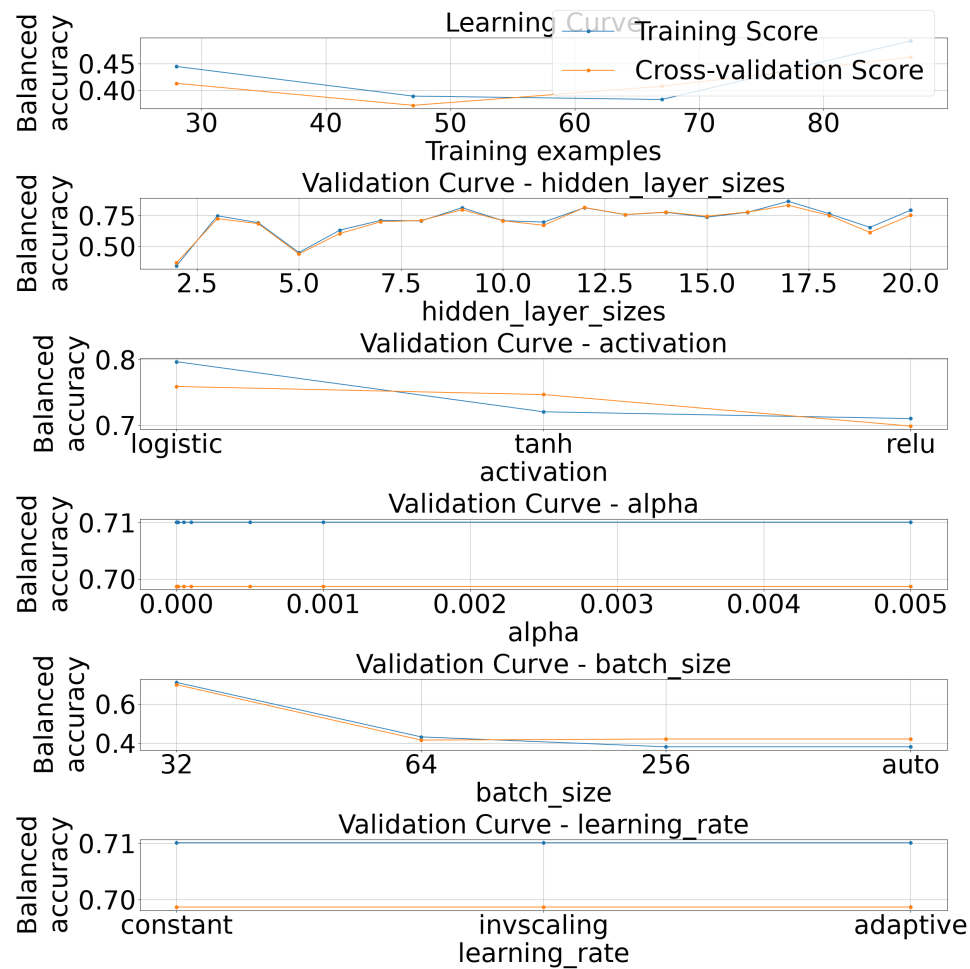


Figure 68 – MLP Learning & Validation Curves - Raw

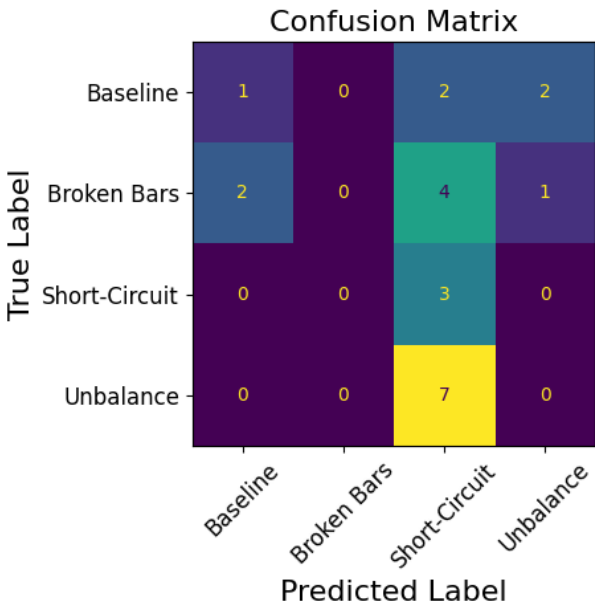


Figure 69 – MLP Confusion Matrix - FFT

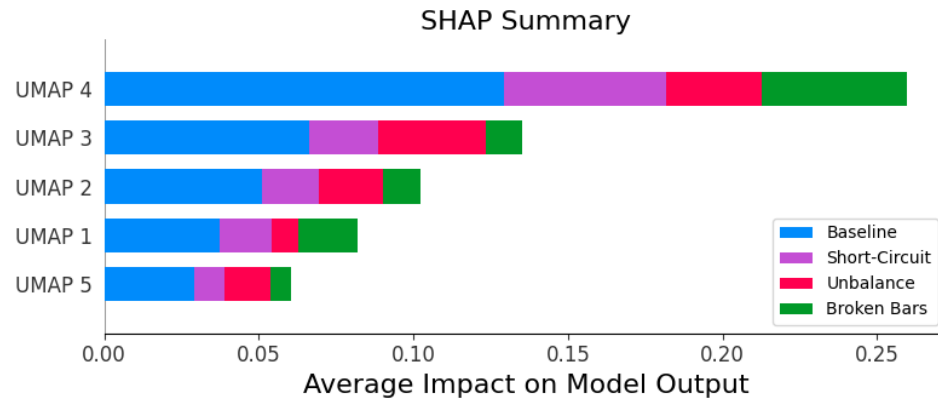


Figure 70 – MLP SHAP Values Summary - FFT

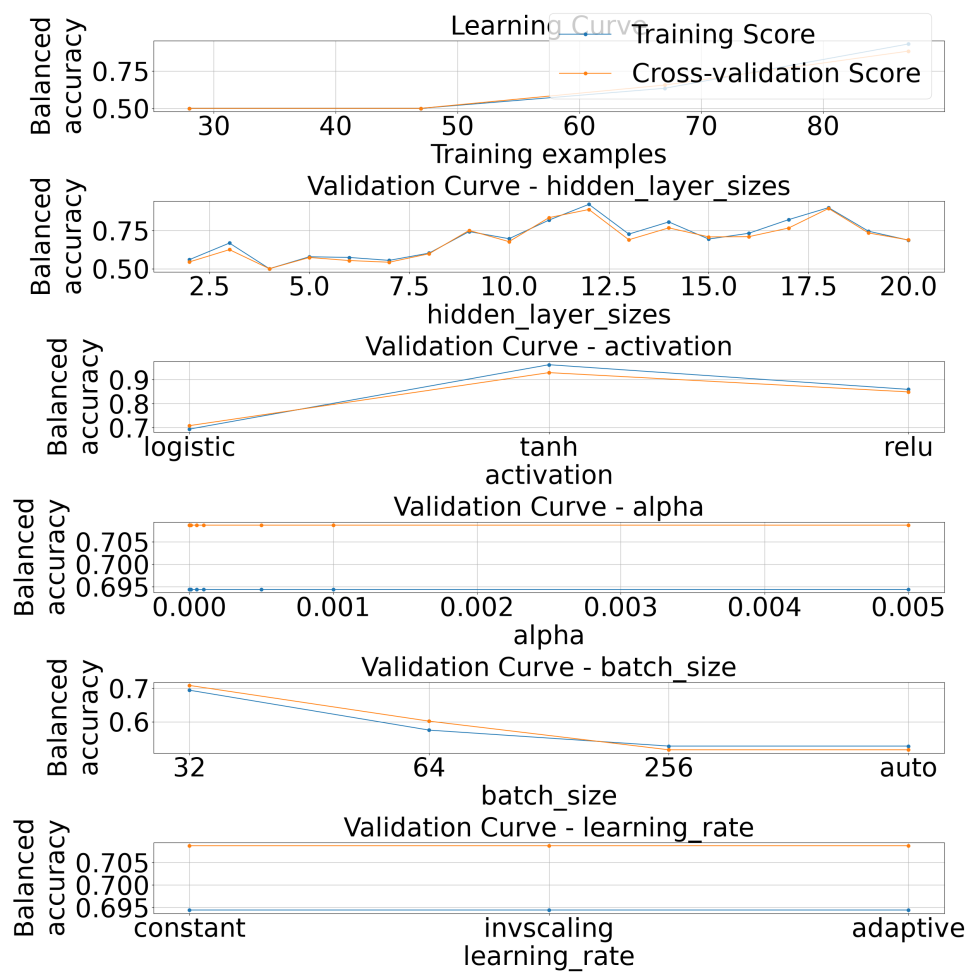


Figure 71 – MLP Learning & Validation Curves - FFT

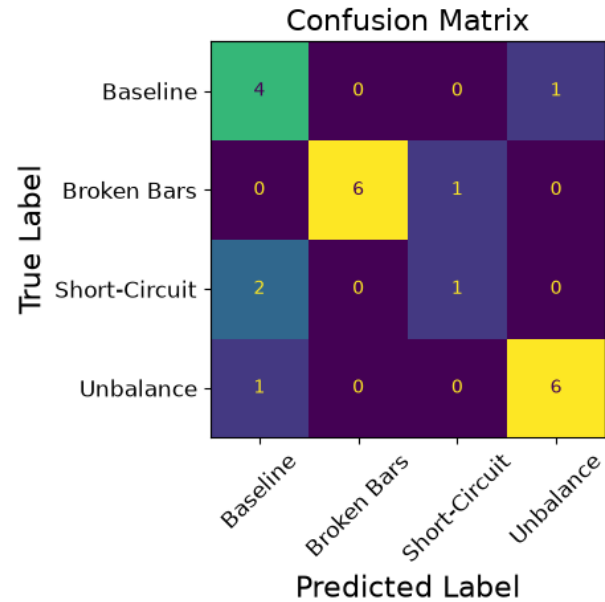


Figure 72 – MLP Confusion Matrix - Koopman

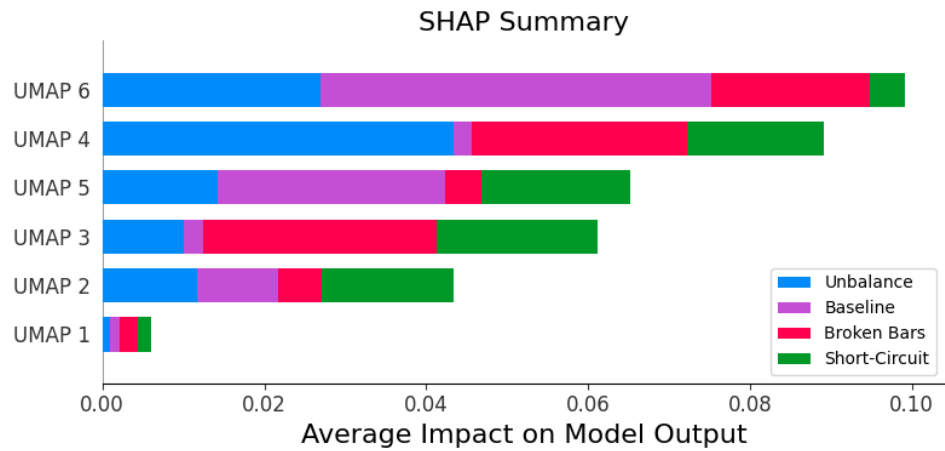


Figure 73 – MLP SHAP Values Summary - Koopman

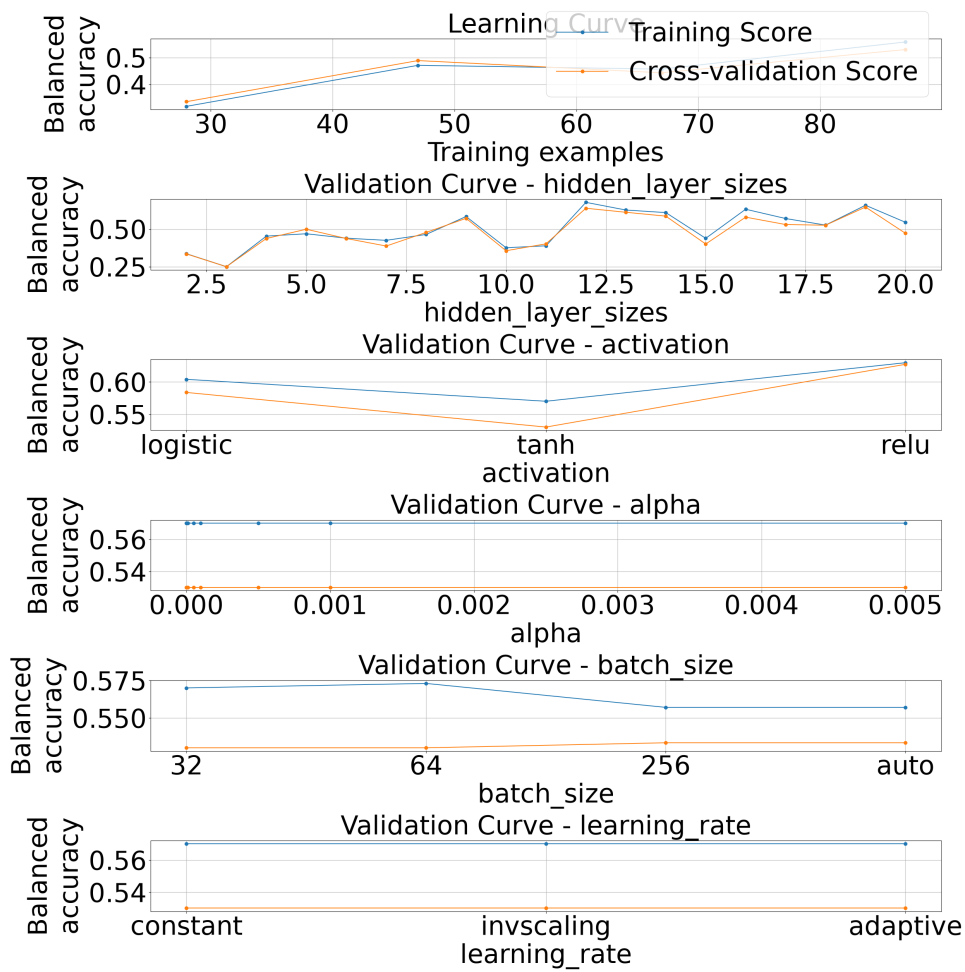


Figure 74 – MLP Learning & Validation Curves - Koopman

4.1.3 Fault Detection with Screws Labels

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	48%	57.5%	49.2%	29.9%	34.3%	56%	29.6%
Test Accuracy	11.9%	8.6%	17.7%	2.9%	19.3%	23.9%	25.3%
F1 Score	0.19	0.11	0.23	0.06	0.15	0.28	0.22
ROC-AUC	0.43	0.52	0.50	0.39	0.32	0.48	0.53

Table 7 – Raw Signal Input - Screws

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	79.6%	43.2%	75.2%	31.5%	28.3%	42.4%	88.4%
Test Accuracy	30.2%	36.9%	40.8%	16.9%	15.2%	4%	40.7%
F1 Score	0.35	0.39	0.46	0.24	0.23	0.04	0.27
ROC-AUC	0.61	0.36	0.66	0.41	0.47	0.46	0.64

Table 8 – Fourier Signal Input - Screws

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	91.2%	94.8%	91.6%	91.7%	72.9%	87.9%	86.8%
Test Accuracy	70.8%	77.4%	77.4%	69.6%	70.6%	80.5%	77.4%
F1 Score	0.67	0.75	0.75	0.64	0.60	0.78	0.77
ROC-AUC	0.85	0.84	0.86	0.86	0.83	0.86	0.82

Table 9 – Koopman Signal Input - Screws

Table 7 to Table 9 summarize the overall performance of all tested supervised algorithms for raw, Fourier Spectrum, and Koopman selected features as inputs for motor fault detection with the "1 Screw" and "2 Screws" labels, showing that, even with high cross-validation balanced accuracy, both raw and Fourier's test balanced accuracy are quite low, indicating a model overfit and an inability to extrapolate to untrained fault data. The Koopman features had more promising results, with XGBoost as the best performing algorithm followed by SVM, Random Forest and MLP all of them had cross-validation accuracy above 85% and test accuracy near or higher than 70%. The worst performing model was the Logistic Regression algorithm due to its linearity assumption being broken, lowering its performance. However, the UMAP + SMOTE preprocessing step partially mitigates this limitation. By enforcing local metric linearity and introducing neighbourhood-based interpolations, the resulting feature space becomes more compatible with linear decision boundaries, allowing this algorithm to achieve relatively competitive accuracy despite its theoretical limitation.

The confusion matrix obtained using the raw current signal (Figure 75) reveals that the k-NN classifier struggled to consistently separate the fault categories. Although all classes were represented in the predictions, the "Broken Bars" condition achieved the highest recognition rate, with three correct identifications and four misclassifications. The SHAP analysis in Figure 76 indicates that the optimization process selected only two projected dimensions, both of which played a dominant role in the decision process. The associated learning and validation curves in Figure 77 show stable convergence behaviour and no clear evidence of overfitting, while also illustrating the sensitivity of performance to the selected hyperparameters.

When the FFT-based representation was used (Figure 78), the classifier again exhibited limited discriminative capability across the full set of classes. In this case, the "1 Screw" condition was the most distinguishable, achieving three correct and three incorrect predictions. The SHAP interpretation in Figure 79 suggests a more distributed contribution among the spectral features, with the fifth component having the strongest influence, followed by the third and sixth components. The remaining features still contributed to the classification, but with smaller impact. The learning and validation curves in Figure 80 confirm a similar generalization behaviour to the raw-signal experiment.

A noticeable change occurs when the Koopman-based features are employed (Figure 81). The confusion matrix shows improved class separation overall, although the "Broken Bars" condition remained challenging and was frequently confused with the "2 Screws" scenario. The SHAP analysis in Figure 82 highlights a different importance pattern, with the eighth feature emerging as the most influential, followed by the fifth and third components. The seventh feature exhibited the smallest contribution but still participated in the classification process. Finally, the learning and validation curves in Figure 83 indicate stable training behaviour without signs of overfitting, supporting the robustness of the optimized k-NN configuration.

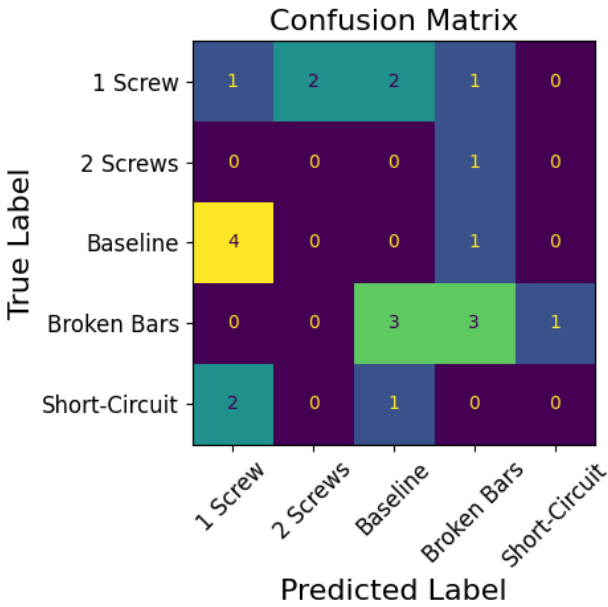


Figure 75 – k-NN Confusion Matrix - Raw

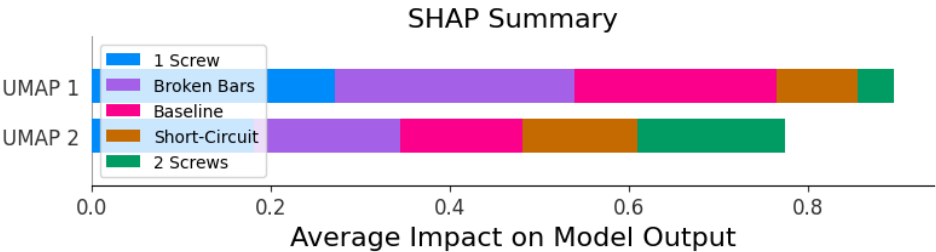


Figure 76 – k-NN SHAP Values Summary - Raw

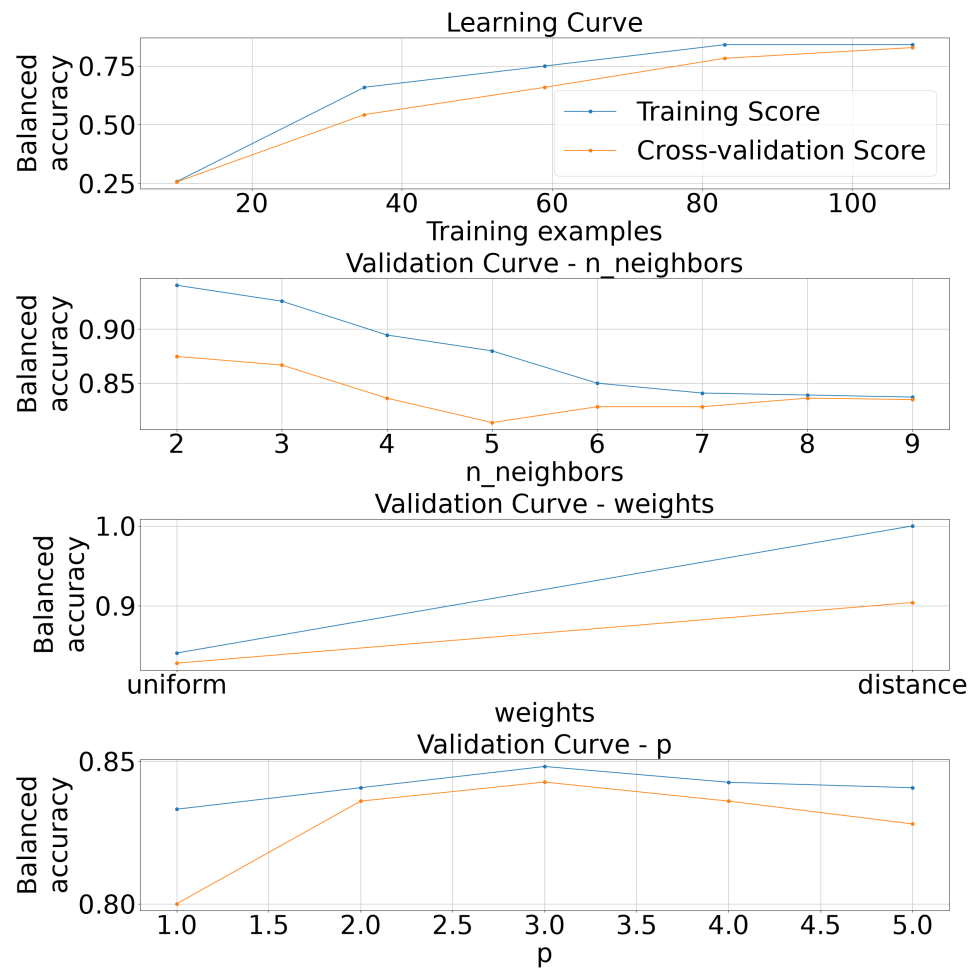


Figure 77 – k-NN Learning & Validation Curves - Raw

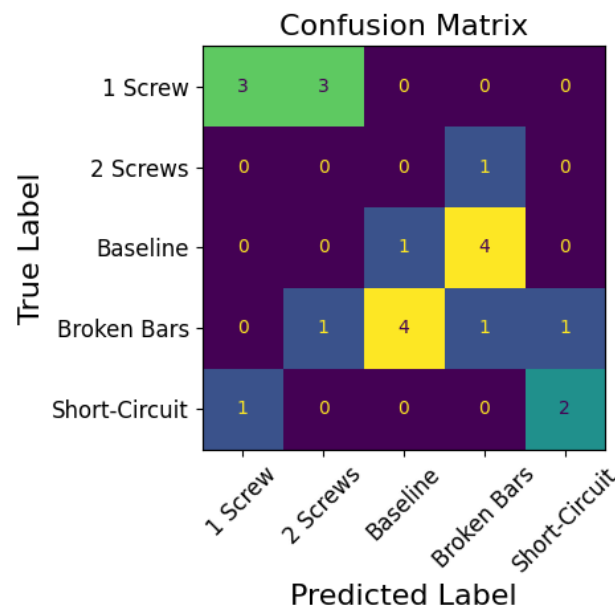


Figure 78 – k-NN Confusion Matrix - FFT

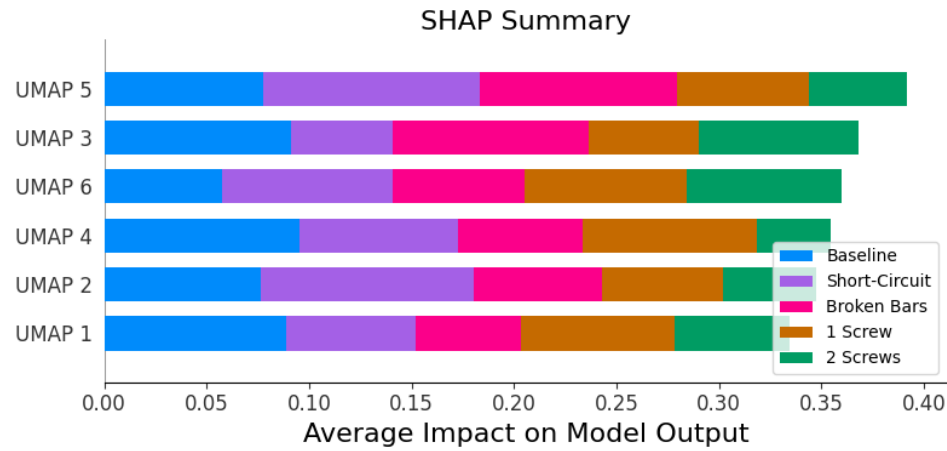


Figure 79 – k-NN SHAP Values Summary - FFT

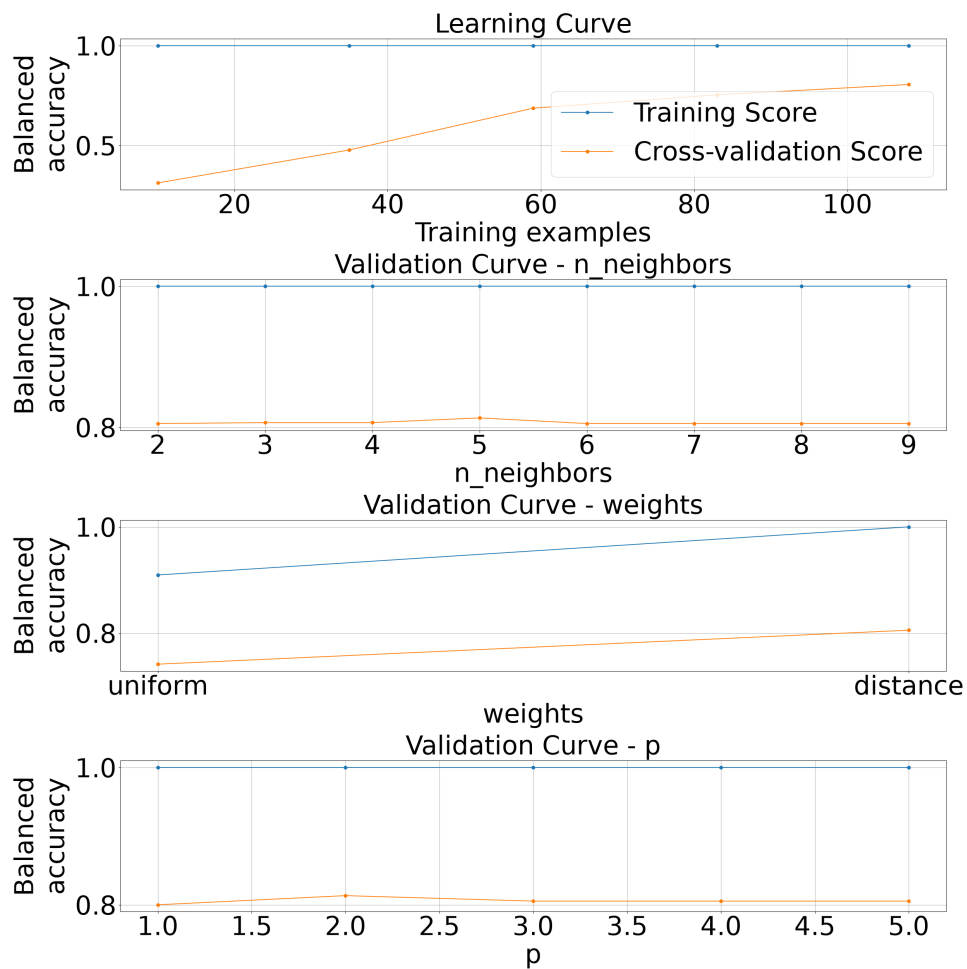


Figure 80 – k-NN Learning & Validation Curves - FFT

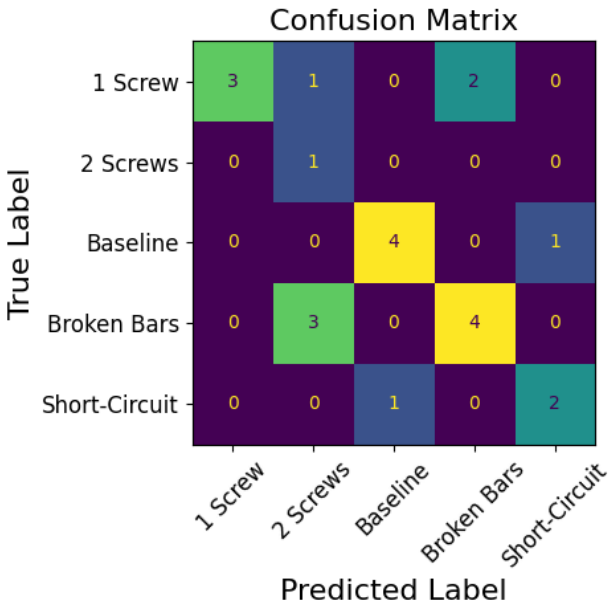


Figure 81 – k-NN Confusion Matrix - Koopman

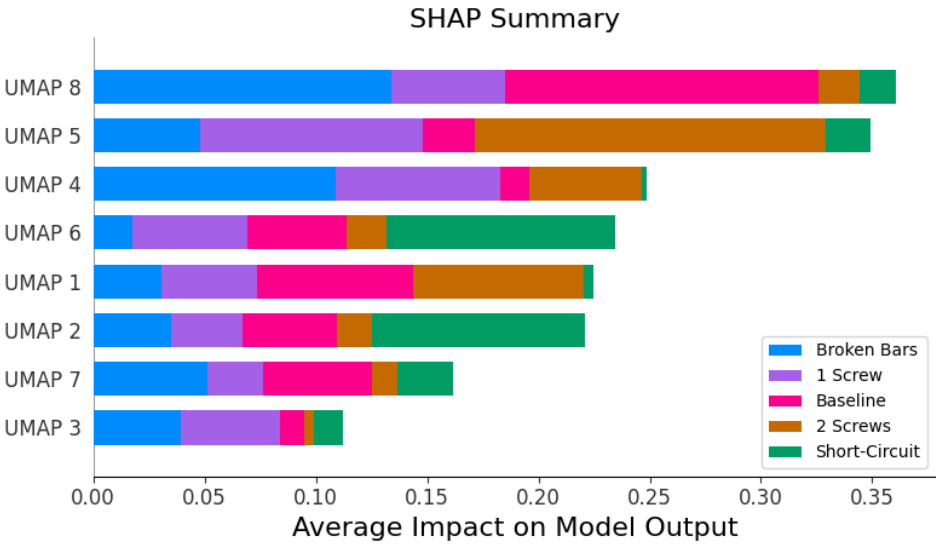


Figure 82 – k-NN SHAP Values Summary - Koopman

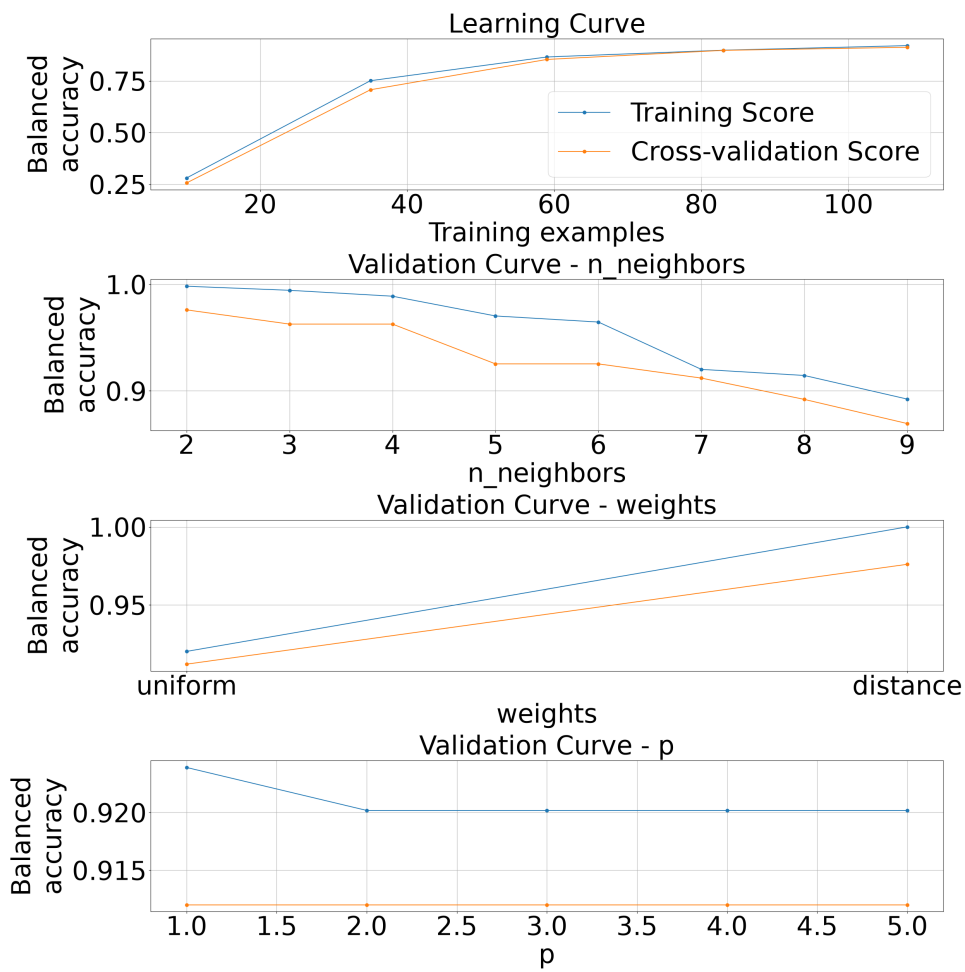


Figure 83 – k-NN Learning & Validation Curves - Koopman

Figure 84 illustrates the class-wise performance obtained when the SVM is trained directly on the raw signal representation. The confusion matrix reveals clear difficulty in separating the fault categories, with only limited successful predictions across classes; the "Broken Bars" condition shows the best performance, with three correct predictions out of seven test samples. The SHAP analysis in Figure 85 indicates that the model ultimately relied on only two projected features, both of which played a dominant role in the decision process. The learning and validation curves shown in Figure 86 suggest a noticeable gap between training and validation performance, highlighting signs of overfitting and demonstrating the sensitivity of the model to the selected hyperparameters.

When the FFT representation is used as input, the confusion matrix in Figure 87 shows that the classifier still struggles to consistently discriminate between the fault categories. The "Short-Circuit" condition emerges as the most accurately recognized class, with two correct classifications out of three test samples. The feature attribution analysis in Figure 88 reveals a more distributed contribution among the projected variables, with the fifth feature having the largest influence, followed by the fourth, sixth, and third features, while the first and second features contribute less strongly but remain relevant. The corresponding learning and validation curves in Figure 89 provide stronger evidence of overfitting compared to the raw-signal case, again emphasizing the impact of hyperparameter tuning on the generalization capability.

The results obtained using the Koopman-based feature representation are summarized in Figure 90. In this case, the classifier demonstrates improved behaviour overall, although the "Broken Bars" condition remains challenging and is occasionally confused with the "1 Screw" and "2 Screws" classes. The SHAP interpretation in Figure 91 shows that the third feature provides the strongest contribution to the decision process, followed by the first and second features, then the fifth, while the fourth feature plays a comparatively smaller role. The learning and validation curves in Figure 92 still indicate a certain level of overfitting, but the training dynamics suggest a more stable and informative feature space relative to the raw and FFT representations.

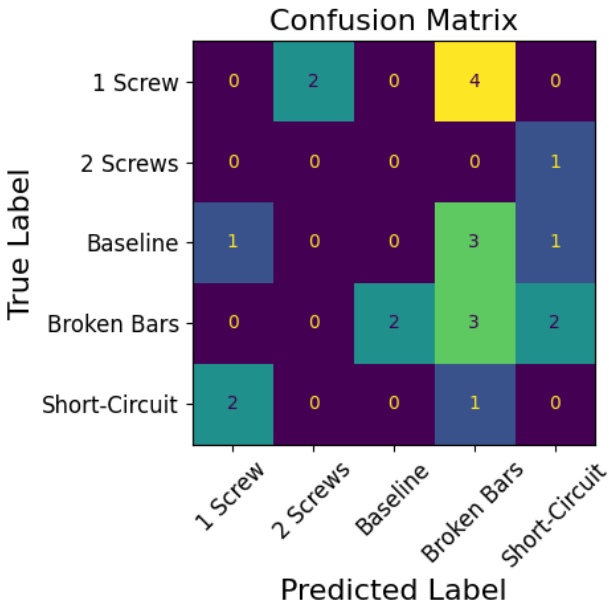


Figure 84 – SVM Confusion Matrix - Raw

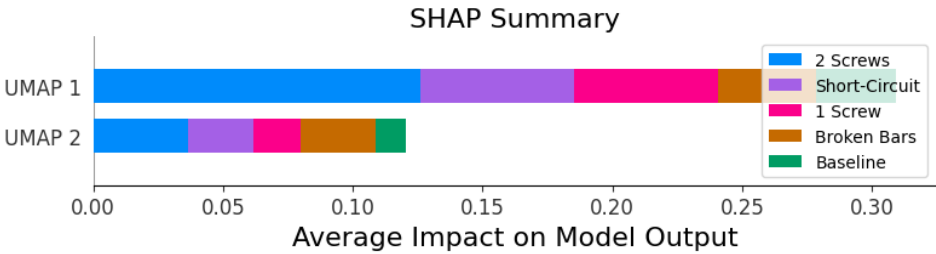


Figure 85 – SVM SHAP Values Summary - Raw

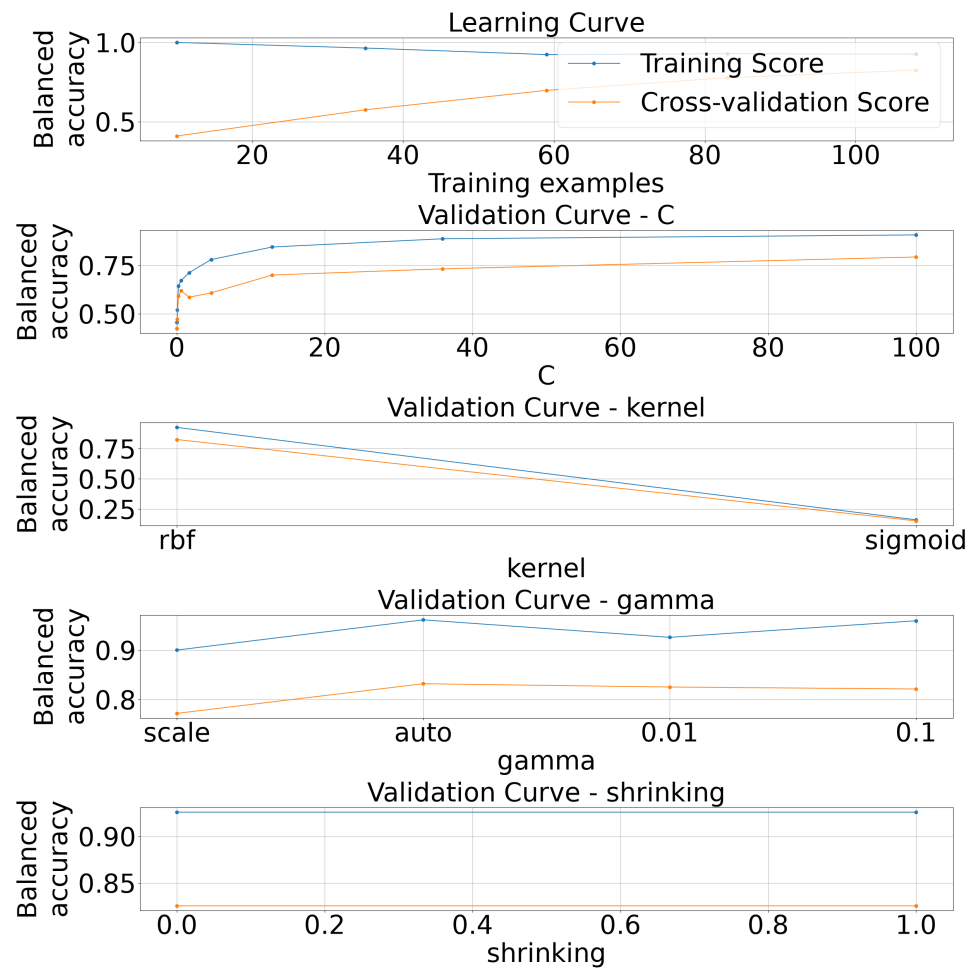


Figure 86 – SVM Learning & Validation Curves - Raw

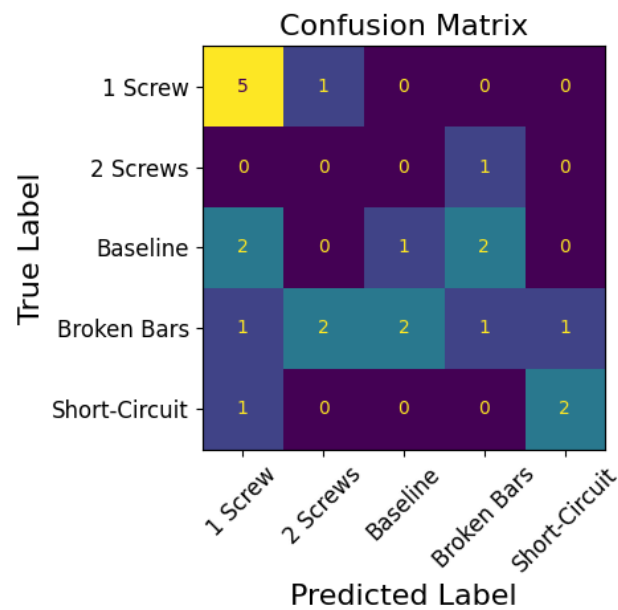


Figure 87 – SVM Confusion Matrix - FFT

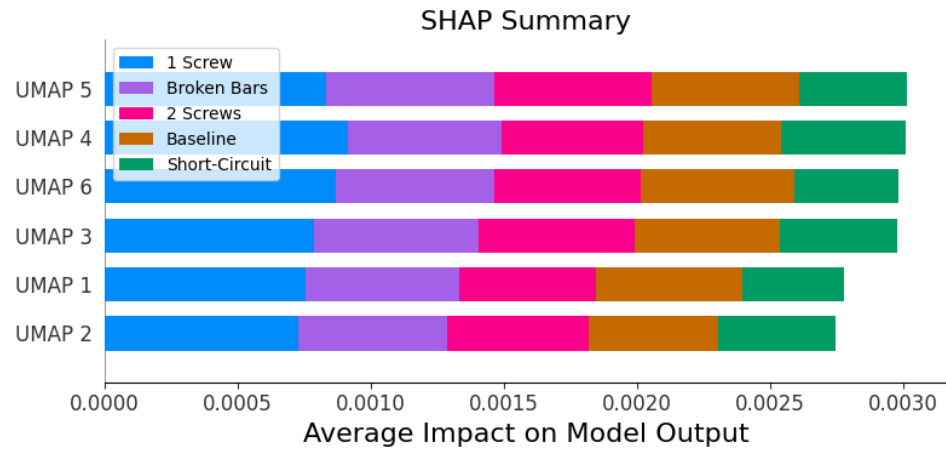


Figure 88 – SVM SHAP Values Summary - FFT

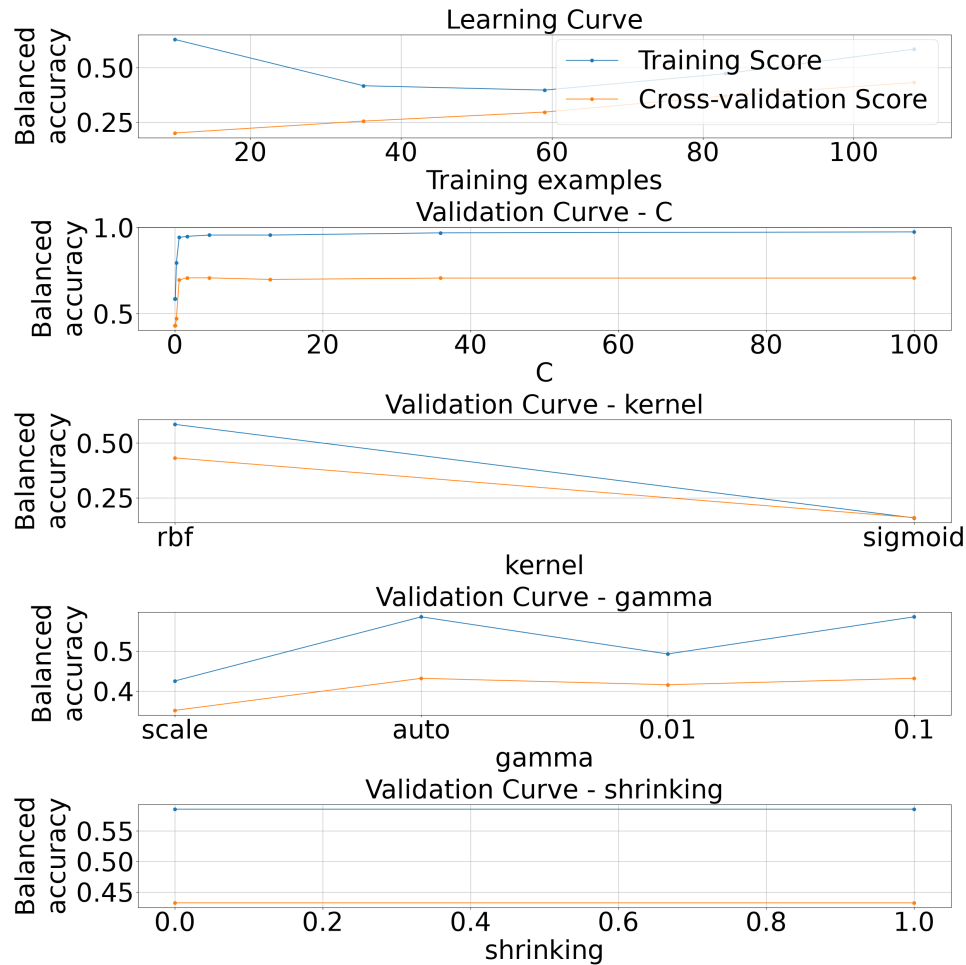


Figure 89 – SVM Learning & Validation Curves - FFT

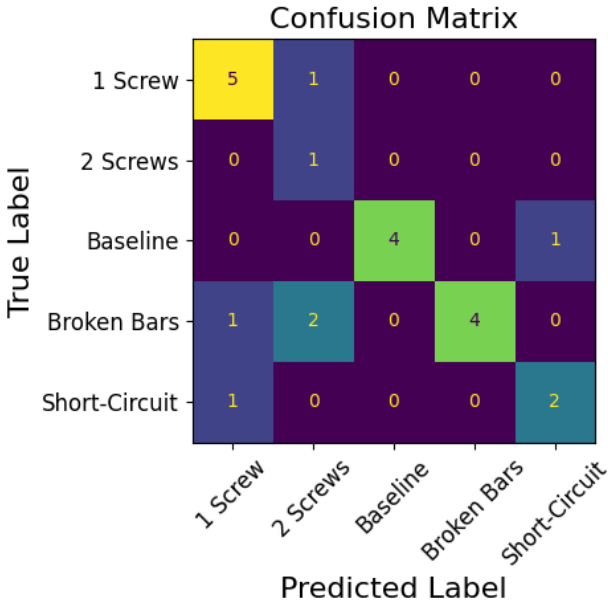


Figure 90 – SVM Confusion Matrix - Koopman

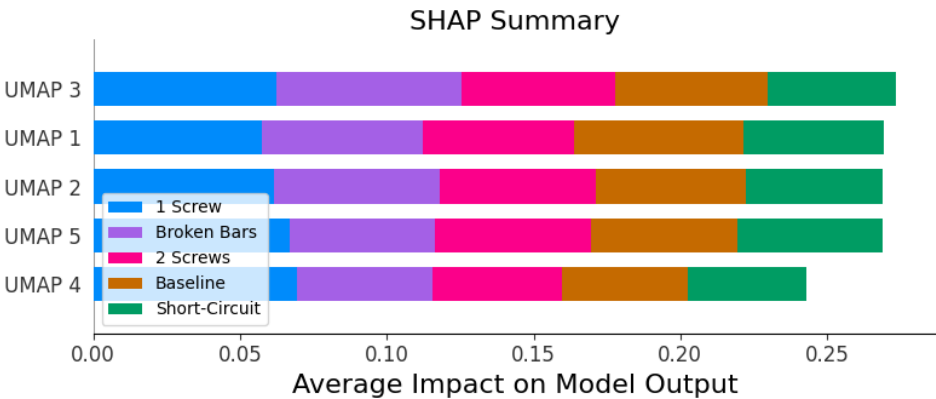


Figure 91 – SVM SHAP Values Summary - Koopman

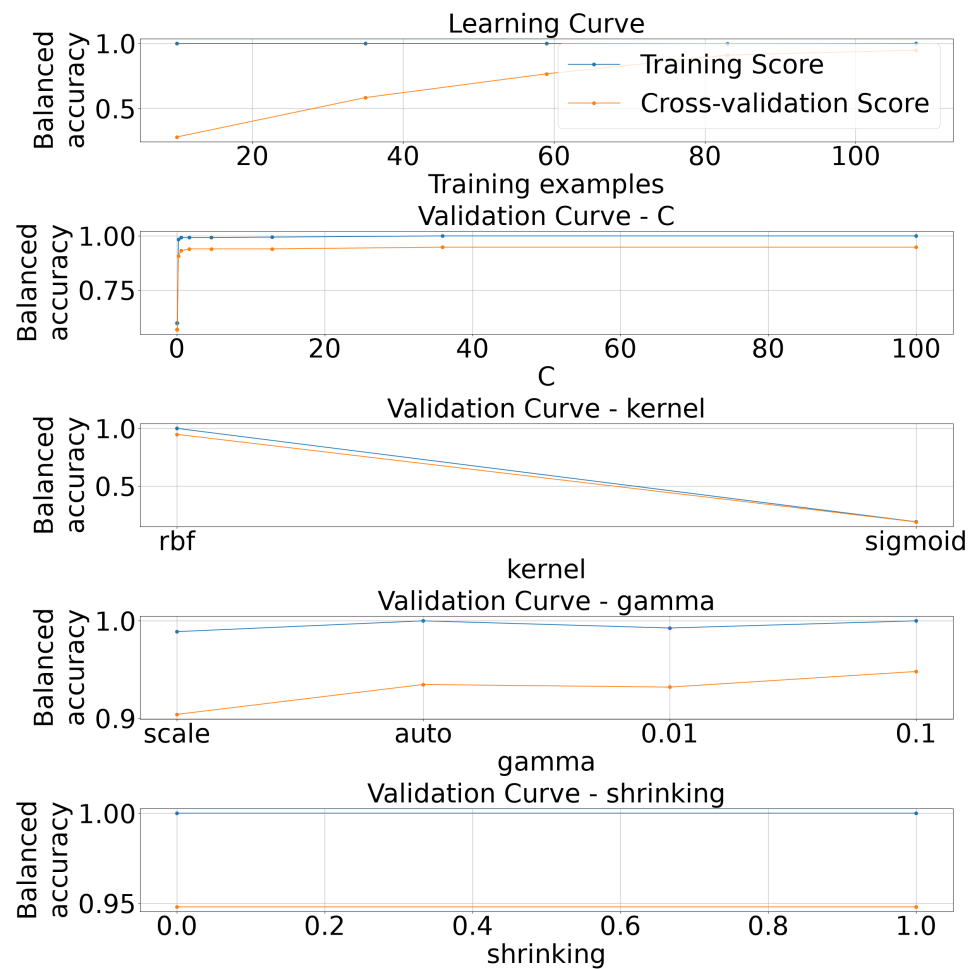


Figure 92 – SVM Learning & Validation Curves - Koopman

Figure 93 illustrates the performance of the Random Forest classifier when trained directly on the raw current signals. The confusion matrix shows that the model struggled to capture most fault categories, achieving consistent recognition only for the "Baseline" condition, where three of the five test samples were correctly classified. The feature-importance analysis in Figure 94 indicates that the hyperparameter optimization process retained only two projected features, both of which played a dominant role in the decision process. The learning and validation curves in Figure 95 reveal noticeable divergence between training and validation performance, suggesting the presence of overfitting and a strong sensitivity to hyperparameter selection.

When the same classifier was trained using Fourier-domain features, the behaviour changed but important limitations remained. The confusion matrix in Figure 96 shows persistent confusion between the "2 Screws" and "Broken Bars" conditions, indicating that the spectral representation alone was insufficient to clearly separate these faults. The SHAP analysis in Figure 97 highlights that the third feature provided the largest contribution to the classification, followed by the second and fourth features, while the first and fifth still provided non-negligible support. However, the learning curves in Figure 98 show a pronounced gap between training and validation scores, reinforcing the evidence of overfitting in the Fourier-based representation.

A different pattern emerges when the Koopman-derived features are used. The confusion matrix in Figure 99 shows a clear improvement in the identification of the "Broken Bars" condition, with four correct predictions out of seven test samples. The feature-importance results in Figure 100 indicate that the third feature again plays the most significant role, followed by the second and fourth, while the first feature contributes less strongly. In contrast to the previous representations, the learning and validation curves in Figure 101 display a much closer agreement, suggesting improved generalization and the absence of strong overfitting effects. Together, these results indicate that the Koopman-based representation provides a more stable and discriminative feature space for fault classification.

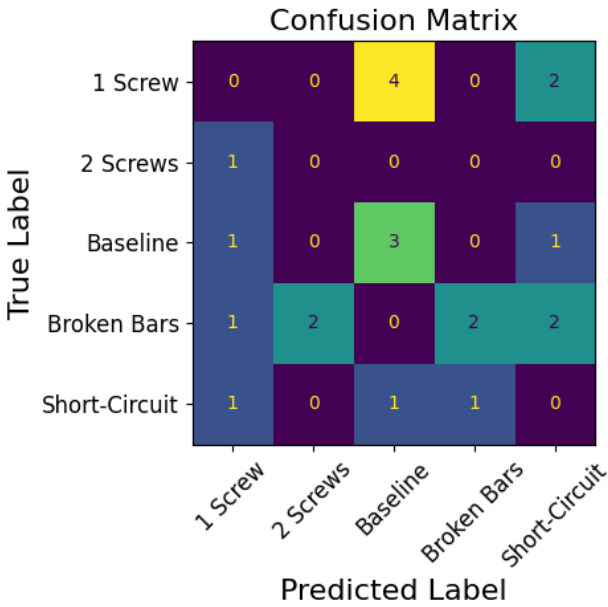


Figure 93 – Random Forest Confusion Matrix - Raw

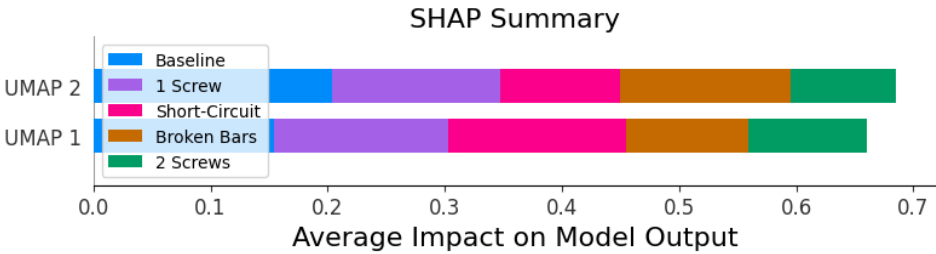


Figure 94 – Random Forest SHAP Values Summary - Raw

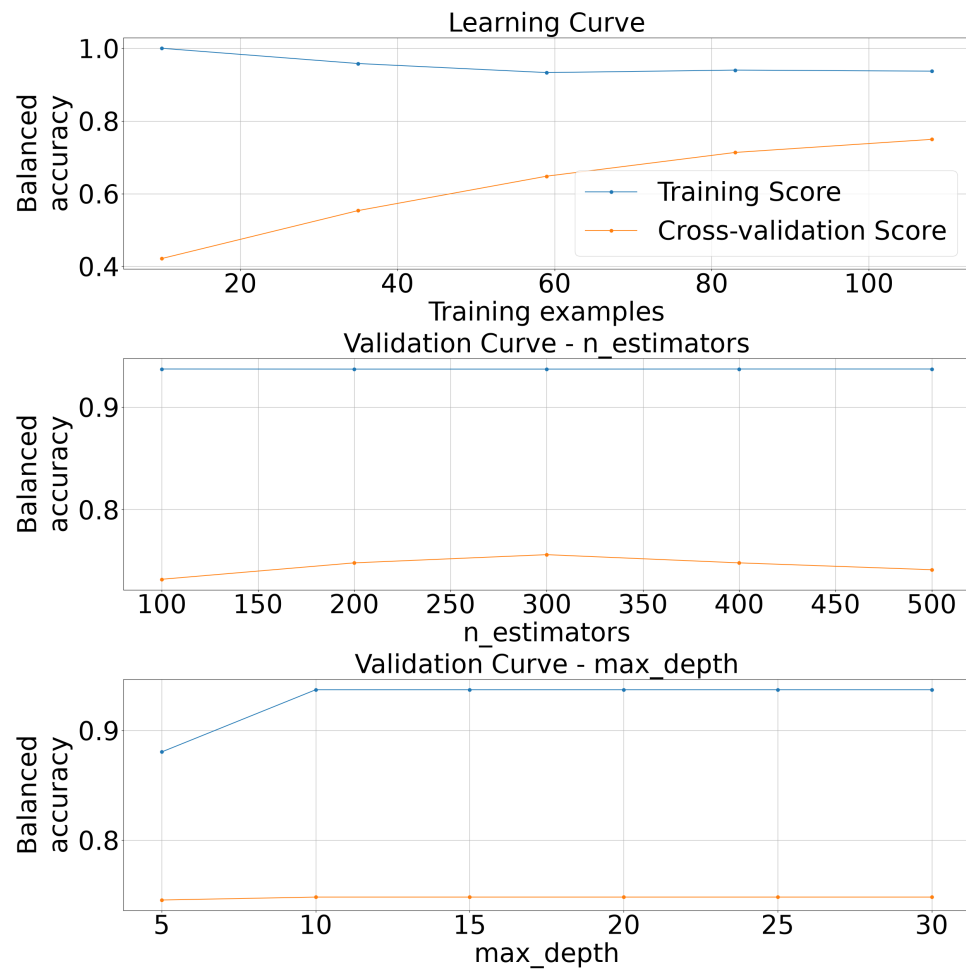


Figure 95 – Random Forest Learning & Validation Curves - Raw

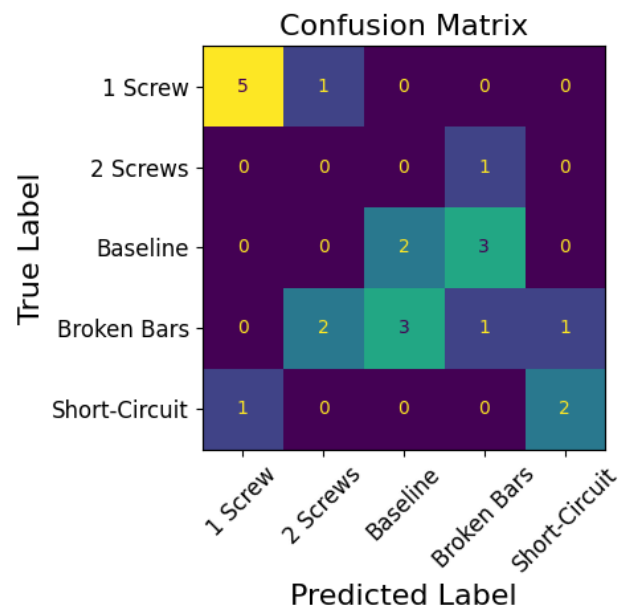


Figure 96 – Random Forest Confusion Matrix - FFT

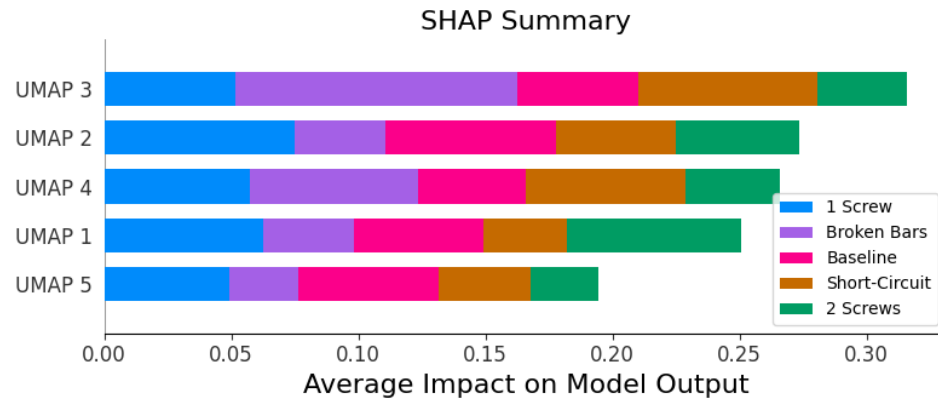


Figure 97 – Random Forest SHAP Values Summary - FFT

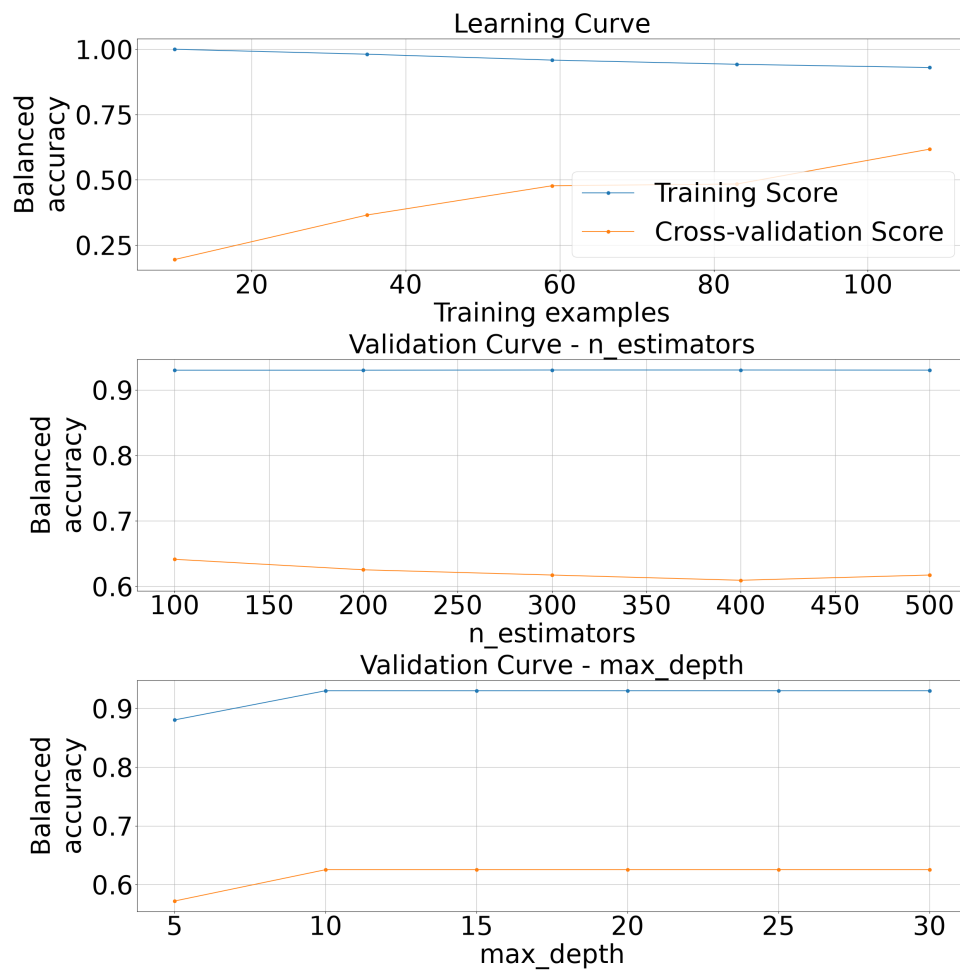


Figure 98 – Random Forest Learning & Validation Curves - FFT

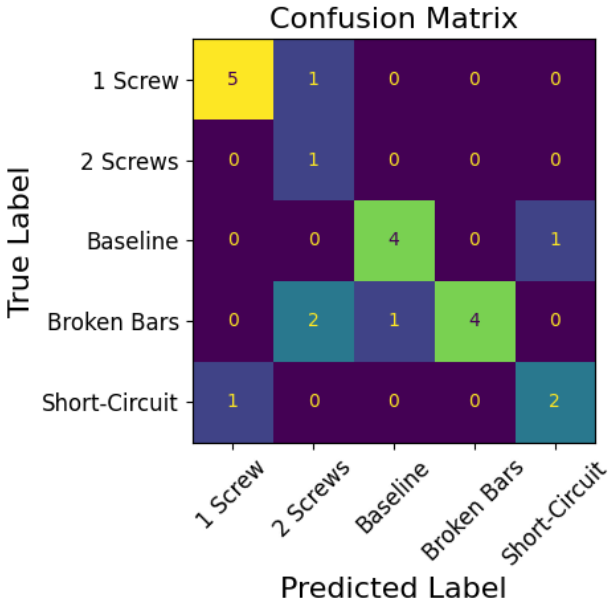


Figure 99 – Random Forest Confusion Matrix - Koopman

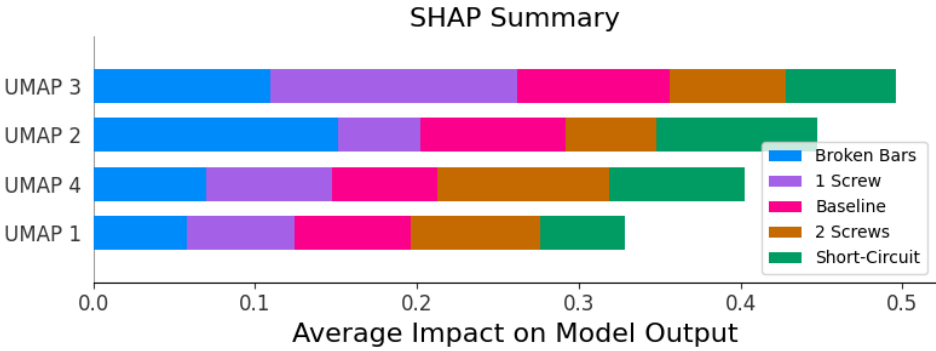


Figure 100 – Random Forest SHAP Values Summary - Koopman

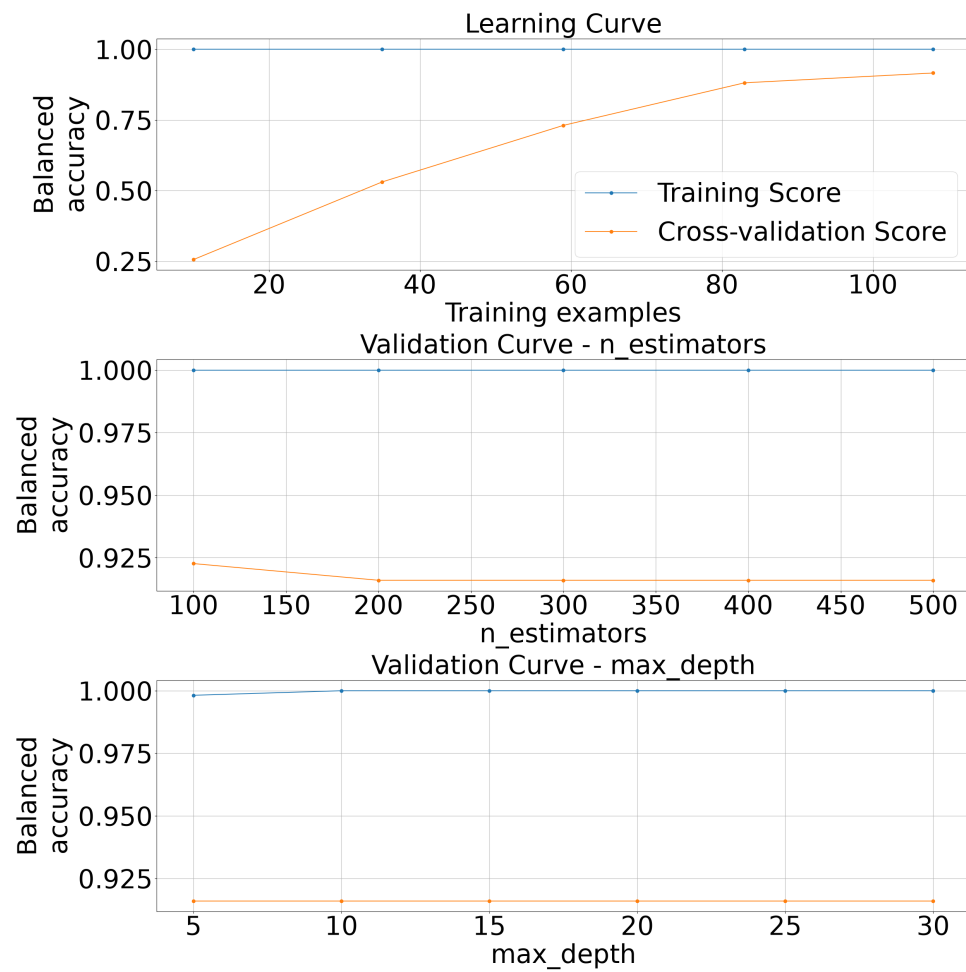


Figure 101 – Random Forest Learning & Validation Curves - Koopman

The confusion matrix obtained using the raw current signals ([Figure 102](#)) reveals that Logistic Regression struggled to build a consistent decision boundary for most classes. Only the "Broken Bars" condition was identified with limited success, with a single correct prediction among seven test samples. The SHAP analysis shown in [Figure 103](#) indicates that the optimization procedure retained only two projected features, both of which played a dominant role in the classification decisions. The learning and validation curves in [Figure 104](#) further suggest the presence of overfitting, as well as a noticeable sensitivity of model performance to hyperparameter selection.

When the classifier was trained using Fourier-domain features, the behaviour changed but remained suboptimal. As shown in [Figure 105](#), the model exhibited significant confusion between classes, particularly mislabelling most "Broken Bars" samples as the "2 Screws" condition. The feature attribution results in [Figure 106](#) highlight that the second projected feature contributed most strongly to the decision process, followed by the fourth, while the remaining components still provided non-negligible contributions. The learning and validation curves in [Figure 107](#) again reveal signs of overfitting and illustrate the impact of hyperparameter tuning on the classifier's ability to generalize.

A clear improvement is observed when Koopman-based features are employed. The confusion matrix in [Figure 108](#) shows that most classes were learned more reliably, although the "1 Screw" condition still presented some difficulty, with three correct predictions out of six test samples. The SHAP analysis in [Figure 109](#) indicates a more distributed contribution across multiple features, with the third component being the most influential, followed by the fifth and eighth, while the remaining features still played a supportive role in the decision process. Finally, the learning and validation curves in [Figure 110](#) suggest only mild overfitting and demonstrate a more stable relationship between hyperparameters and model performance compared to the previous feature representations.

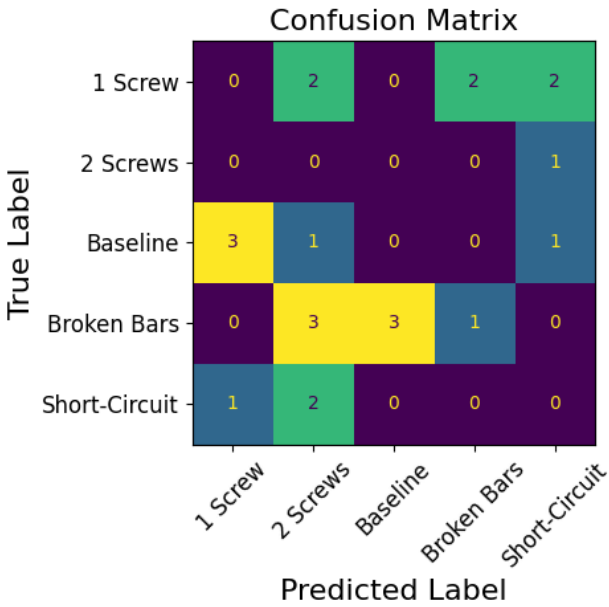


Figure 102 – Logistic Regression Confusion Matrix - Raw

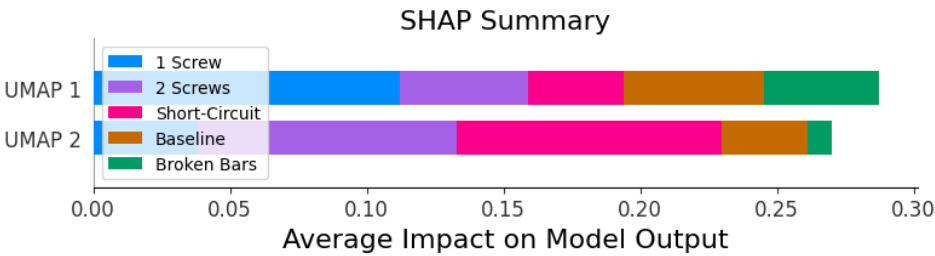


Figure 103 – Logistic Regression SHAP Values Summary - Raw

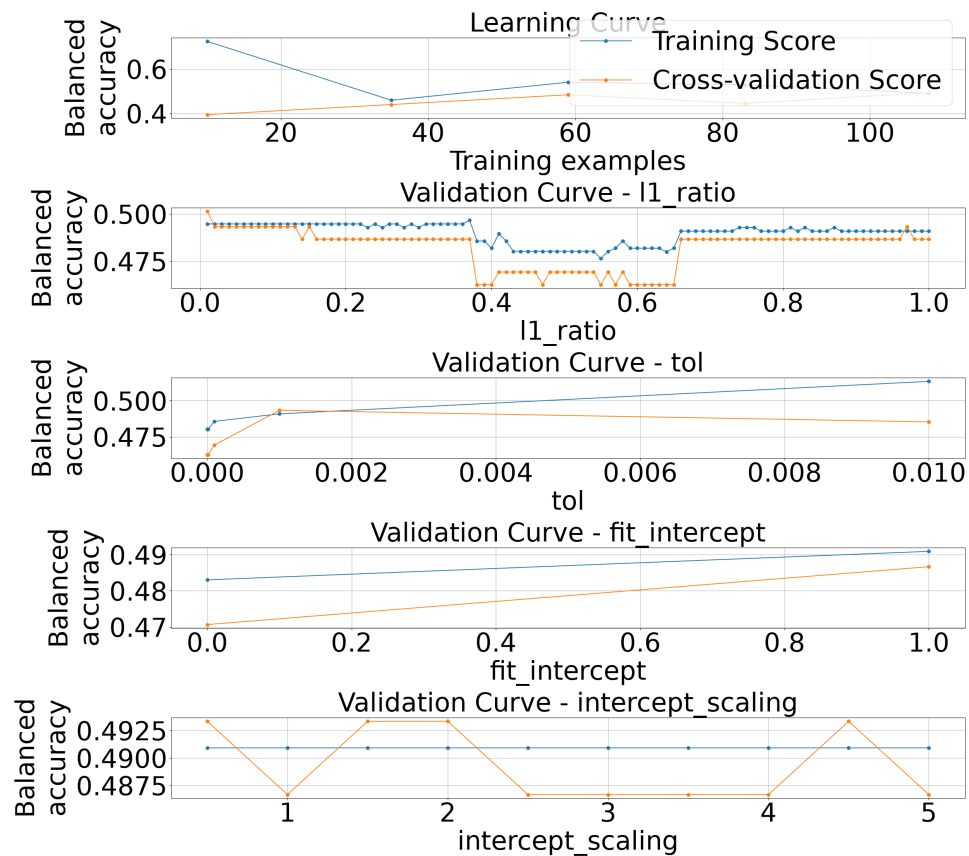


Figure 104 – Logistic Regression Learning & Validation Curves - Raw

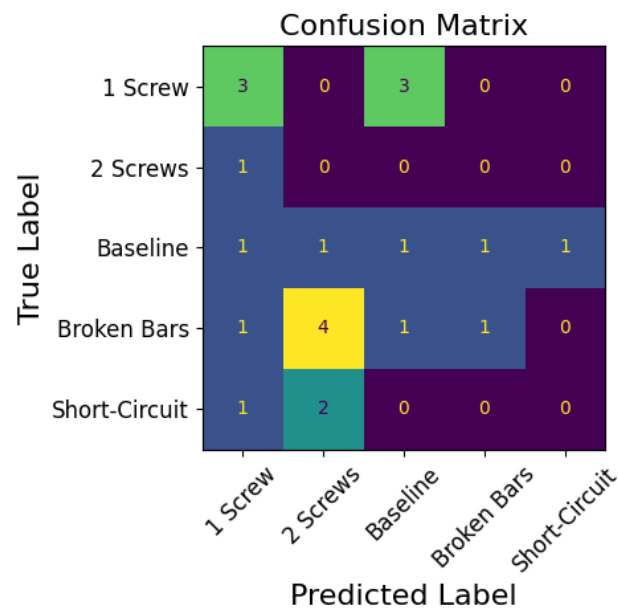


Figure 105 – Logistic Regression Confusion Matrix - FFT

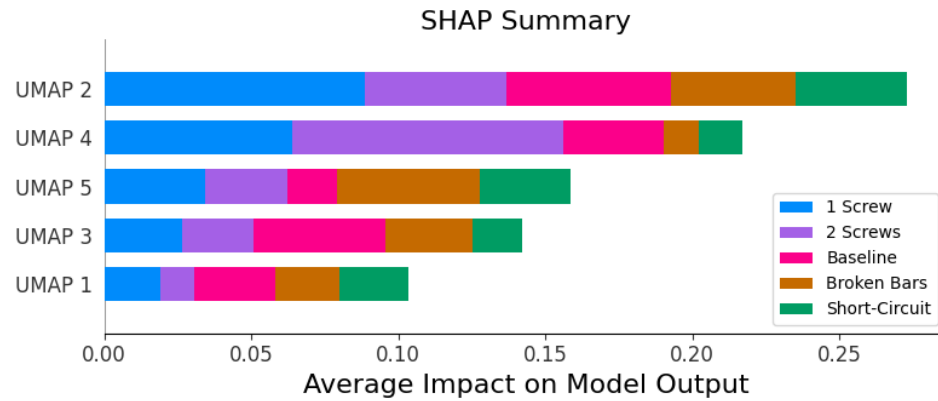


Figure 106 – Logistic Regression SHAP Values Summary - FFT

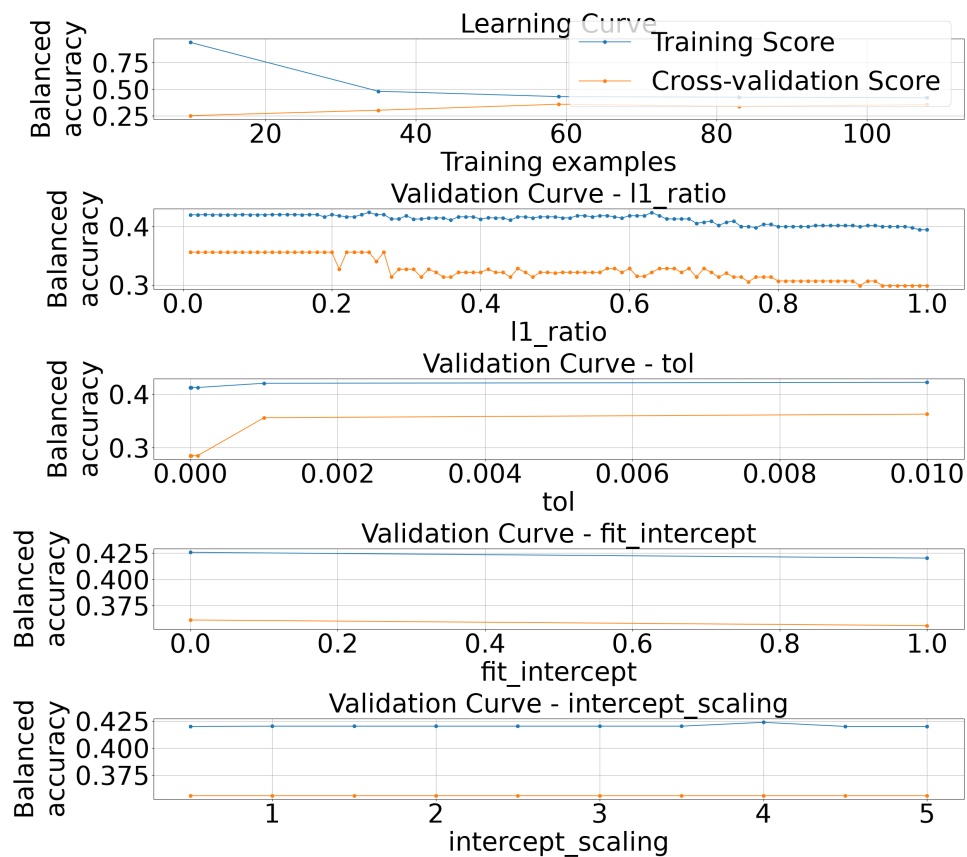


Figure 107 – Logistic Regression Learning & Validation Curves - FFT

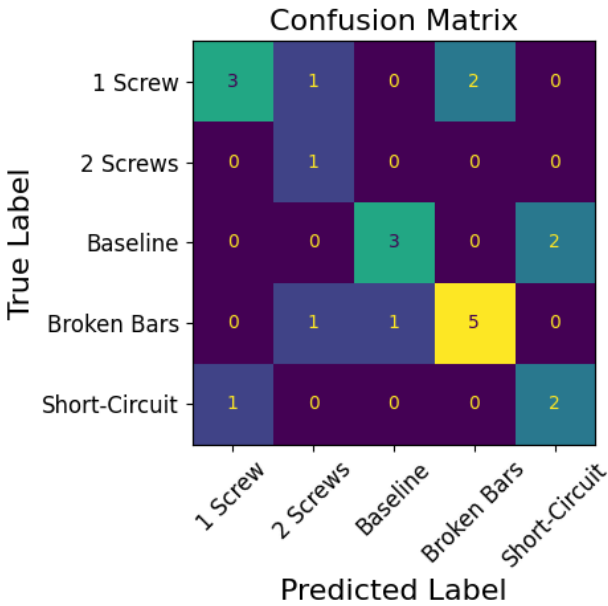


Figure 108 – Logistic Regression Confusion Matrix - Koopman

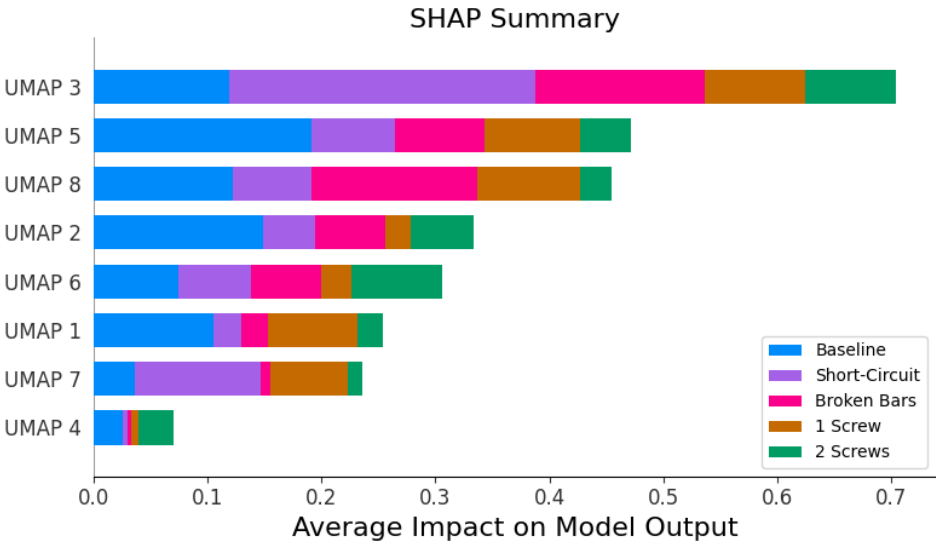


Figure 109 – Logistic Regression SHAP Values Summary - Koopman

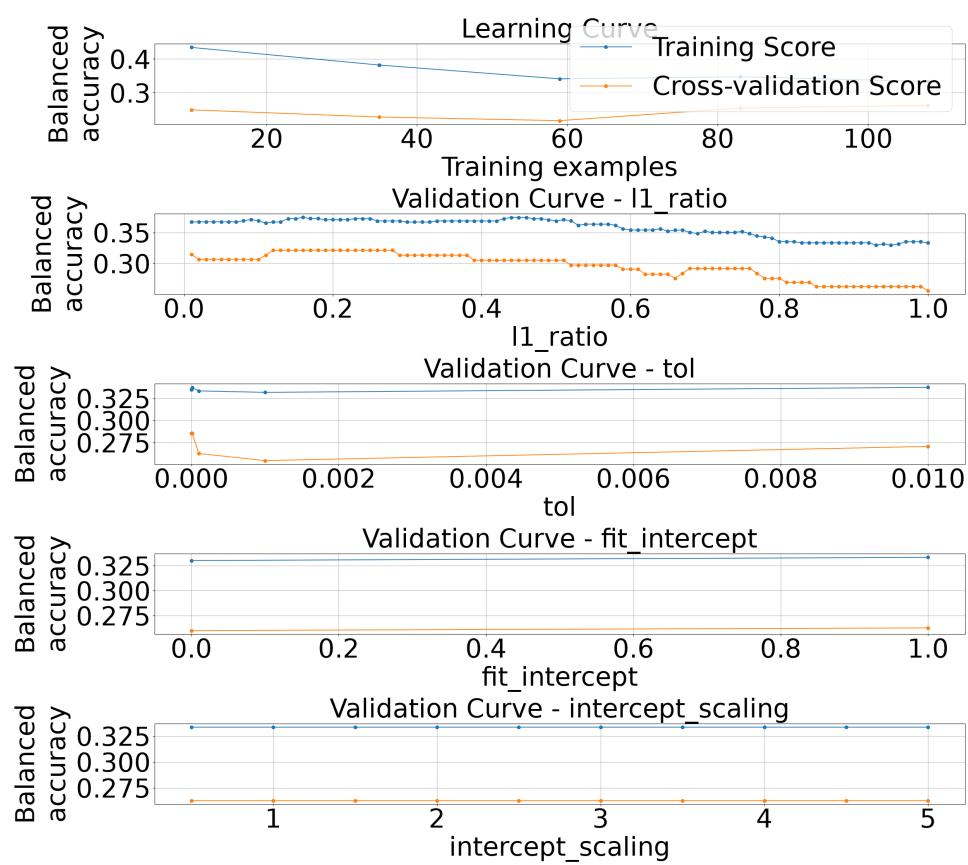


Figure 110 – Logistic Regression Learning & Validation Curves - Koopman

Figure 111 illustrates the class-wise performance of the Gaussian Naive Bayes classifier when trained using the raw current signals. The confusion matrix reveals that the model struggled to separate most fault categories, achieving reliable recognition only for the "Baseline" condition, which was correctly identified in four out of five test instances. The SHAP analysis in Figure 112 indicates that the optimized model relied on only two projected features, both of which exhibited strong influence on the decision process. The learning and validation curves shown in Figure 113 suggest mild overfitting and highlight the sensitivity of the classifier to its hyperparameter configuration.

When the same classifier was trained using FFT-based features, the results shown in Figure 114 indicate a different error pattern. The model achieved its best recognition performance for the "1 Screw" and "Broken Bars" conditions, correctly classifying two out of six and three out of seven test samples, respectively, while the remaining classes were poorly separated. The feature attribution analysis in Figure 115 shows that the fifth spectral feature contributed most strongly to the predictions, followed by the fourth, whereas the remaining features had comparatively minor influence. The learning curves in Figure 116 again suggest the presence of overfitting and demonstrate the impact of hyperparameter tuning on generalization.

The behaviour of the classifier changes significantly when Koopman-derived features are used as inputs. The confusion matrix in Figure 117 shows improved recognition of the "Broken Bars" condition, which was correctly identified in three out of seven test cases. The SHAP summary in Figure 118 reveals a broader distribution of feature importance, with the seventh feature contributing most strongly, followed by the fifth and the first, while the fourth, sixth, and third features played only a minor role. The learning and validation curves in Figure 119 indicate stable generalization behaviour, with no clear signs of overfitting and a consistent response to hyperparameter adjustments.

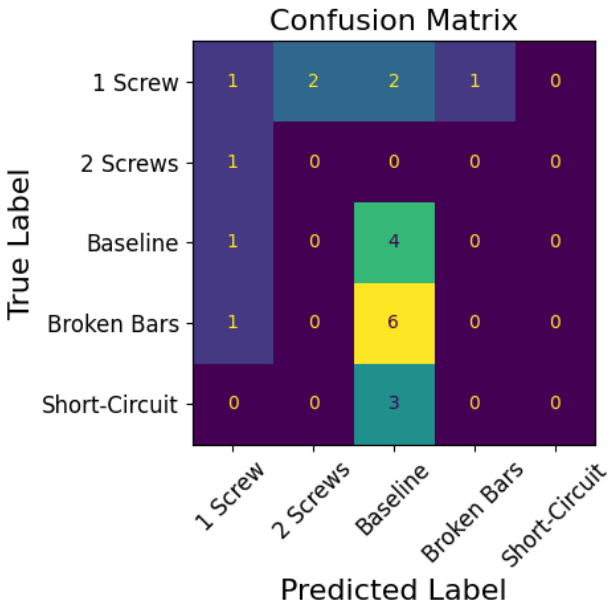


Figure 111 – Gaussian Naive Bayes Confusion Matrix - Raw

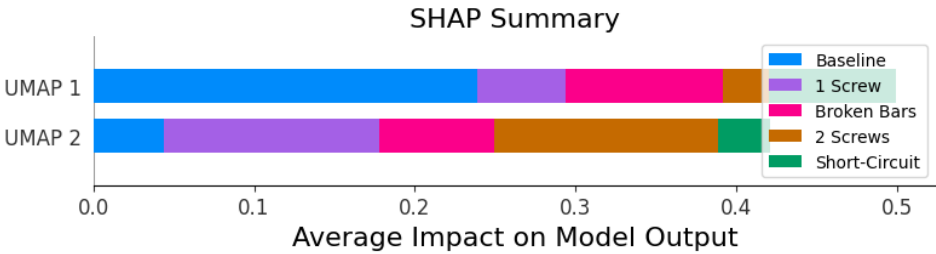


Figure 112 – Gaussian Naive Bayes SHAP Values Summary - Raw

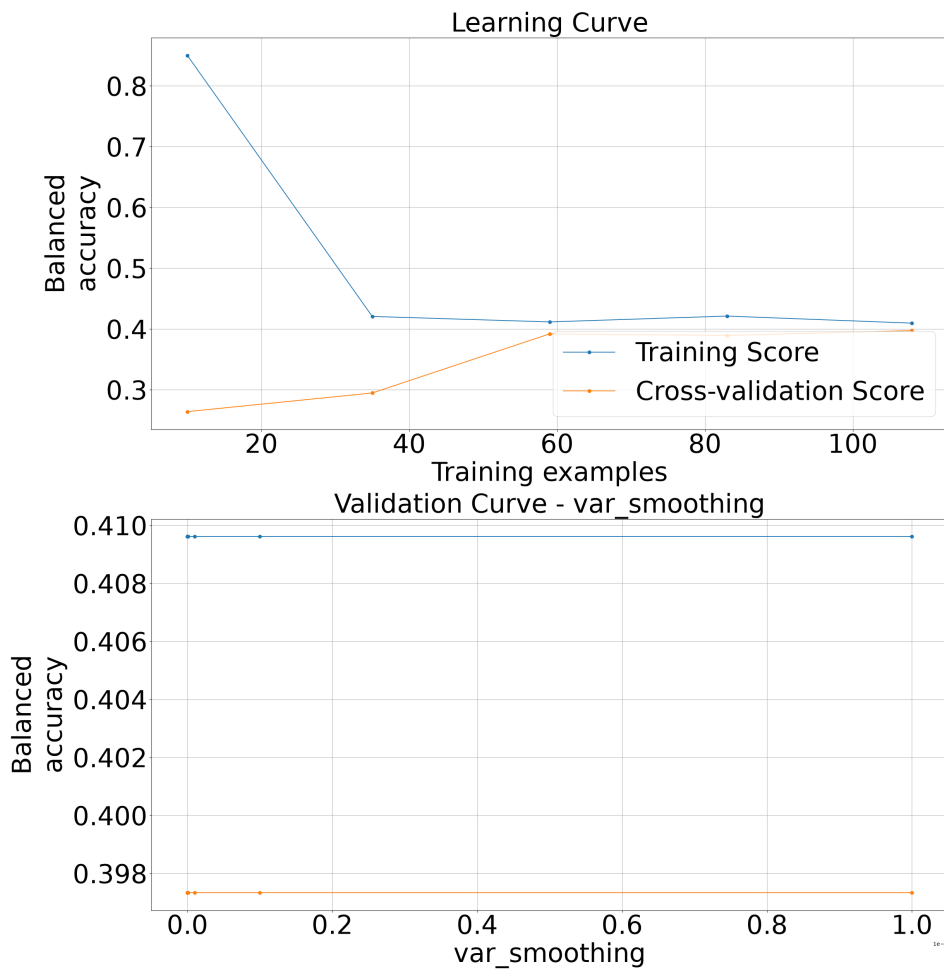


Figure 113 – Gaussian Naive Bayes Learning & Validation Curves - Raw

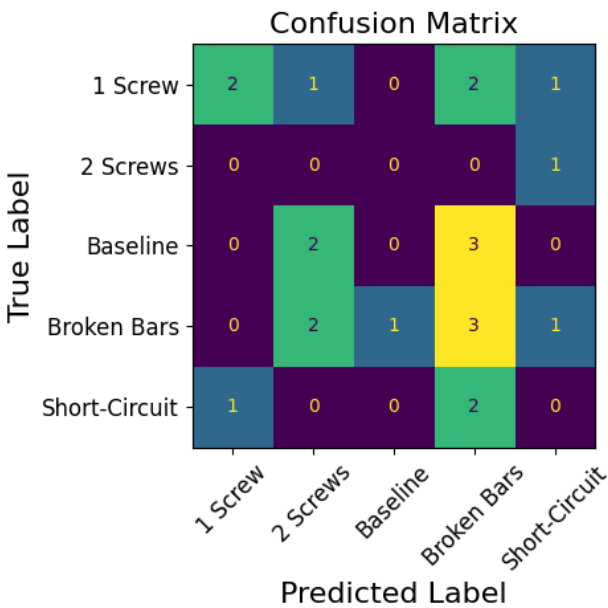


Figure 114 – Gaussian Naive Bayes Confusion Matrix - FFT

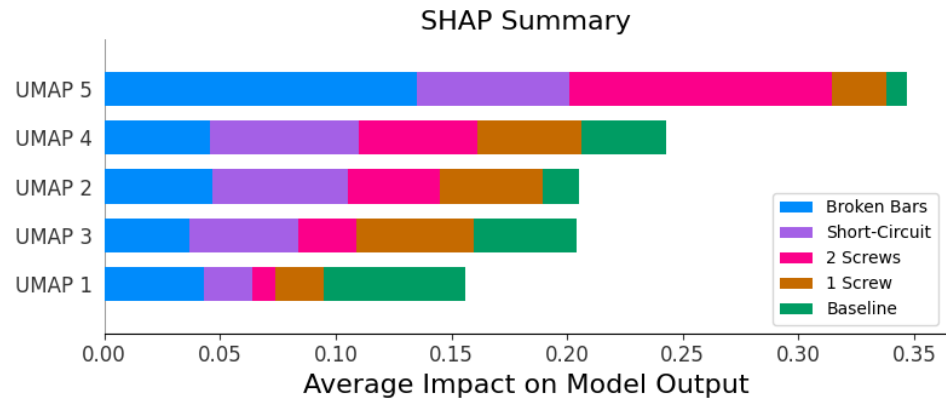


Figure 115 – Gaussian Naive Bayes SHAP Values Summary - FFT

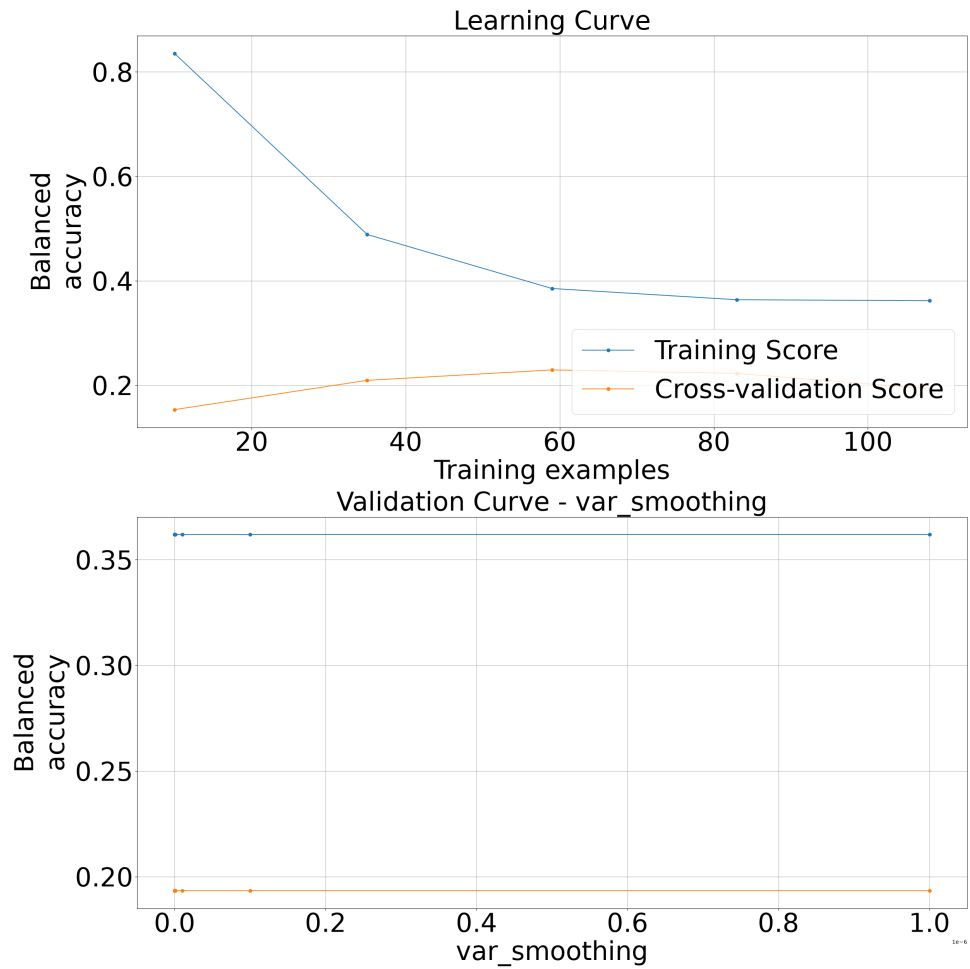


Figure 116 – Gaussian Naive Bayes Learning & Validation Curves - FFT

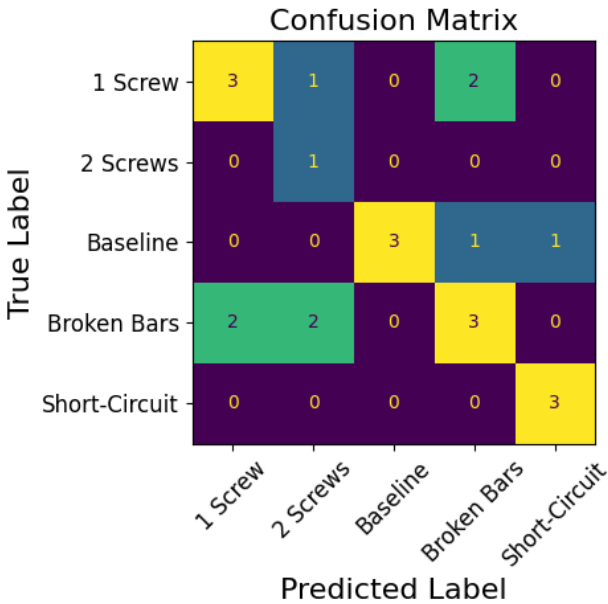


Figure 117 – Gaussian Naive Bayes Confusion Matrix - Koopman

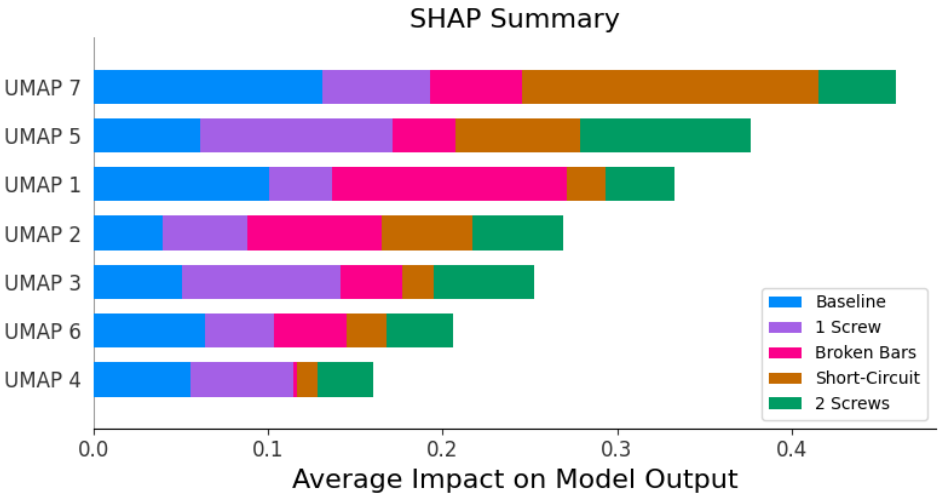


Figure 118 – Gaussian Naive Bayes SHAP Values Summary - Koopman

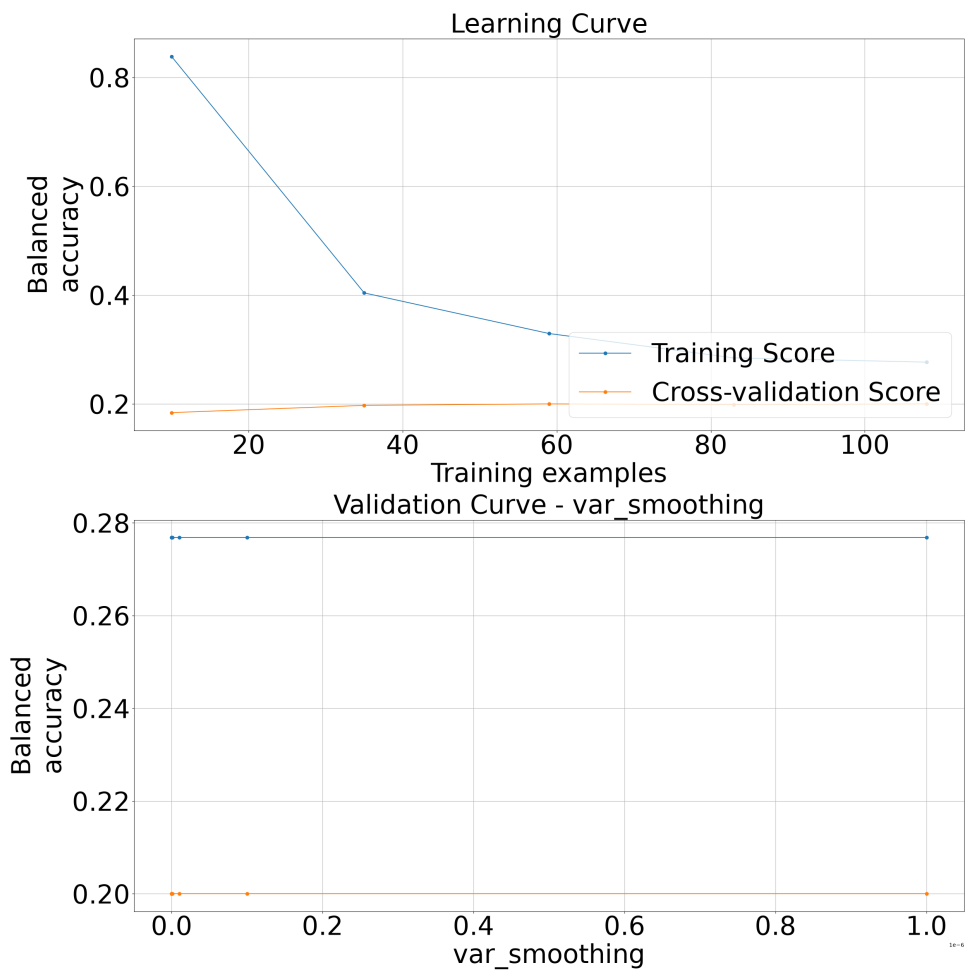


Figure 119 – Gaussian Naive Bayes Learning & Validation Curves - Koopman

The confusion matrix obtained using raw time-domain inputs (Figure 120) reveals that the XGBoost model struggled to correctly separate several fault categories. In particular, signals belonging to the "1 Screw" and "Short-Circuit" classes were predominantly assigned to the "Broken Bars" class, indicating that the discriminative information available in the raw measurements was insufficient for robust class separation. Feature-importance analysis shown in Figure 121 further shows that the trained model relied heavily on only two latent variables after hyperparameter optimization, with the first feature dominating the decision process and the second contributing marginally. The corresponding learning and validation curves shown in Figure 122 suggest mild overfitting behaviour and illustrate how the selected hyperparameters influenced the learning dynamics.

When Fourier-domain features were used (Figure 123), the classification performance degraded significantly. The confusion matrix indicates that the model failed to form reliable class boundaries, with the "Baseline" condition being the only class occasionally recognized correctly. The SHAP analysis shown in Figure 124 shows a more distributed but weak feature contribution pattern, where the first and fifth components provided the strongest influence while the remaining components contributed minimally. The learning curves (Figure 125) again suggest slight overfitting, indicating that the spectral representation alone did not provide sufficiently separable information for this task.

A markedly different behaviour was observed when the Koopman-based feature set was employed (Figure 126). The confusion matrix demonstrates a substantial improvement in class separability, with only the "1 Screw" class condition showing moderate confusion. The SHAP analysis (Figure 127) highlights a richer and more balanced feature contribution profile, where multiple Koopman features played meaningful roles in the decision process, particularly the fourth and fifth components. The learning and validation curves shown in Figure 128 show stable convergence with no evident signs of overfitting, suggesting that the Koopman representation yields a more informative and generalizable feature space for fault classification.

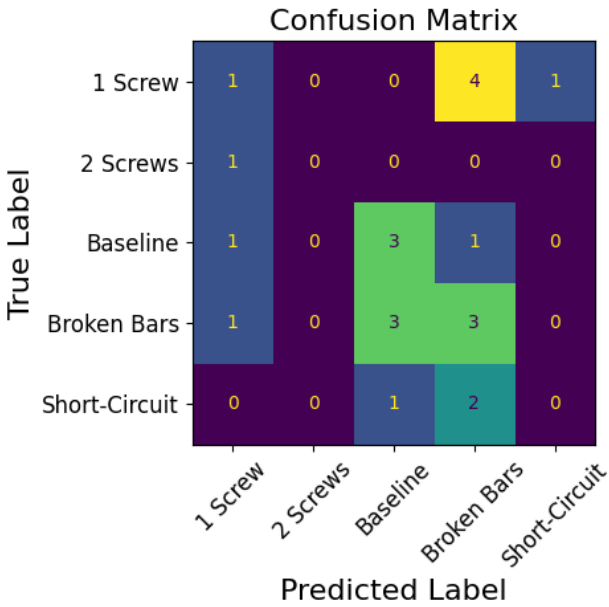


Figure 120 – XGBoost Confusion Matrix - Raw

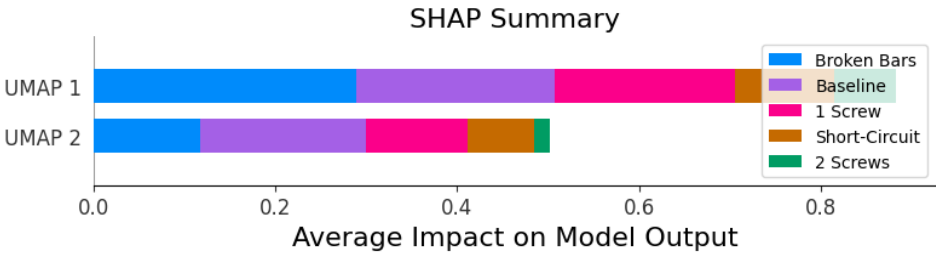


Figure 121 – XGBoost SHAP Values Summary - Raw

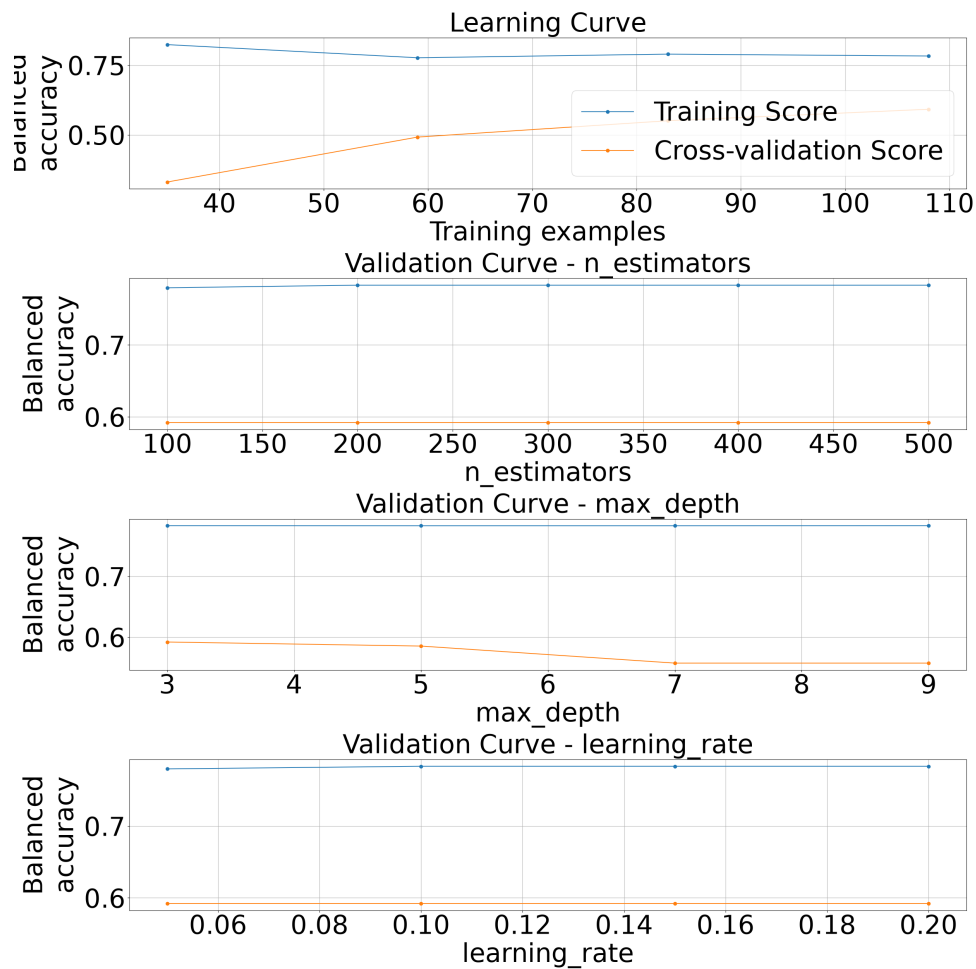


Figure 122 – XGBoost Learning & Validation Curves - Raw

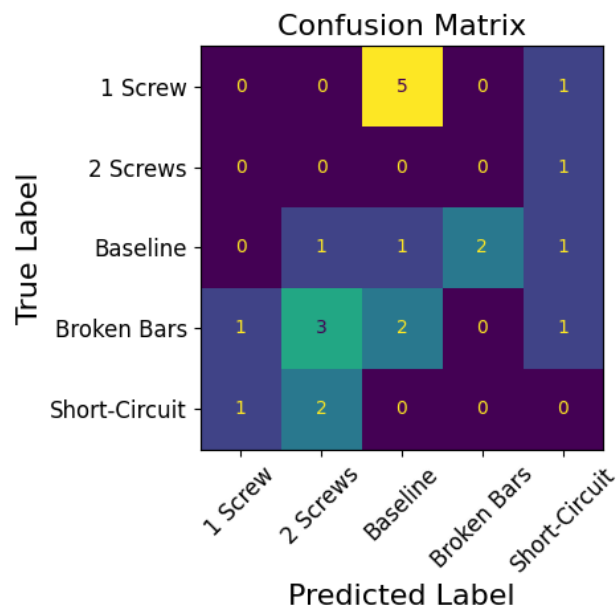


Figure 123 – XGBoost Confusion Matrix - FFT

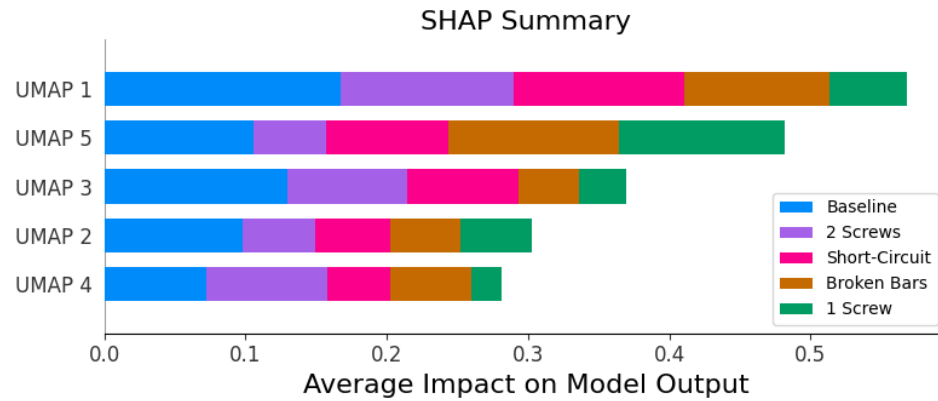


Figure 124 – XGBoost SHAP Values Summary - FFT

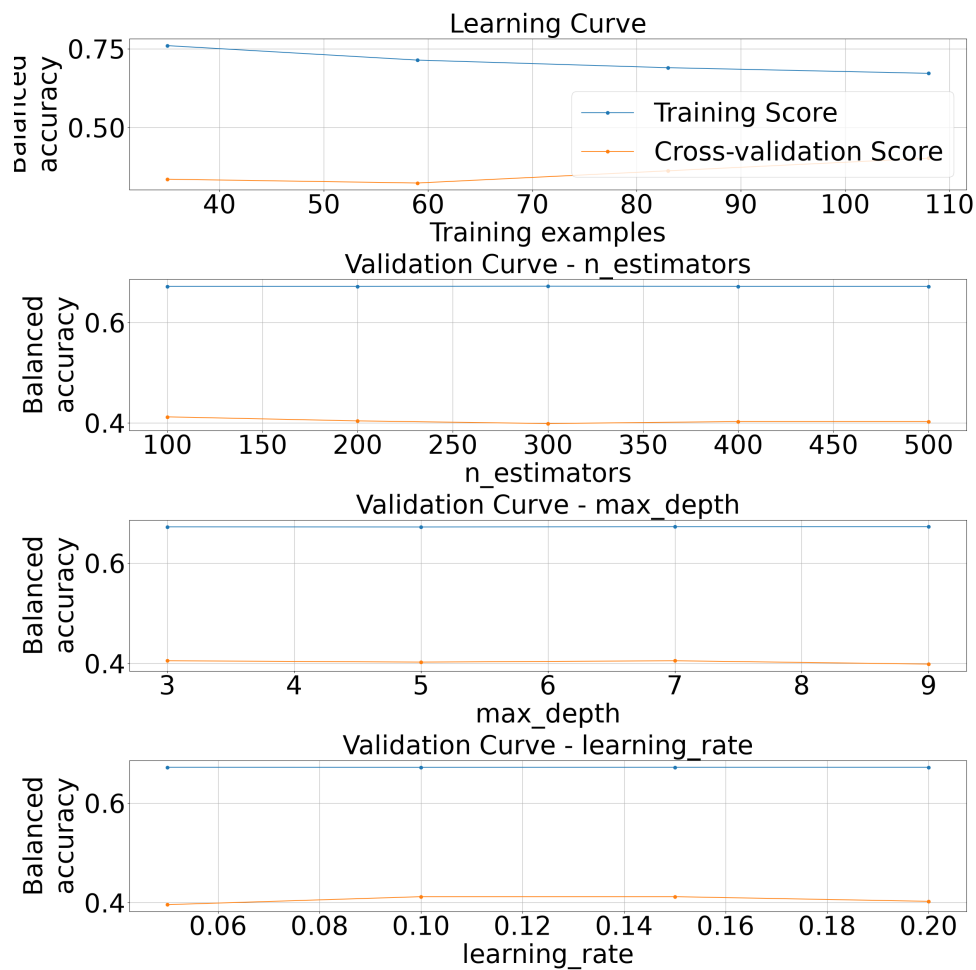


Figure 125 – XGBoost Learning & Validation Curves - FFT

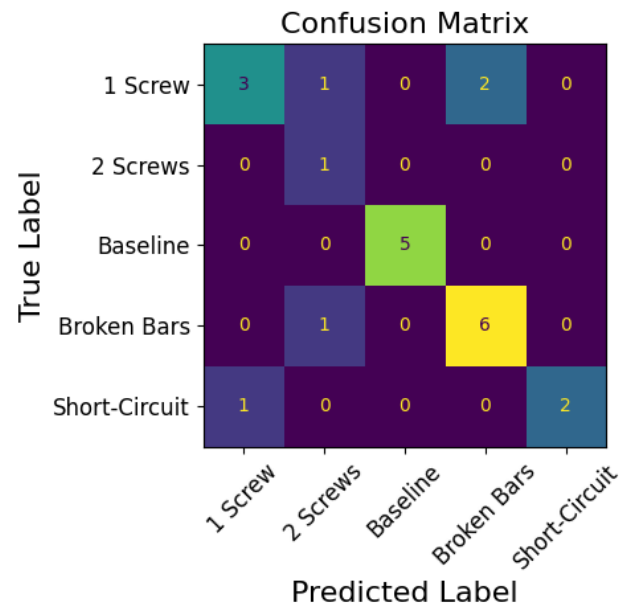


Figure 126 – XGBoost Confusion Matrix - Koopman

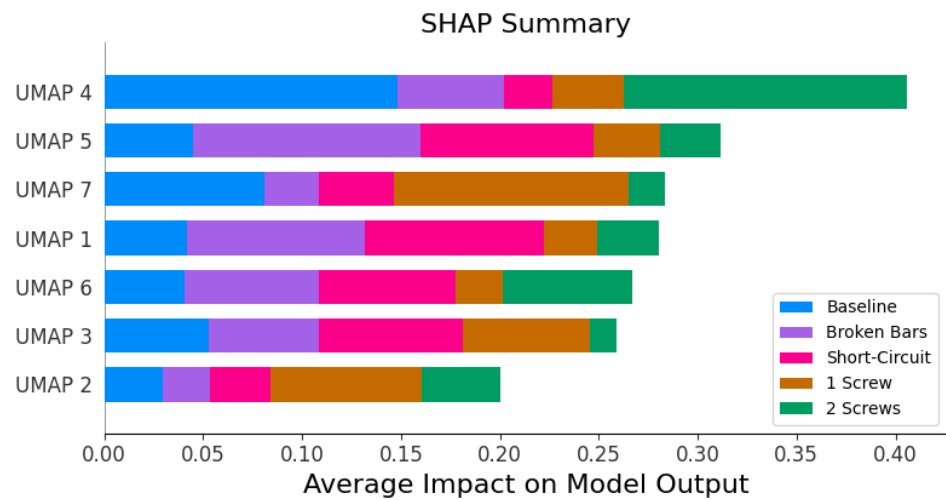


Figure 127 – XGBoost SHAP Values Summary - Koopman

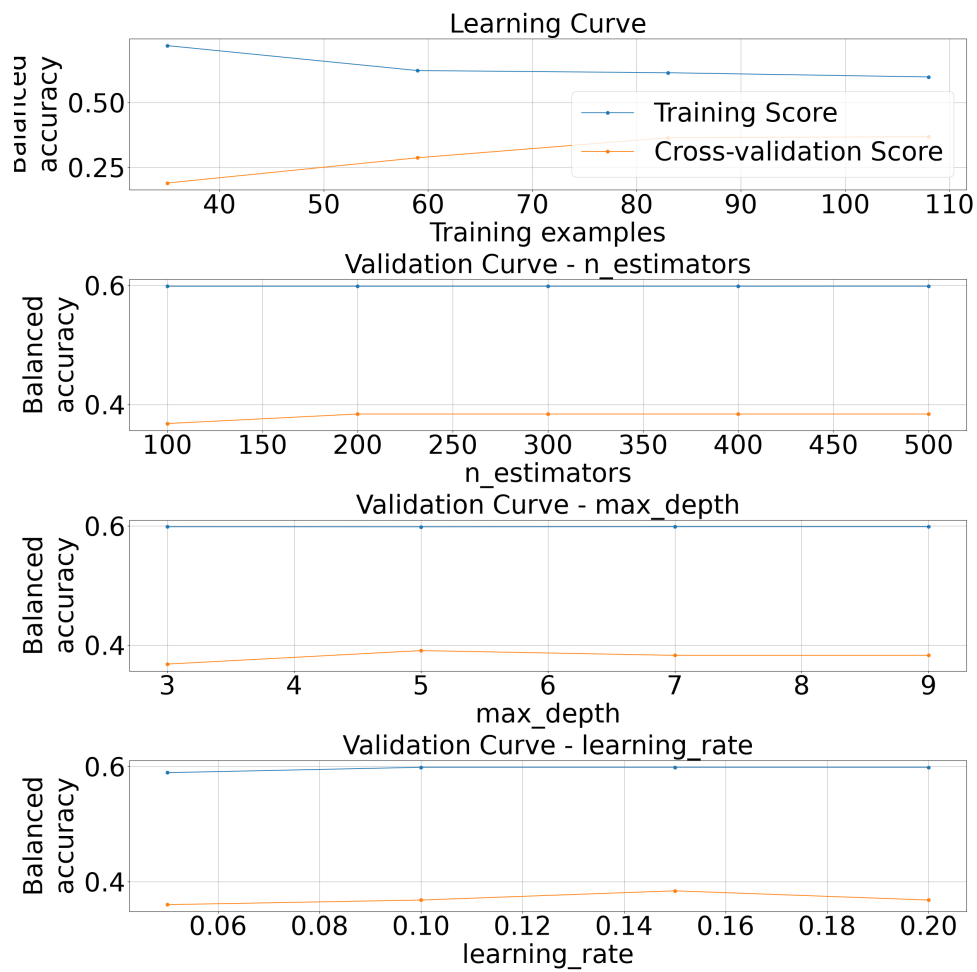


Figure 128 – XGBoost Learning & Validation Curves - Koopman

The performance obtained with the raw time-domain inputs reveals important limitations in the ability of the MLP to capture discriminative patterns directly from the projected signals. The confusion matrix (Figure 129) shows that only the "1 Screw" condition was consistently recognized, achieving four correct classifications out of six test samples, while the remaining classes were frequently misclassified. The SHAP analysis shown in Figure 130 indicates that the final model relied on only two projected features after hyperparameter optimization, both of which exhibited strong influence on the decision process. However, the learning and validation curves (Figure 131) suggest a clear gap between training and validation performance, indicating that the model struggled to generalize and was prone to overfitting when operating directly on raw inputs.

When the FFT-based representation was used, the classification performance remained limited. The confusion matrix shown in Figure 132 indicates that the network was unable to consistently separate the classes, with the best performance again observed for the "1 Screw" condition, where only three out of six test samples were correctly identified. Unlike the raw-signal case, the SHAP analysis (Figure 133) shows that the classification relied on a broader set of projected spectral features, with the first feature contributing most strongly, followed by the fifth and second features. Although several additional features contributed to the decision process, the learning curves (Figure 134) still reveal instability and signs of overfitting, suggesting that the Fourier representation alone was insufficient to provide robust class separation for this dataset.

A substantially different behaviour is observed when the Koopman feature representation is employed. The confusion matrix (Figure 135) shows a more balanced classification performance across the fault categories, although some confusion remains between the "1 Screw" and "Broken Bars" classes, which were occasionally predicted as "2 Screws." The SHAP analysis shown in Figure 136 highlights that the fourth Koopman feature was the most influential in the decision process, followed by the first feature, while the remaining features contributed in a more moderate but still meaningful way. Most importantly, the learning and validation curves shown in Figure 137 indicate stable convergence with no clear evidence of overfitting, demonstrating improved generalization capability compared to the raw and Fourier-based representations. This behaviour suggests that the Koopman features provide a more compact and informative description of the system dynamics, enabling the neural network to learn more robust fault-related patterns.

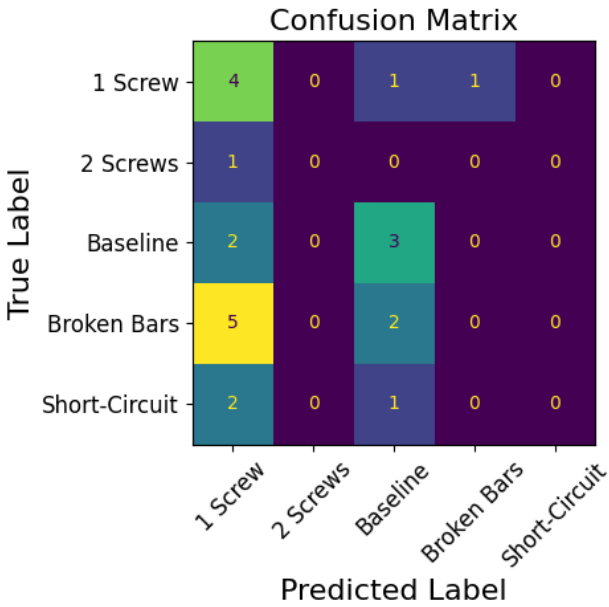


Figure 129 – MLP Confusion Matrix - Raw

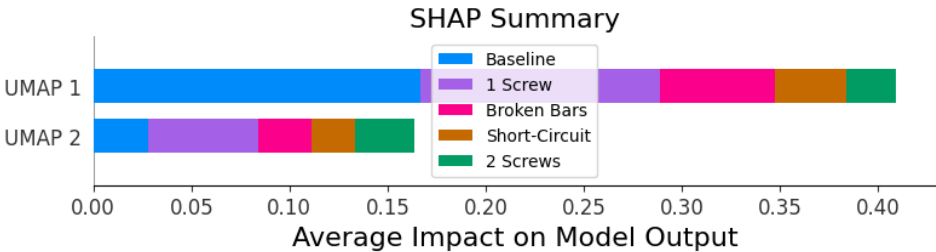


Figure 130 – MLP SHAP Values Summary - Raw

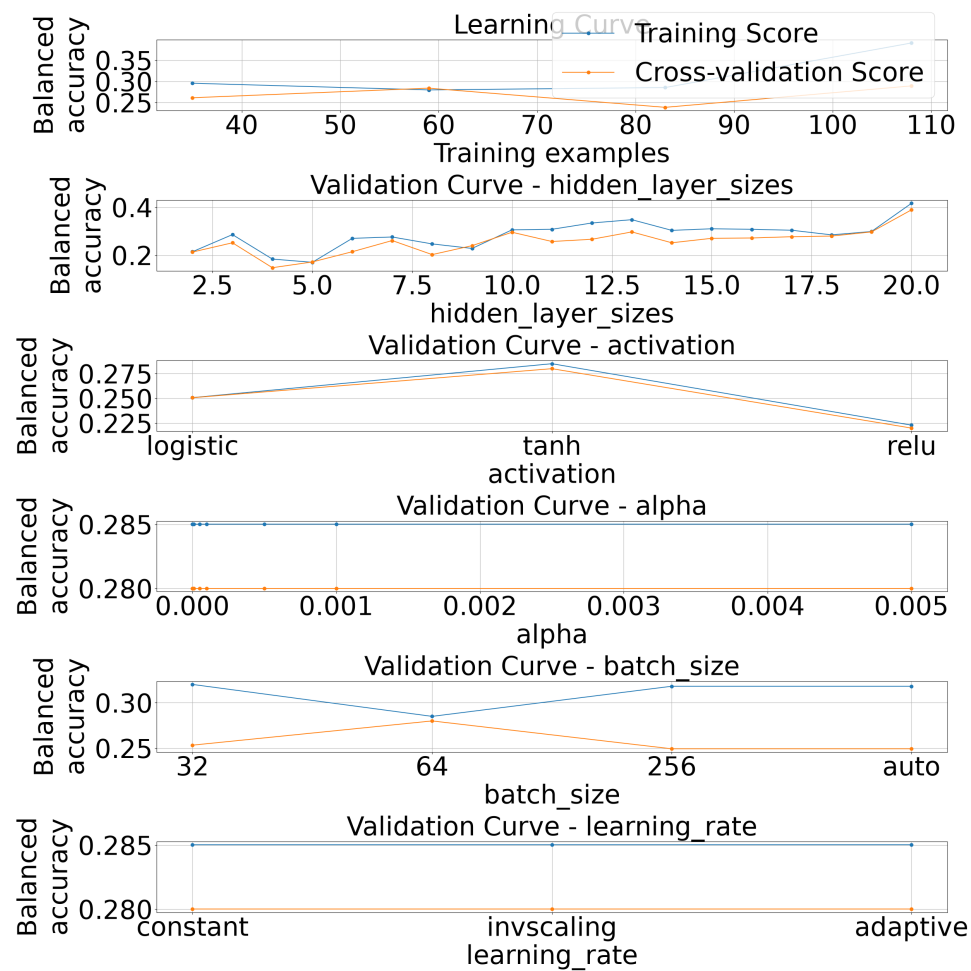


Figure 131 – MLP Learning & Validation Curves - Raw

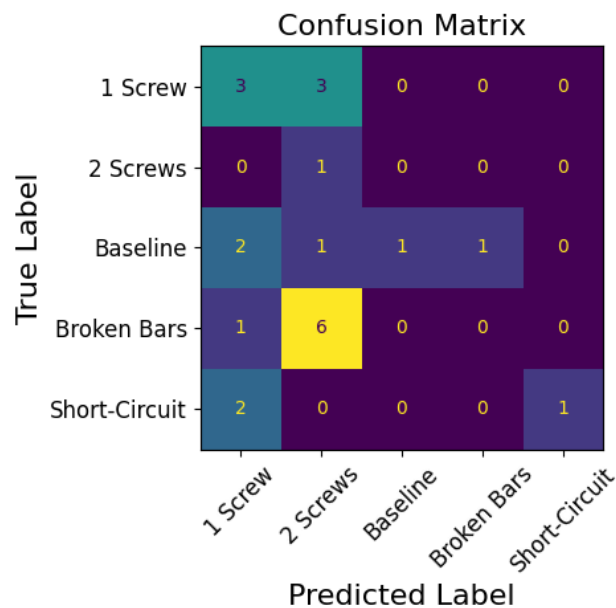


Figure 132 – MLP Confusion Matrix - FFT

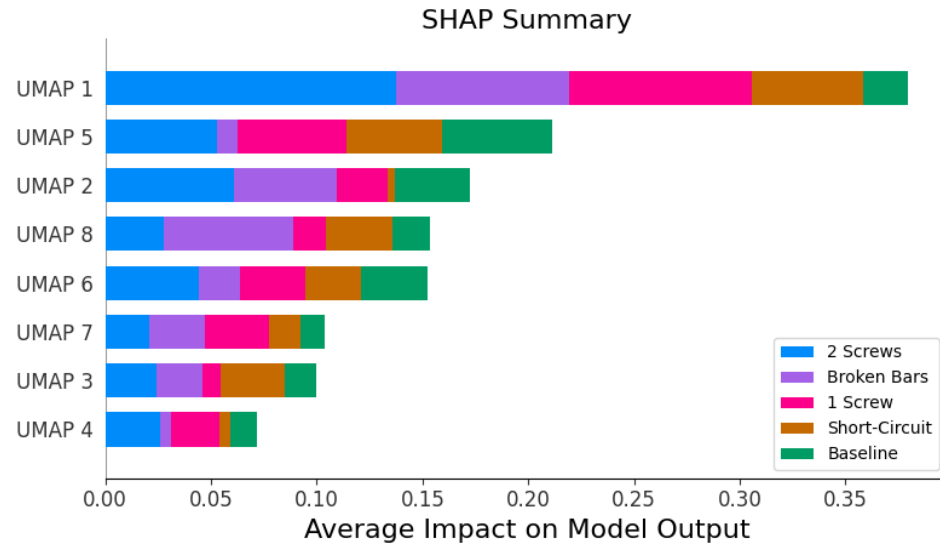


Figure 133 – MLP SHAP Values Summary - FFT

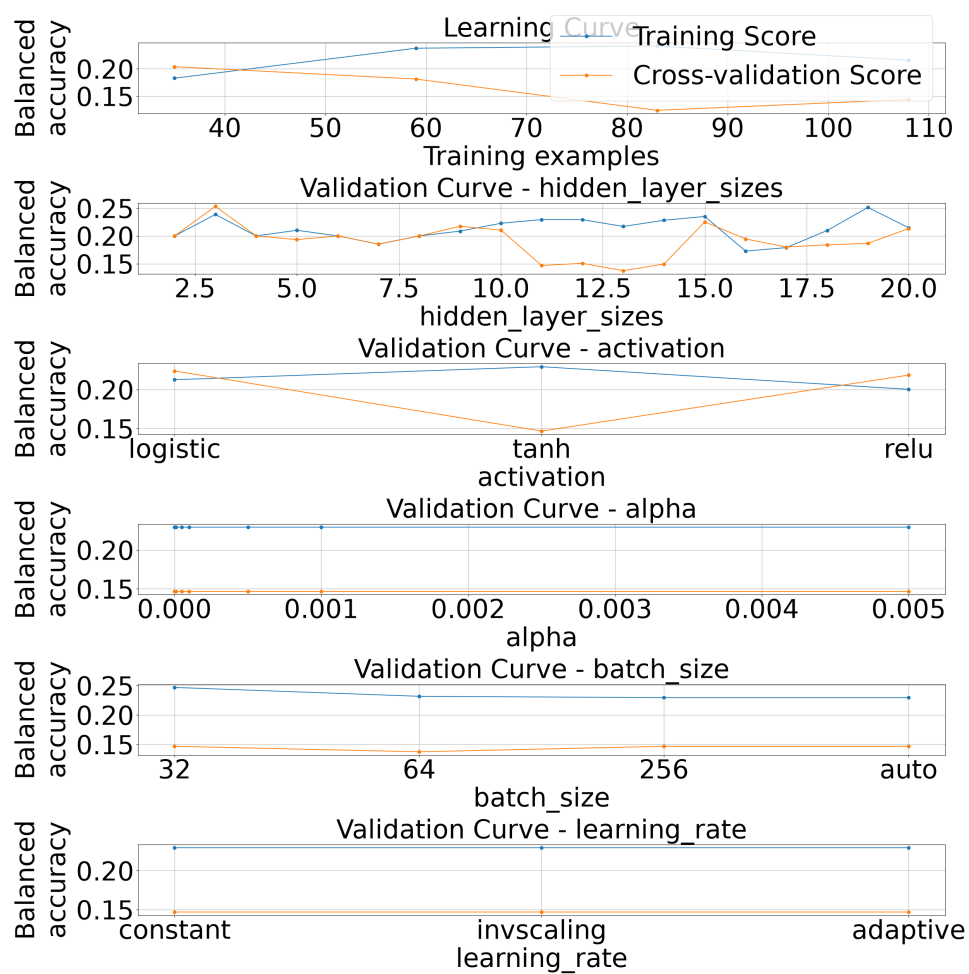


Figure 134 – MLP Learning & Validation Curves - FFT

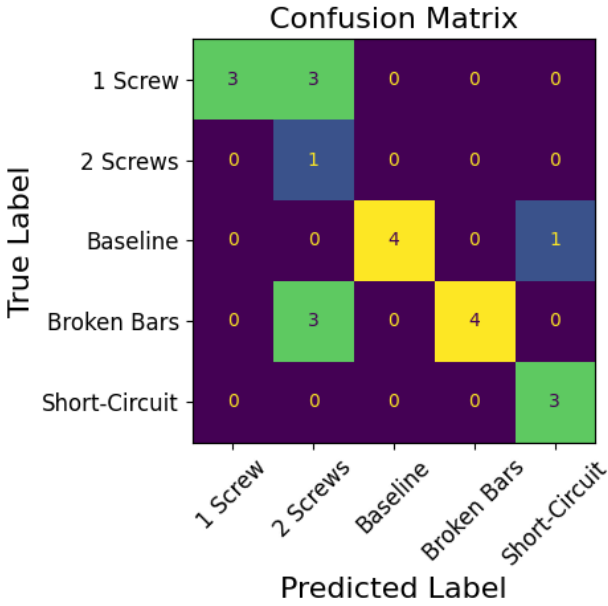


Figure 135 – MLP Confusion Matrix - Koopman

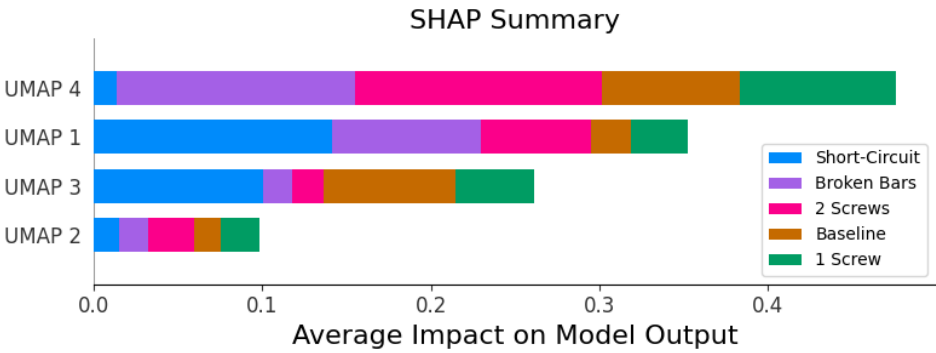


Figure 136 – MLP SHAP Values Summary - Koopman

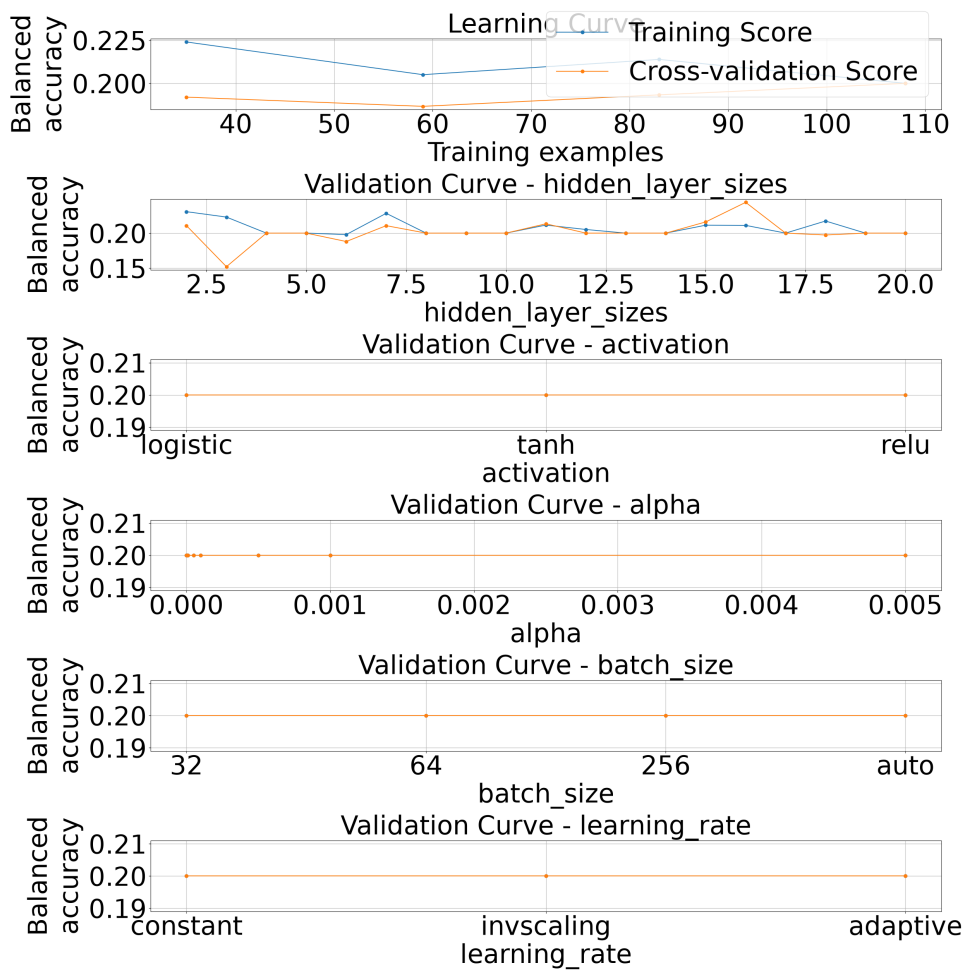


Figure 137 – MLP Learning & Validation Curves - Koopman

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	87.2%	96.1%	90.3%	79.1%	72.1%	86.4%	97.4%
Test Accuracy	90.5%	90.2%	87.9%	72.8%	70.7%	91.1%	91.7%
F1 Score	0.90	0.89	0.88	0.67	0.64	0.90	0.91
ROC-AUC	0.97	0.98	0.96	0.92	0.92	0.99	0.98

Table 10 – Supervised Learning Results - C

4.2 Classification Pipeline Results and Evaluation

This section presents and interprets the results obtained from both proposed supervised and unsupervised flow rate classification pipeline applied to the Koopman features extracted from each signal in the dataset.

4.2.1 Only 3 Currents (C)

4.2.1.1 Supervised Pipeline

Table 10 summarize the overall performance of all tested supervised algorithms, showing that MLP was the best performing algorithm followed by XGBoost, k-NN SVM, MLP and Random Forest. The Logistic Regression and Gaussian Naive Bayes had lower accuracies due to the algorithm requirement being broken. Logistic Regression, for example, uses linear combinations of the inputs to classify the data, but if the data is nonlinear, then the algorithm will struggle more to learn the feature manifold and make correct classifications. Likewise, GNB assumes Gaussian distributed features so, if this assumption is broken, it would lead to a lower classification accuracy. However, the UMAP + SMOTE preprocessing step partially mitigates this limitations. By enforcing local metric linearity and introducing neighbourhood-based interpolations, the resulting feature space becomes more compatible with linear decision boundaries and Gaussian class-conditional modeling, allowing these algorithms achieve relatively high accuracies despite their theoretical limitations.

Figure 138 presents the algorithm's ability to learn each individual class, from which is clear that the k-NN algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 139 presents how much each projected feature contributes to the k-NN classification process, the first feature had the most impact followed by the fourth, then second, the tenth and fifth were the ones with least impact but still having meaningful contribution for the classification. Finally, Figure 140 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

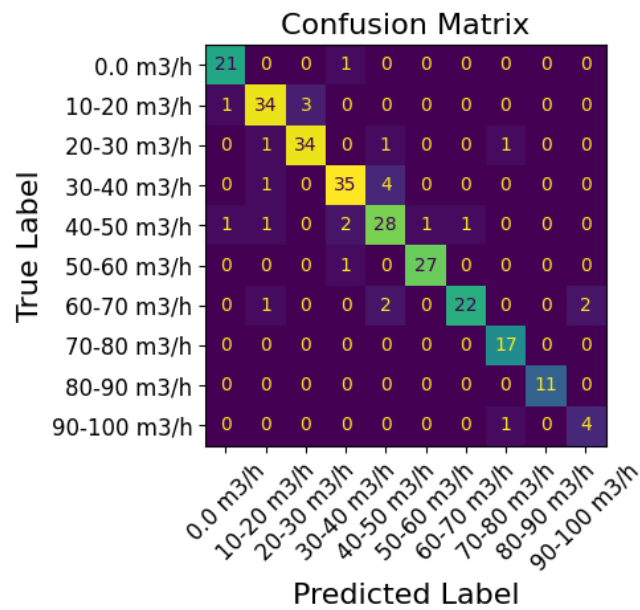


Figure 138 – k-NN Confusion Matrix - C

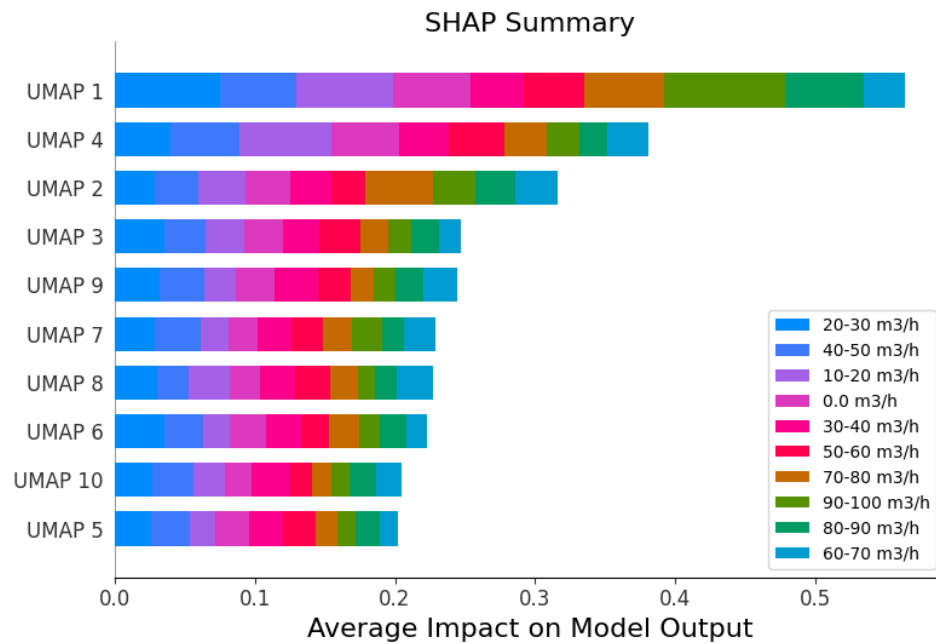


Figure 139 – k-NN SHAP Values Summary - C

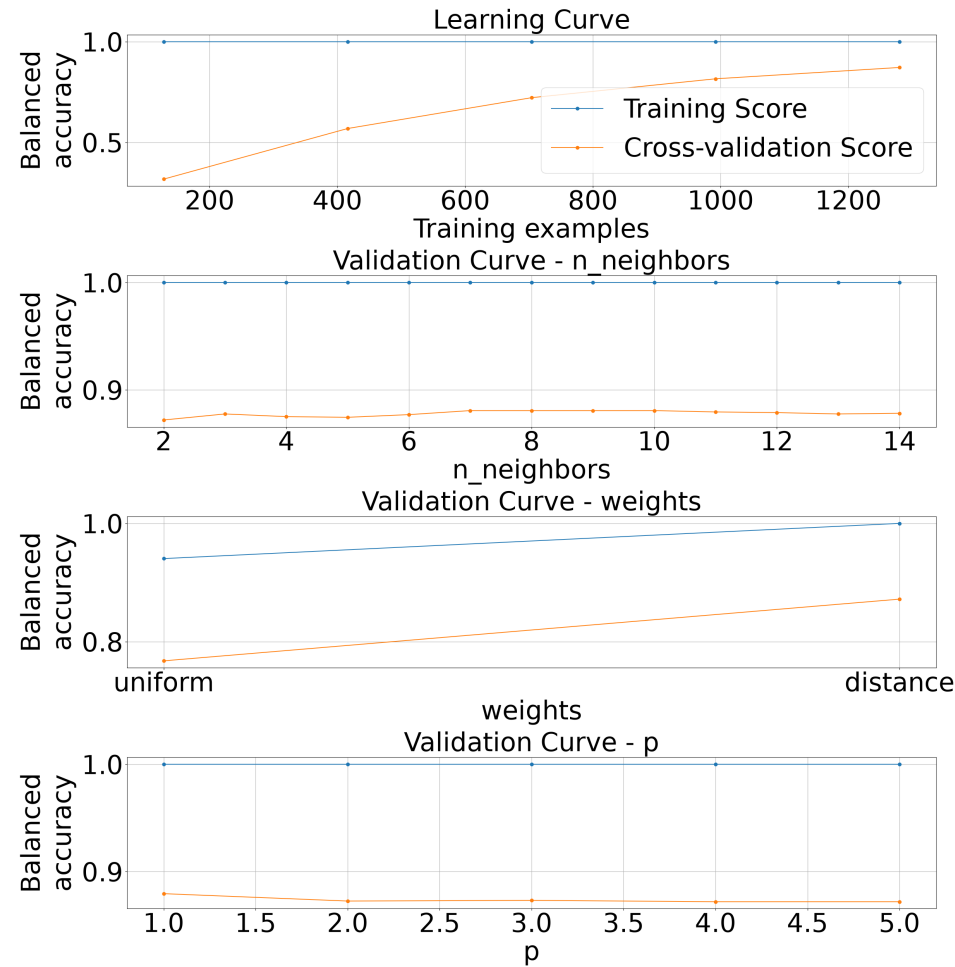


Figure 140 – k-NN Learning & Validation Curves - C

Figure 141 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 142 presents how much each projected feature contributes to the SVM classification process, the first feature had the most impact followed by the second, and third, with all having a great impact on the learning process. Finally, Figure 143 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

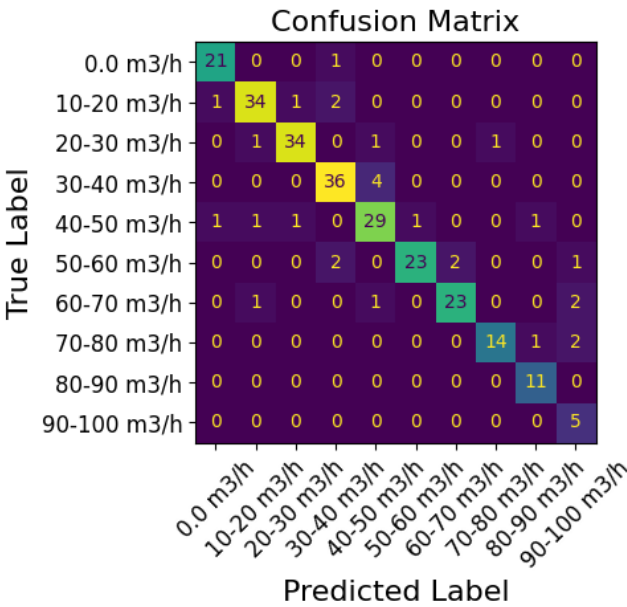


Figure 141 – SVM Confusion Matrix - C

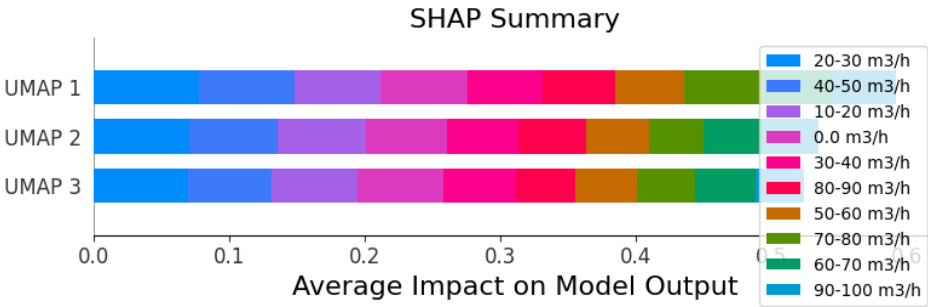


Figure 142 – SVM SHAP Values Summary - C

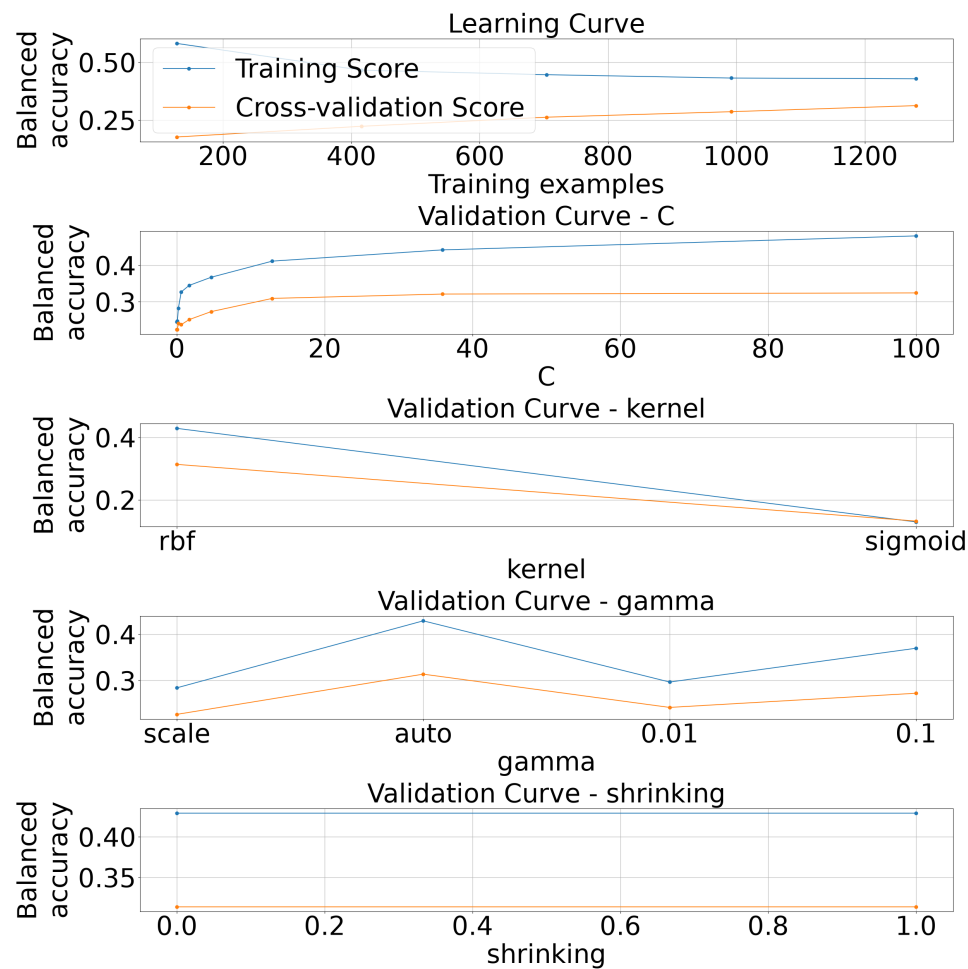


Figure 143 – SVM Learning & Validation Curves - C

Figure 144 presents the algorithm's ability to learn each individual class, from which is clear that the Random Forest algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 145 presents how much each projected feature contributes to the Random Forest classification process, the first feature had the most impact followed by the third, and second, with all having a great impact on the learning process. Finally, Figure 146 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

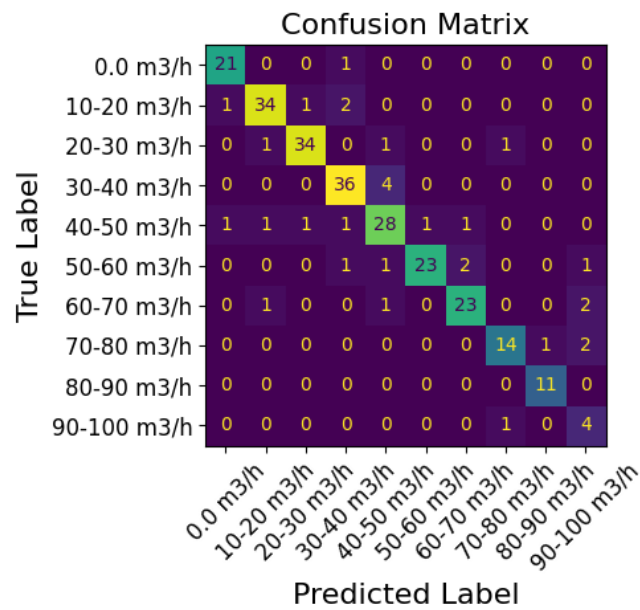


Figure 144 – Random Forest Confusion Matrix - C

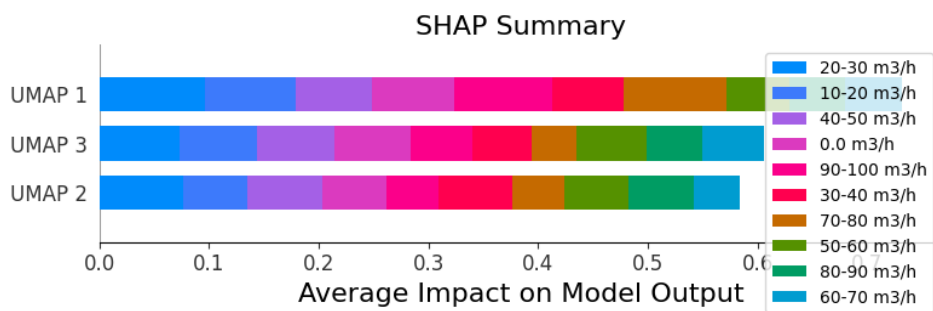


Figure 145 – Random Forest SHAP Values Summary - C

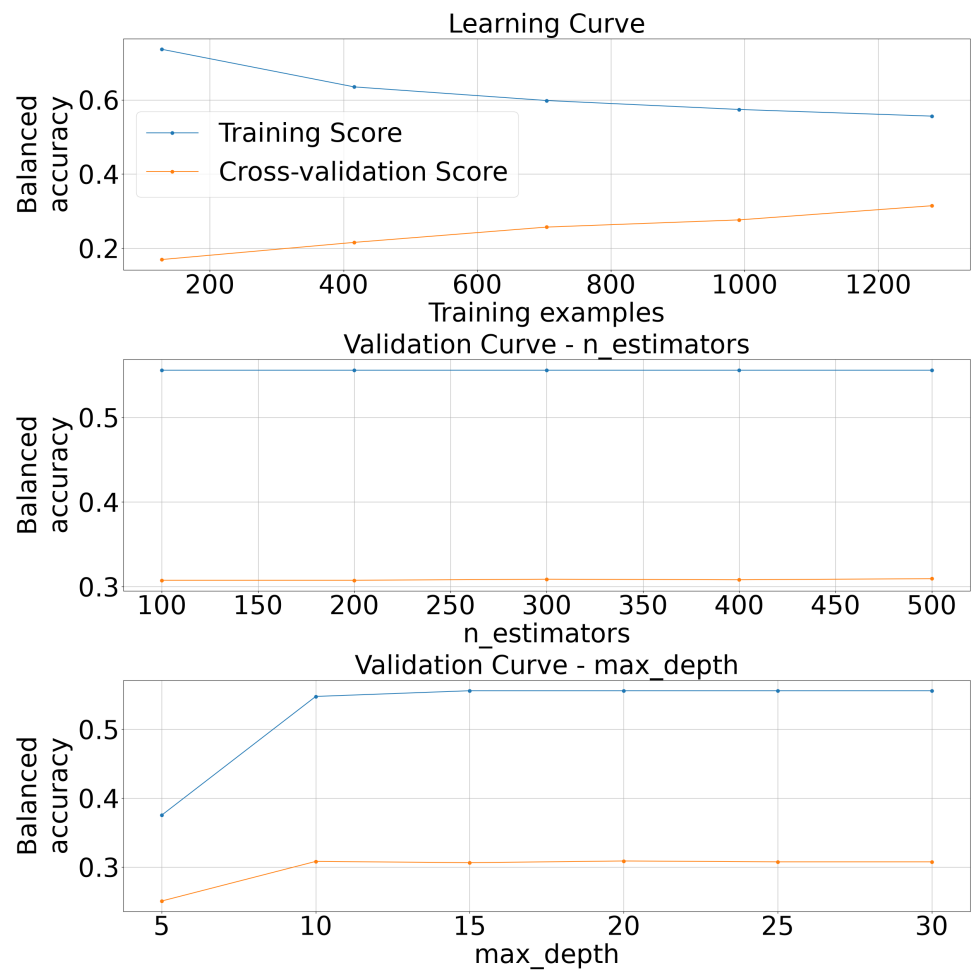


Figure 146 – Random Forest Learning & Validation Curves - C

Figure 147 presents the algorithm's ability to learn each individual class, from which is clear that the Logistic Regression algorithm was able to learn all classes quite well, having more trouble learning the classes "50-60 m^3/h " and "30-40 m^3/h ", with a total of 20 and 16 misclassifications respectively. Figure 148 presents how much each projected feature contributes to the Logistic Regression classification process, the first feature had the most impact followed by the thirteenth, third, second, and fifth with most impact, while the eleventh, tenth, eighth and sixth are the projected features with least impact on the learning process. Finally, Figure 149 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

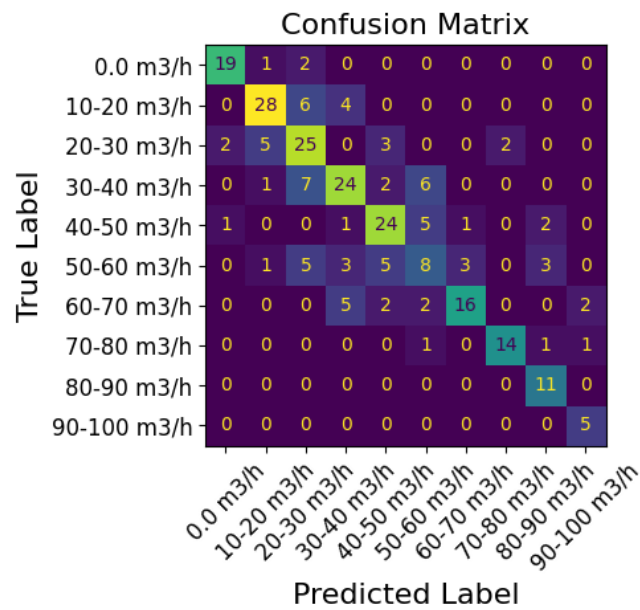


Figure 147 – Logistic Regression Confusion Matrix - C

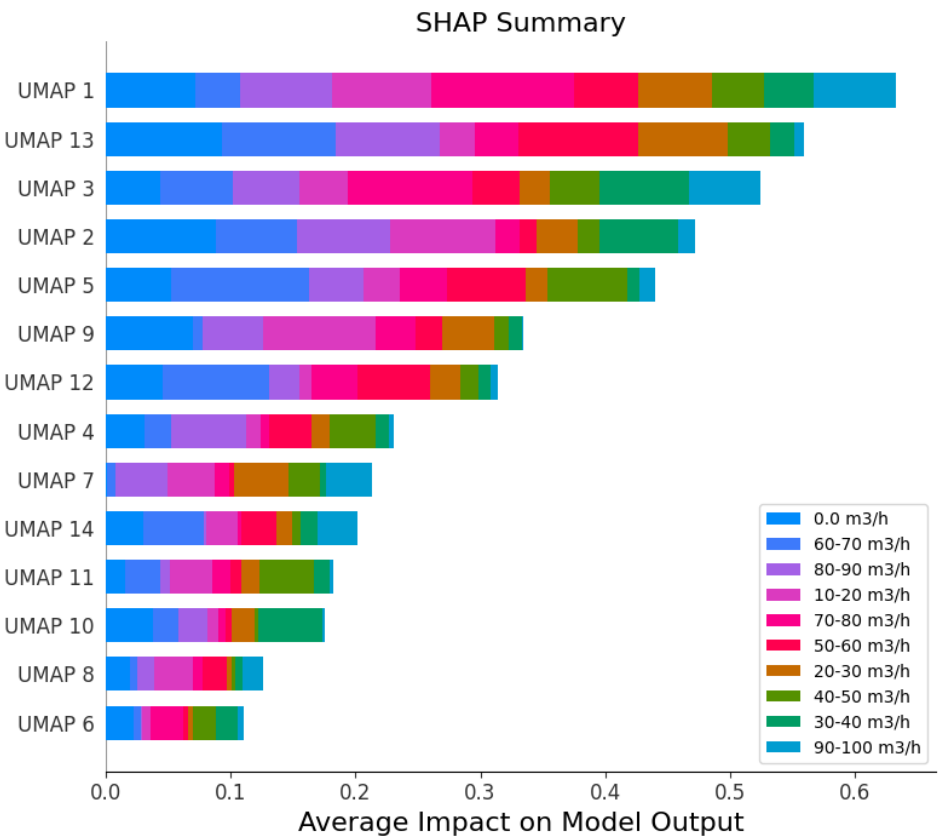


Figure 148 – Logistic Regression SHAP Values Summary - C

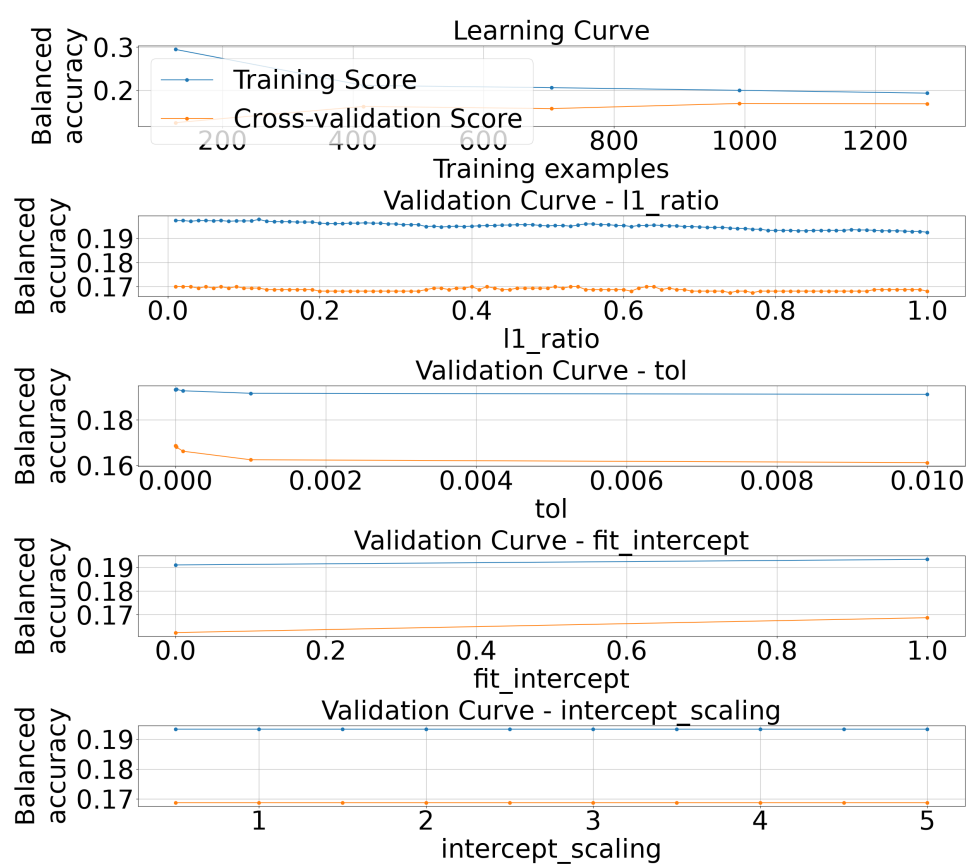


Figure 149 – Logistic Regression Learning & Validation Curves - C

Figure 150 presents the algorithm's ability to learn each individual class, from which is clear that the GNB algorithm was able to learn all classes quite well, having a little more trouble learning "30-40 m^3/h " and "50-60 m^3/h " classes, mistaking them for the "40-50 m^3/h ", with 16 and 17 mistakes respectively. Figure 151 presents how much each projected feature contributes to the Gaussian Naive Bayes classification process, the third feature had the most impact followed by the second, while the tenth, sixth, eighth and twelfth projected features had the least impact on the learning process. Finally, Figure 152 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

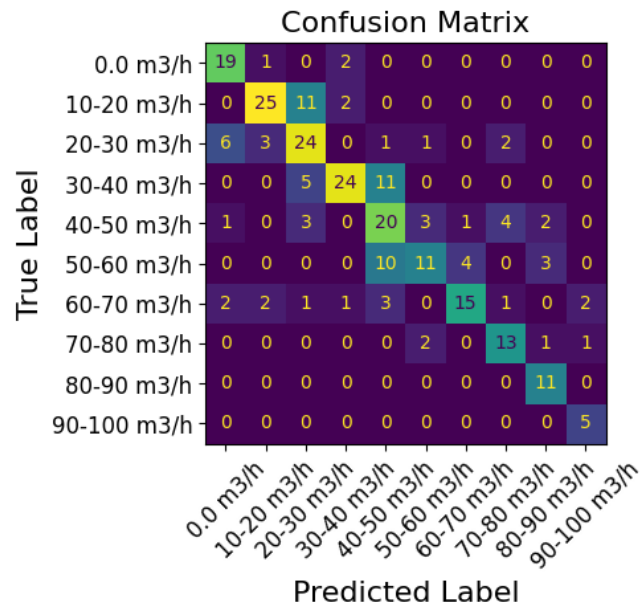


Figure 150 – GNB Confusion Matrix - C

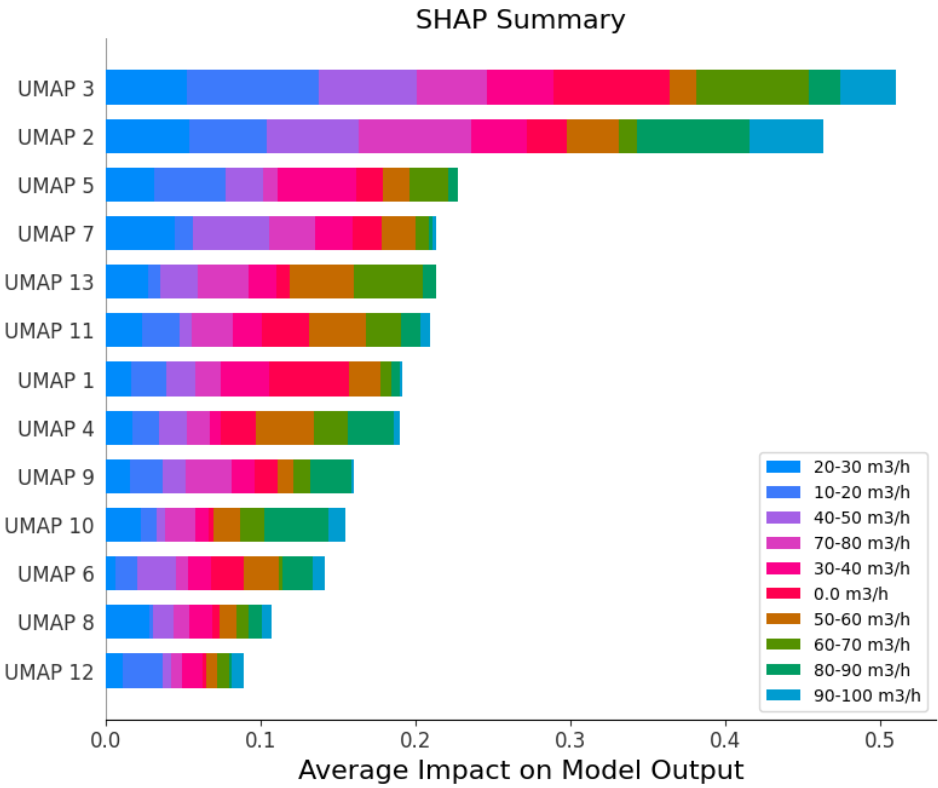


Figure 151 – GNB SHAP Values Summary - C

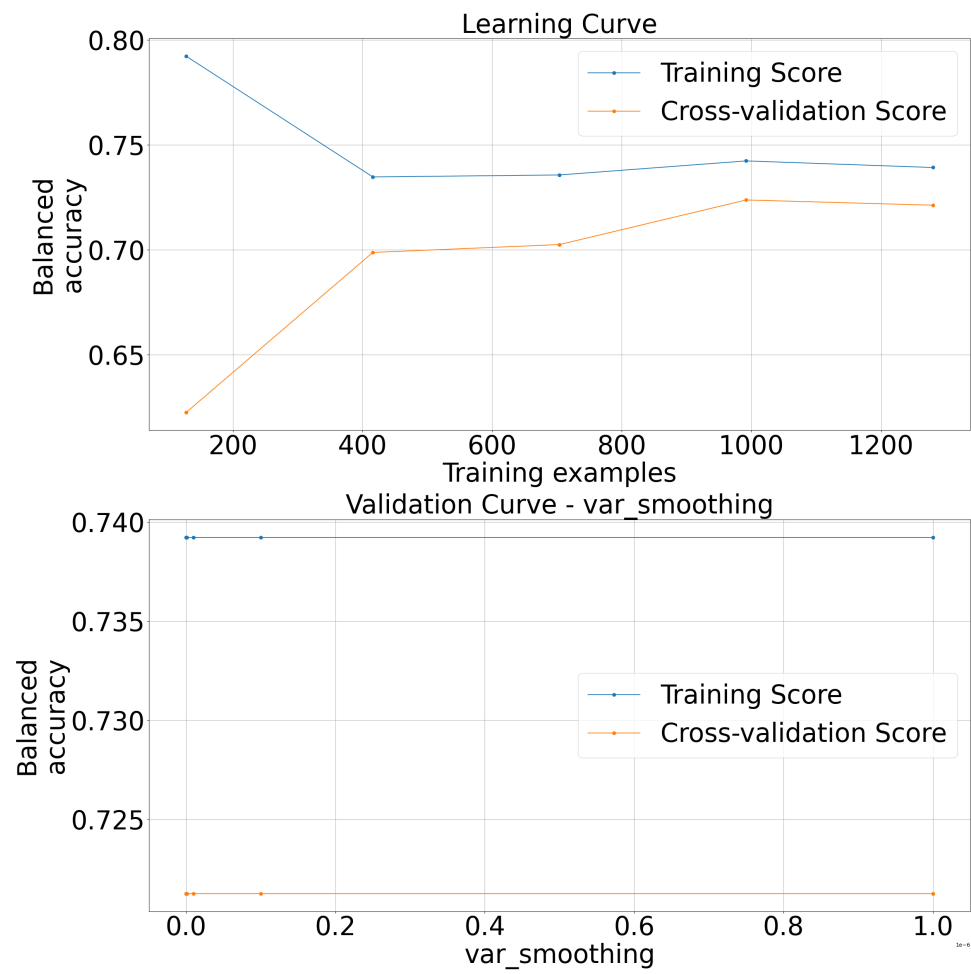


Figure 152 – GNB Learning & Validation Curves - C

Figure 153 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "40-50 m^3/h " class, mistaking it for the "50-60 m^3/h ". Figure 154 presents how much each projected feature contributes to the XGBoost classification process, the first feature had the most impact followed by the fourth, and second, while the tenth, sixth, seventh and eighth projected features had the least impact on the learning process. Finally, Figure 155 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

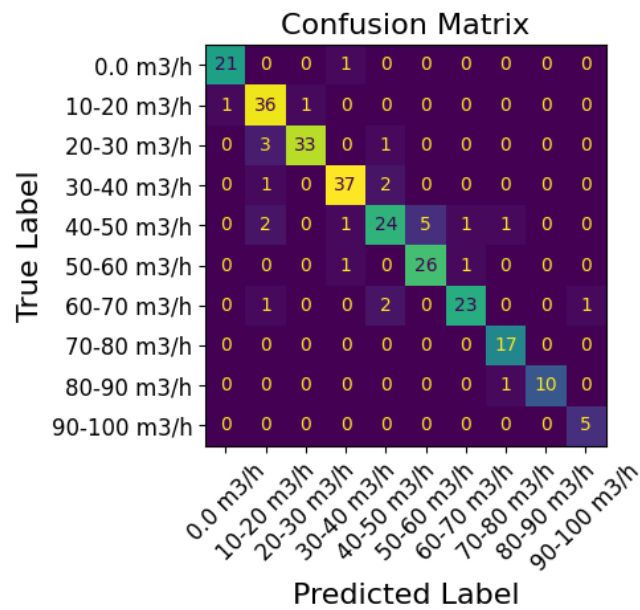


Figure 153 – XGBoost Confusion Matrix - C

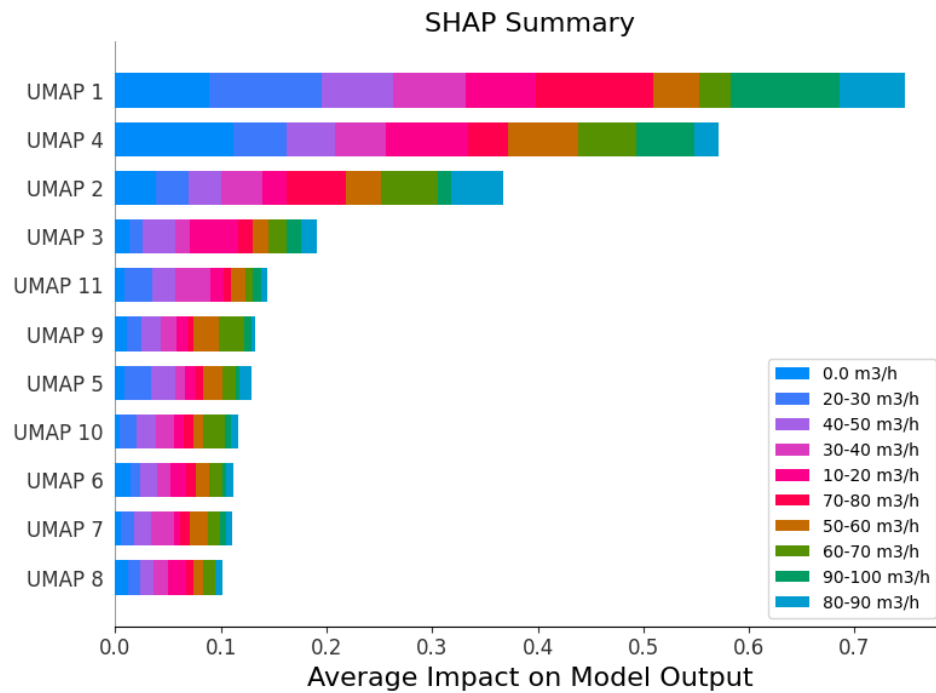


Figure 154 – XGBoost SHAP Values Summary - C

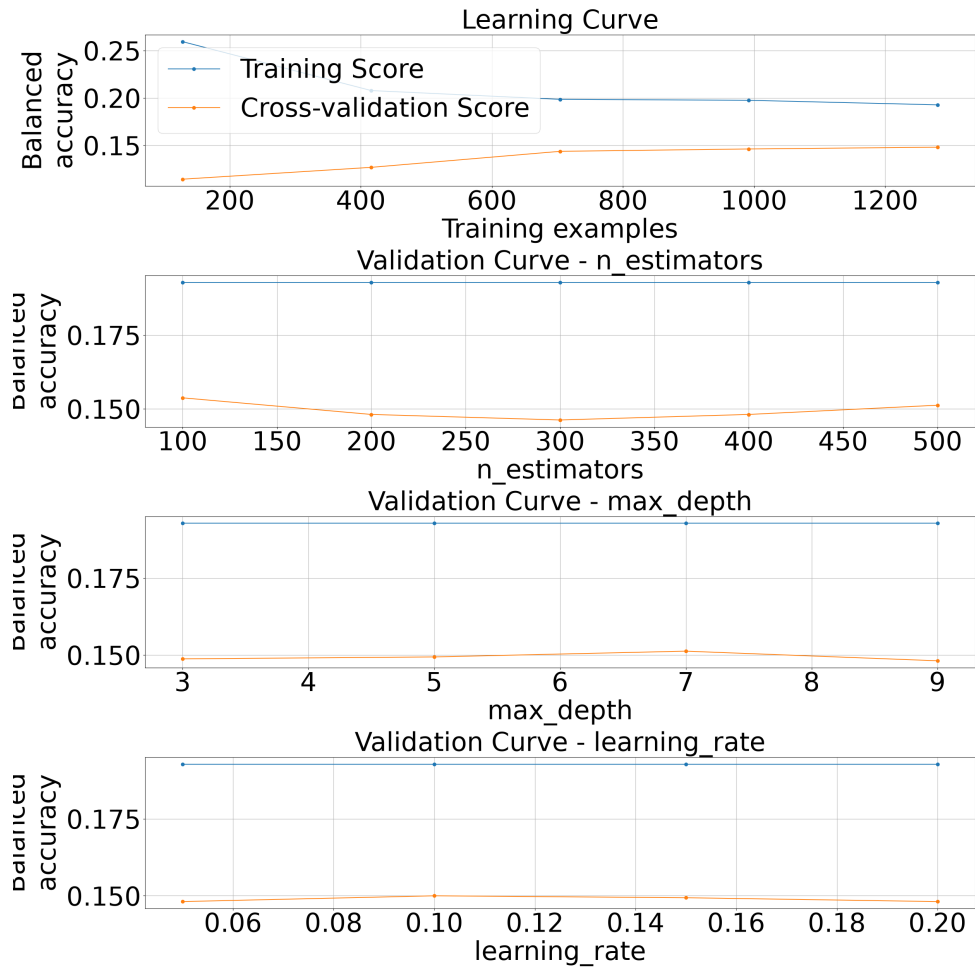


Figure 155 – XGBoost Learning & Validation Curves - C

Figure 156 presents the algorithm's ability to learn each individual class, from which is clear that the MLP algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 157 presents how much each projected feature contributes to the MLP classification process, the first feature had the most impact followed by the fourth, and third, while the second, seventh, eighth and eleventh had a least impact on the learning process. Finally, Figure 158 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

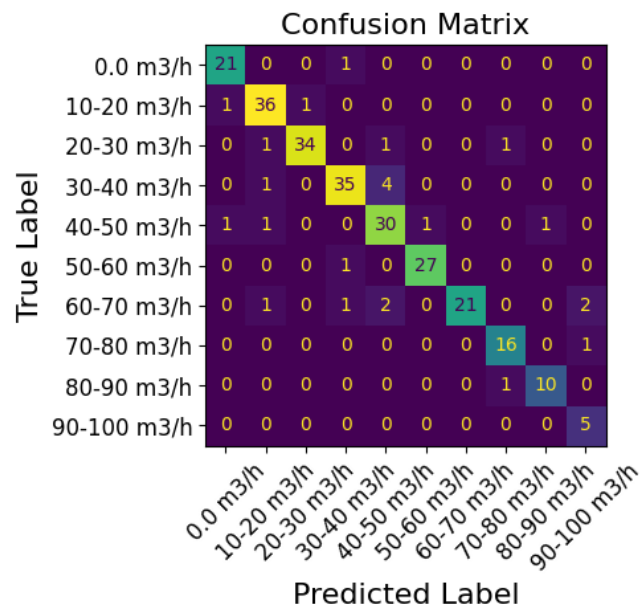


Figure 156 – MLP Confusion Matrix - C

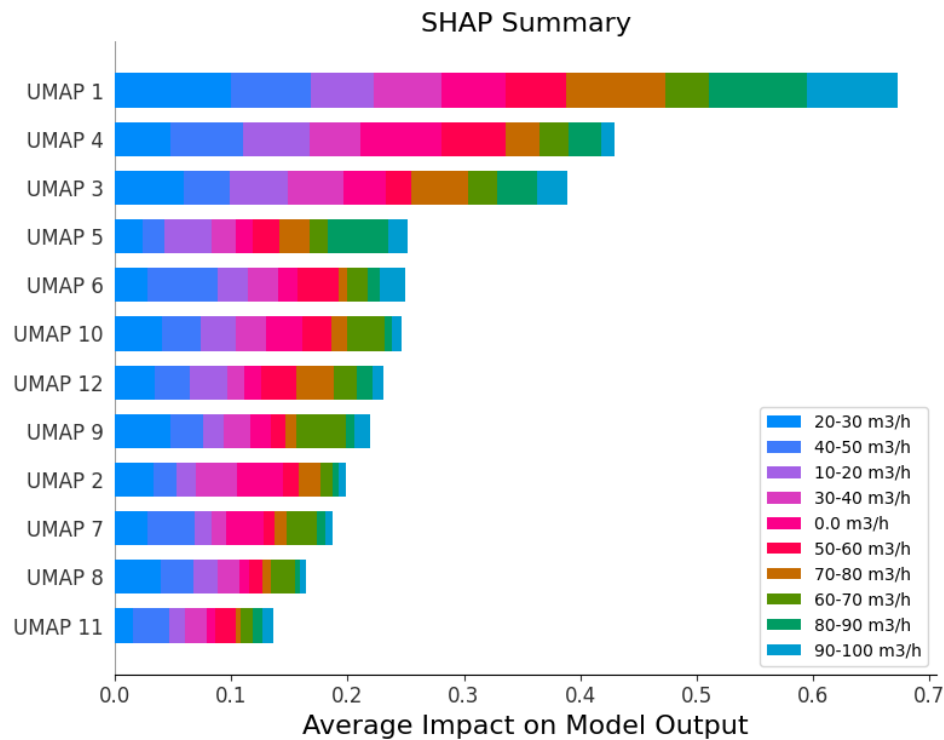


Figure 157 – MLP SHAP Values Summary - C

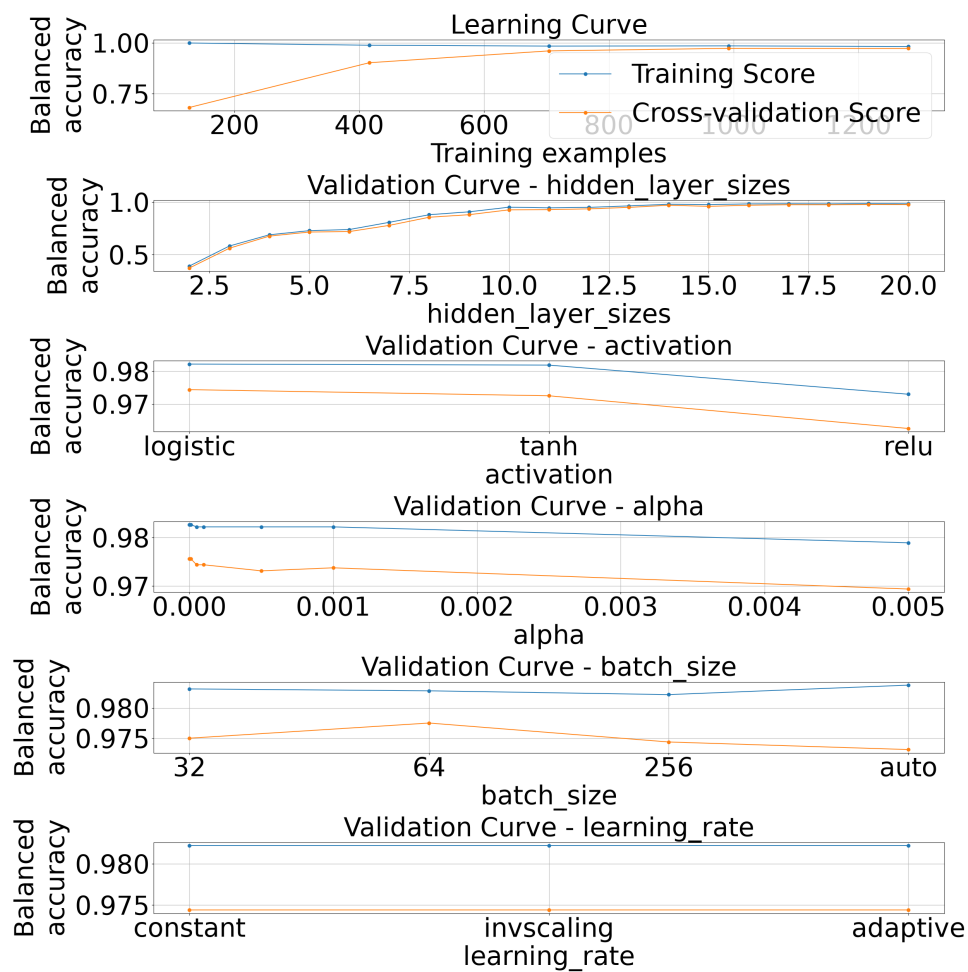


Figure 158 – MLP Learning & Validation Curves - C

	GMM	K-Means	HDBSCAN	Agglomerative	Spectral	Mean Shift
Silhouette	0.74	0.74	0.79	0.70	0.56	0.97
ARI	0.13	0.12	0.06	0.13	0.12	0.02
DBI	0.54	0.47	1.18	0.52	0.86	0.06
CH	1452.85	1989.1	1328.66	1180.36	248.87	105548.90

Table 11 – Unsupervised Learning Results - C

4.2.1.2 Unsupervised Learning

Table 11 summarize the overall performance of all tested unsupervised algorithms. The consistently low ARI scores indicate poor alignment between the discovered clusters and the ground-truth flow rate labels, suggesting that the algorithms were not able to recover the flow rate structure in the feature manifold. However, the remaining metrics show that several methods still identified a clear and coherent structure in the data. This structure is primarily associated with the rotational speed, which is an expected outcome since the dominant spectral and dynamical components of the motor signals are directly tied to shaft rotation and its harmonics. As a result, the extracted Koopman features naturally exhibit larger variance with respect to RPM than to flow variations, leading the algorithms to form well-defined RPM-driven clusters.

The table shows that GMM’s Silhouette score of 0.74, indicating that the algorithm formed well defined clusters – however, there’s still some mixture with the other metrics re-enforcing that. The low ARI score tells the formed clusters don’t align well with the true classes, which is further confirmed by its confusion matrix in Figure 159. Figure 160 presents how much each projected feature contributes to the GMM classification process, the first feature had the most impact followed by the third, and second, while the eighth, ninth and seventh had the least impact on the learning process.

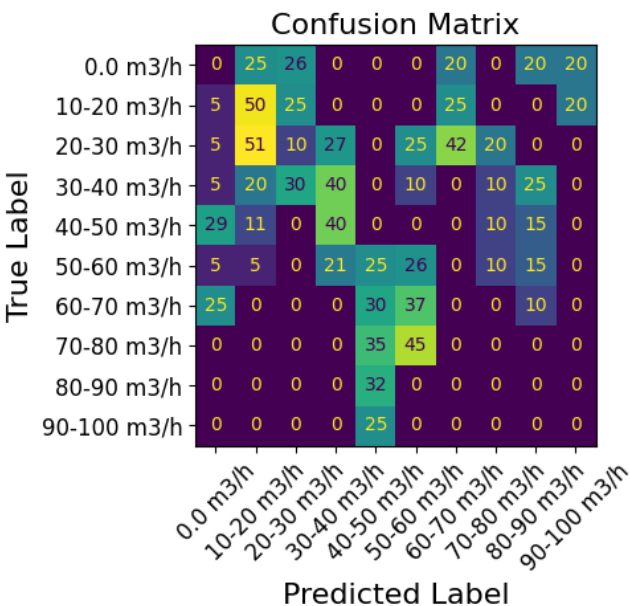


Figure 159 – GMM Confusion Matrix - C

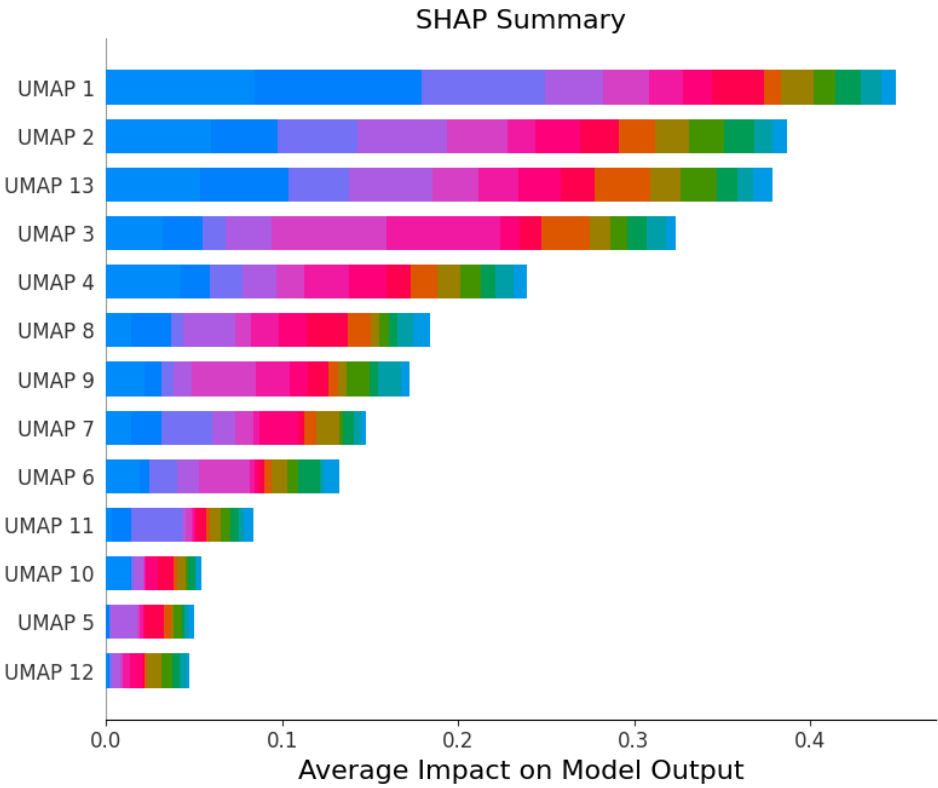


Figure 160 – GMM SHAP Values Summary - C

Kmeans reached a Silhouette score of 0.74, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 161. Figure 162 presents how much each projected feature contributes to the Kmeans classification process, the second feature had the most impact followed by the first, and second, while the seventh, fifth and sixth had the least impact on the learning process.

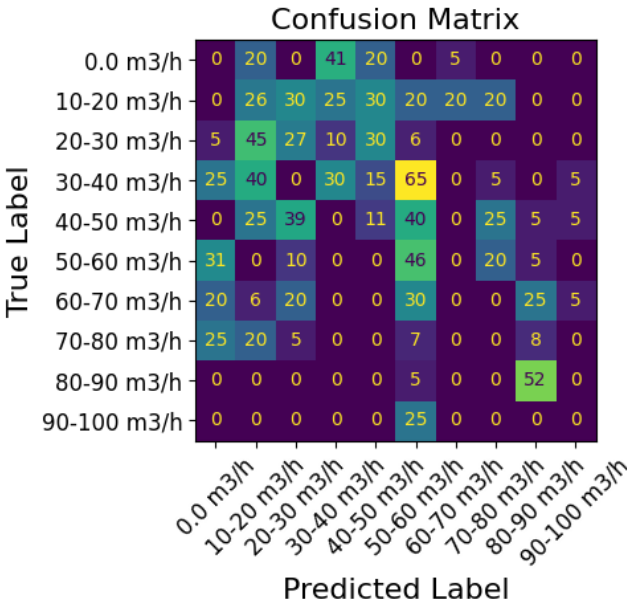


Figure 161 – KMeans Confusion Matrix - C

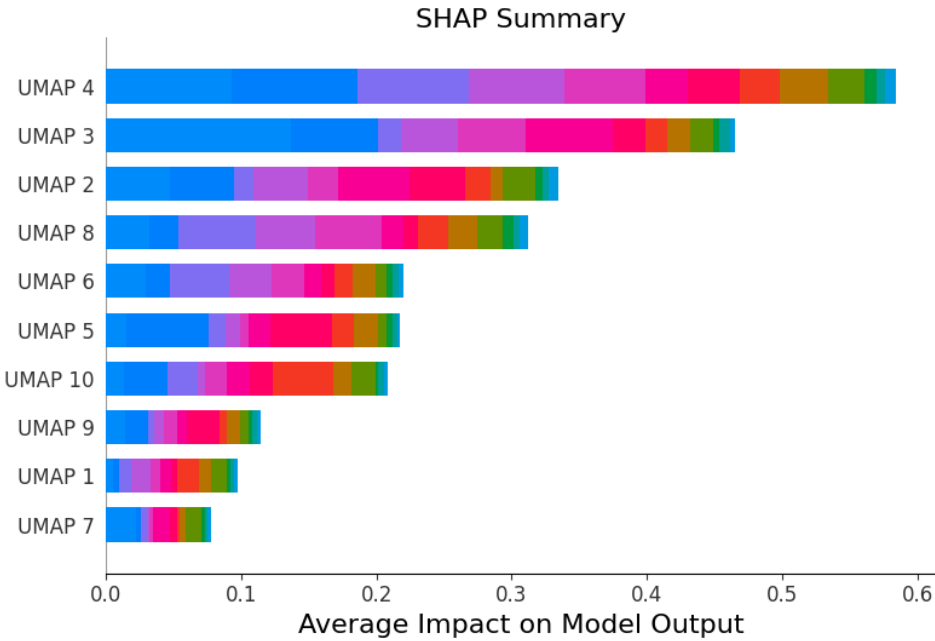


Figure 162 – KMeans SHAP Values Summary - C

HDBSCAN reached a Silhouette score of 0.79, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 163. Figure 164 presents how much each projected feature contributes to the HDBSCAN classification process, the optimization process picked only 2 projected features and both greatly contributed on the learning process.

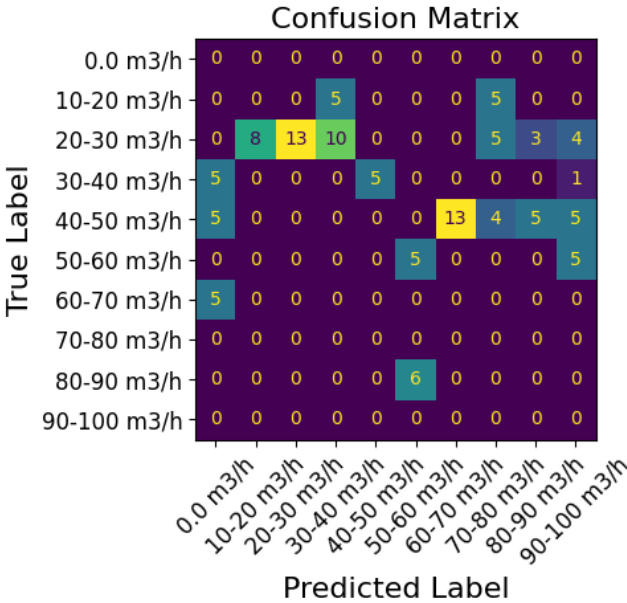


Figure 163 – HDBSCAN Confusion Matrix - C

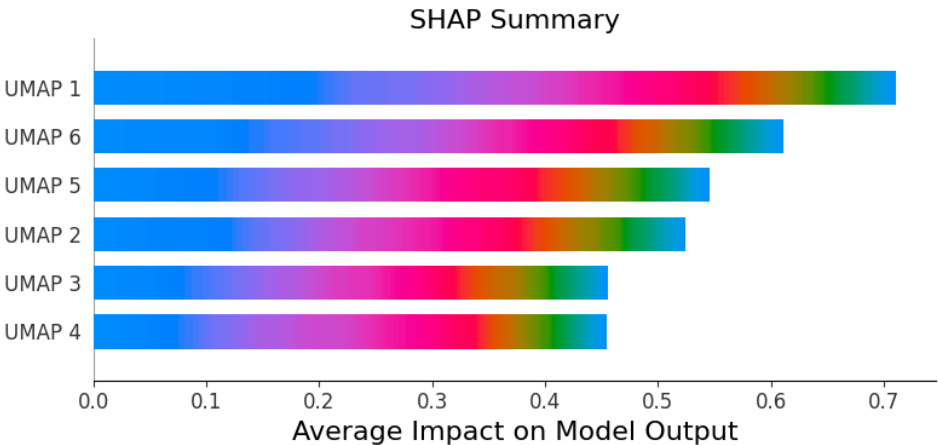


Figure 164 – HDBSCAN SHAP Values Summary - C

Agglomerative Clustering got a Silhouette score of 0.70, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 165. Figure 166 presents how much each projected feature contributes to the Agglomerative Clustering classification process, the first feature had the most impact followed by the second, and third, while the fourth and fifth had the least impact on the learning process.

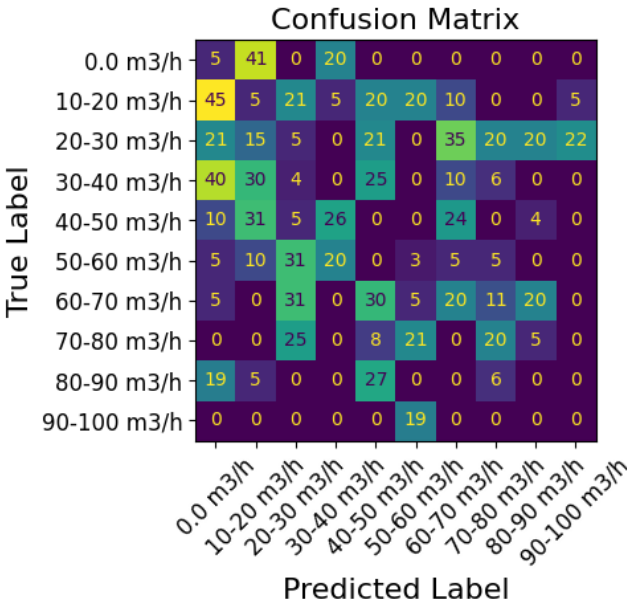


Figure 165 – Agglomerative Clustering Confusion Matrix - C

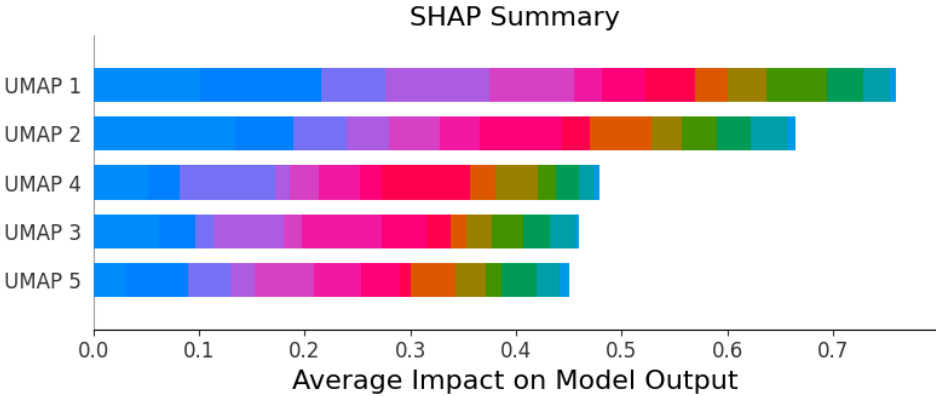


Figure 166 – Agglomerative Clustering SHAP Values Summary - C

The Spectral Clustering’s Silhouette score of 0.56, indicating that the algorithm that the algorithm had more trouble forming clusters, meaning that this method isn’t suited to cluster signals with this mixture, with other metrics re-enforcing that. The low ARI score tells the formed clusters don’t align well with the true classes, which is further confirmed by its confusion matrix in Figure 167. Figure 168 presents how much each projected feature contributes to the Spectral Clustering classification process, the eighth feature had the most impact followed by the first, and eleventh, while the fourth, tenth and fifth had the least impact on the learning process.

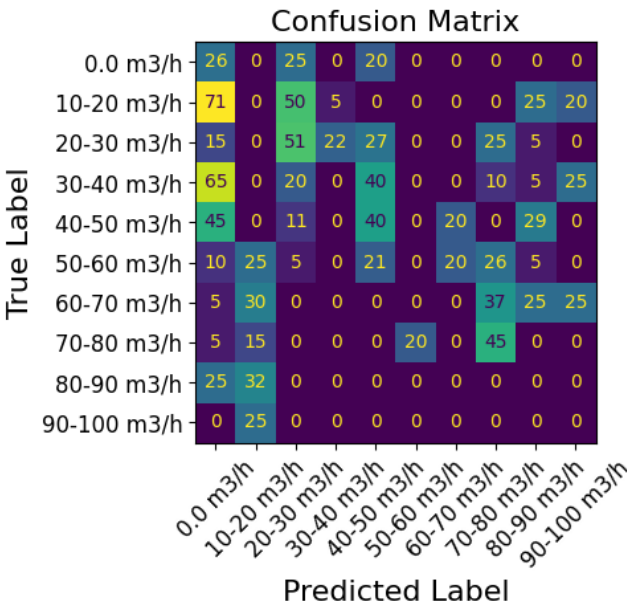


Figure 167 – Spectral Clustering Confusion Matrix - C

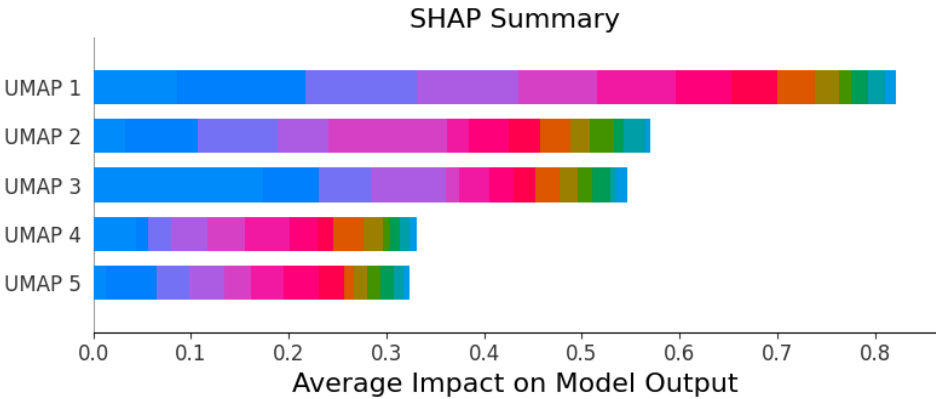


Figure 168 – Spectral Clustering SHAP Values Summary - C

Finally, the Mean Shift algorithm reached a Silhouette score of 0.97, indicating that the algorithm formed almost perfect clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in [Figure 169](#). [Figure 170](#) presents how much each projected feature contributes to the Mean Shift classification process, the optimization process picked only 2 projected features and both greatly contributed on the learning process.

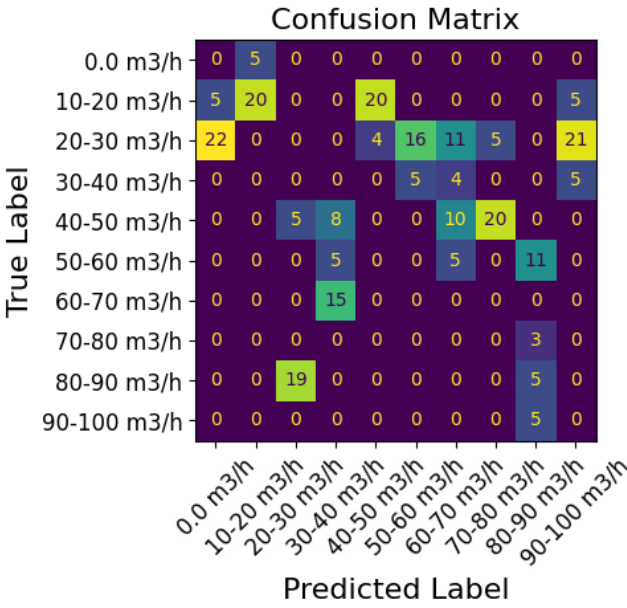


Figure 169 – Mean Shift Confusion Matrix - C

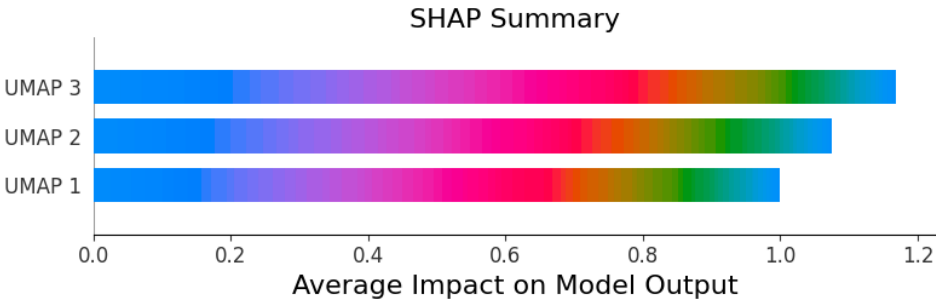


Figure 170 – Mean Shift SHAP Values Summary - C

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	85.5%	88.2%	83.4%	85.8%	80.7%	98.1%	89.4%
Test Accuracy	90.5%	89.2%	89.5%	79.4%	75.6%	85.4%	79%
F1 Score	0.91	0.90	0.90	0.76	0.75	0.83	0.76
ROC-AUC	0.98	0.97	0.98	0.94	0.93	0.93	0.92

Table 12 – Supervised Learning Results - V

4.2.2 Only 2 Vibrational Axes (V)

4.2.2.1 Supervised Pipeline

Table 12 summarize the overall performance of all tested supervised algorithms, showing that k-NN was the best performing algorithm followed by Random Forest, SVM, and XGBoost. The Logistic Regression and Gaussian Naive Bayes had lower accuracies due to the algorithm requirement being broken. Logistic Regression, for example, uses linear combinations of the inputs to classify the data, but if the data is nonlinear, then the algorithm will struggle more to learn the feature manifold and make correct classifications. Likewise, GNB assumes Gaussian distributed features so, if this assumption is broken, it would lead to a lower classification accuracy. However, the UMAP + SMOTE preprocessing step partially mitigates this limitations. By enforcing local metric linearity and introducing neighbourhood-based interpolations, the resulting feature space becomes more compatible with linear decision boundaries and Gaussian class-conditional modeling, allowing these algorithms achieve relatively high accuracies despite their theoretical limitations.

Figure 171 presents the algorithm's ability to learn each individual class, from which is clear that the k-NN algorithm was able to learn all classes quite well, having a little more trouble learning the "40-50 m^3/h " class, mistaking it for the "50-60 m^3/h ". Figure 172 presents how much each projected feature contributes to the k-NN classification process, the first feature had the most impact followed by the fifth, second, than third, while the ninth, eleventh and sixth were the ones with least impact but still having meaningful contribution for the classification. Finally, Figure 173 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

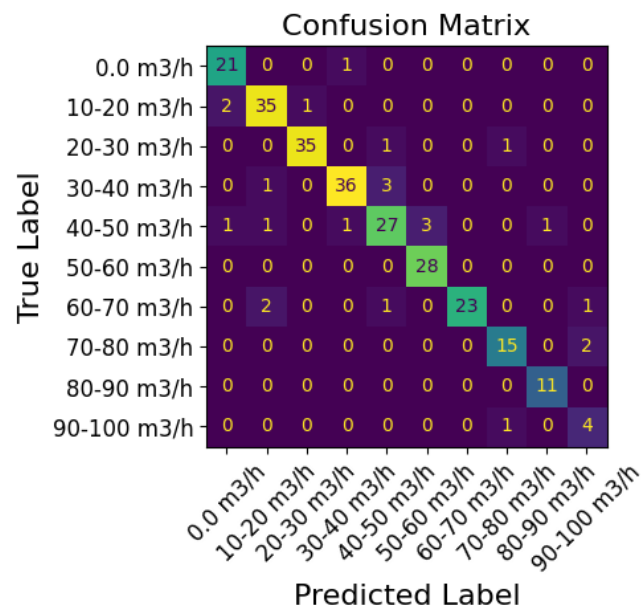


Figure 171 – k-NN Confusion Matrix - V

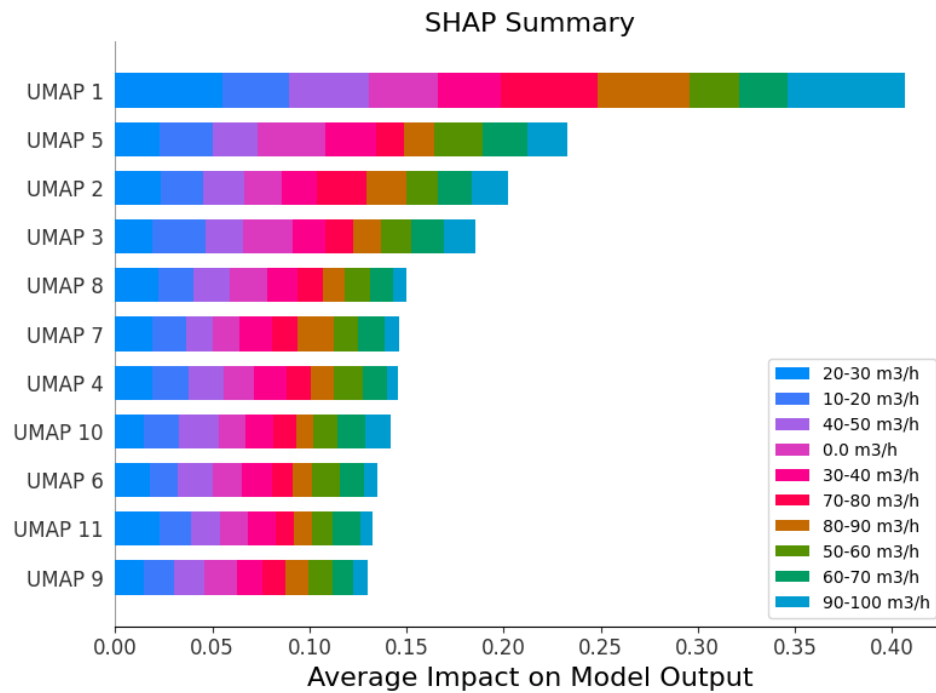


Figure 172 – k-NN SHAP Values Summary - V

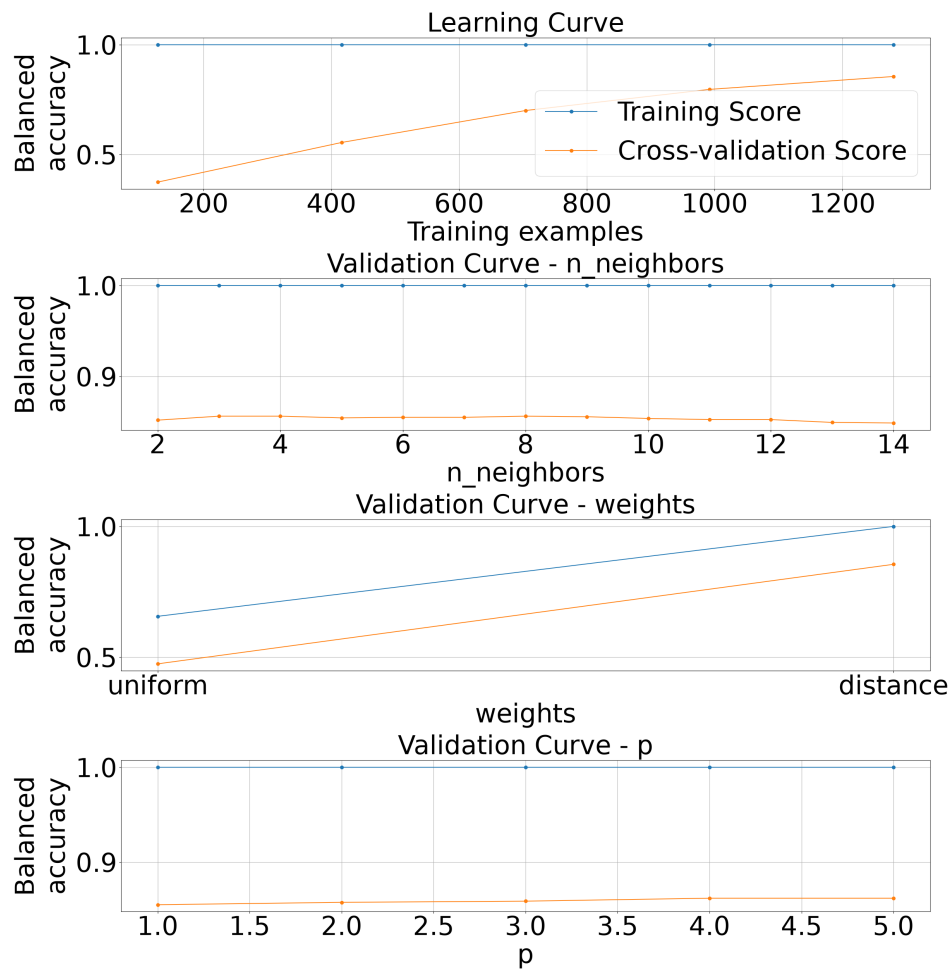


Figure 173 – k-NN Learning & Validation Curves - V

Figure 174 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "60-70 m^3/h " class. Figure 175 presents how much each projected feature contributes to the SVM classification process, the first feature had the most impact followed by the second, and fourth, while the eleventh, tenth and sixth were the ones with least impact but still having meaningful contribution for the classification. Finally, Figure 176 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

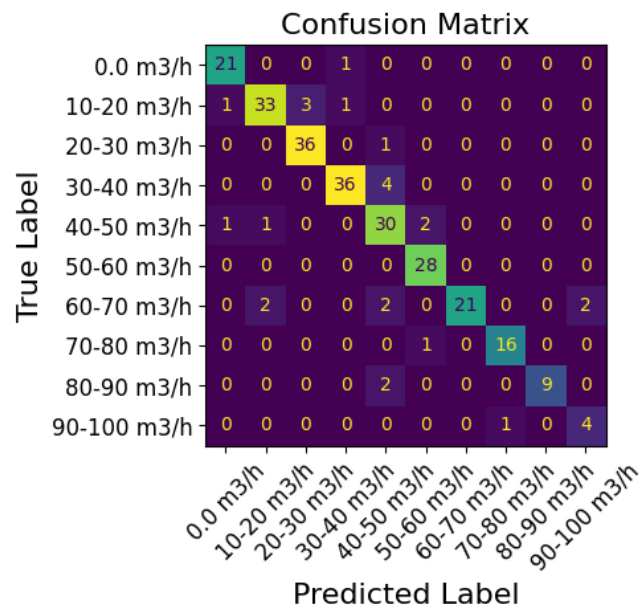


Figure 174 – SVM Confusion Matrix - V

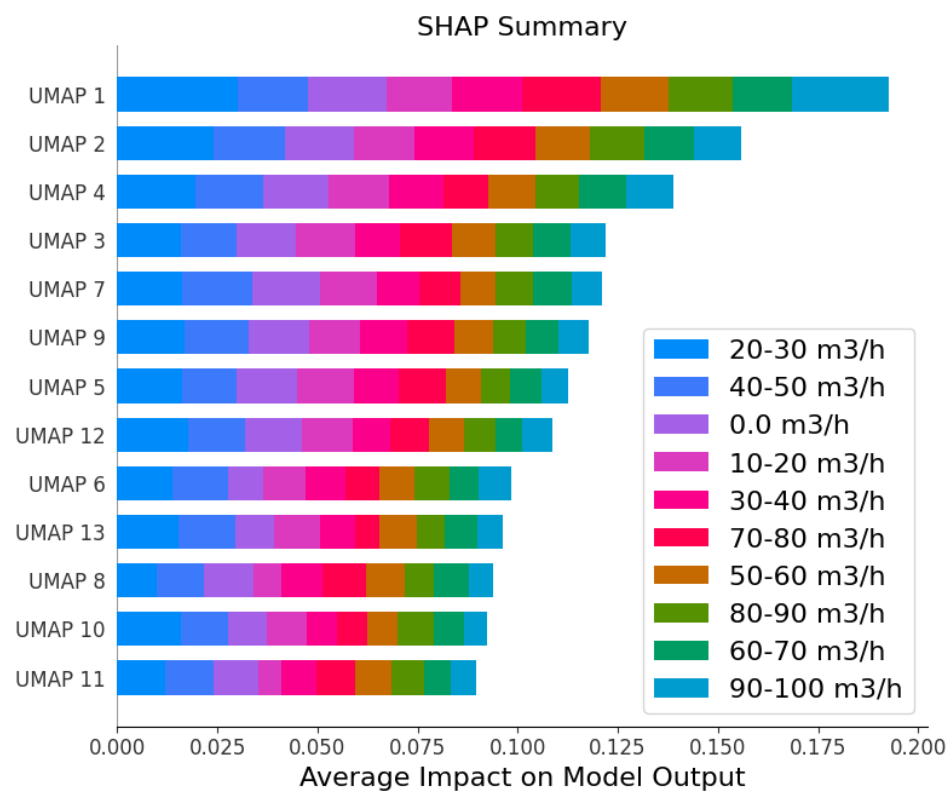


Figure 175 – SVM SHAP Values Summary - V

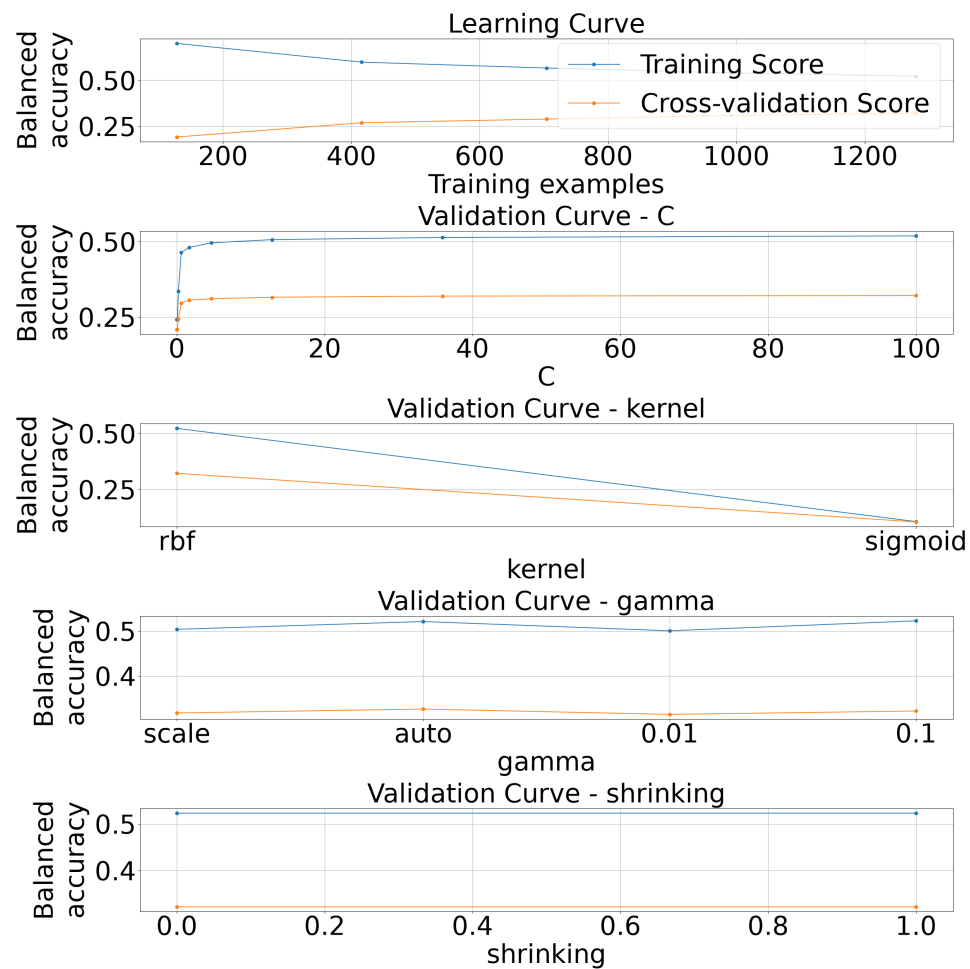


Figure 176 – SVM Learning & Validation Curves - V

Figure 177 presents the algorithm's ability to learn each individual class, from which is clear that the Random Forest algorithm was able to learn all classes quite well, having a little more trouble learning the classes "10-20 m^3/h " and "60-70 m^3/h ". Figure 178 presents how much each projected feature contributes to the Random Forest classification process, the first feature had the most impact followed by the third, second, and fourth, while the sixth, seventh and fifth were the ones with least impact but still having meaningful contribution for the classification. Finally, Figure 179 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

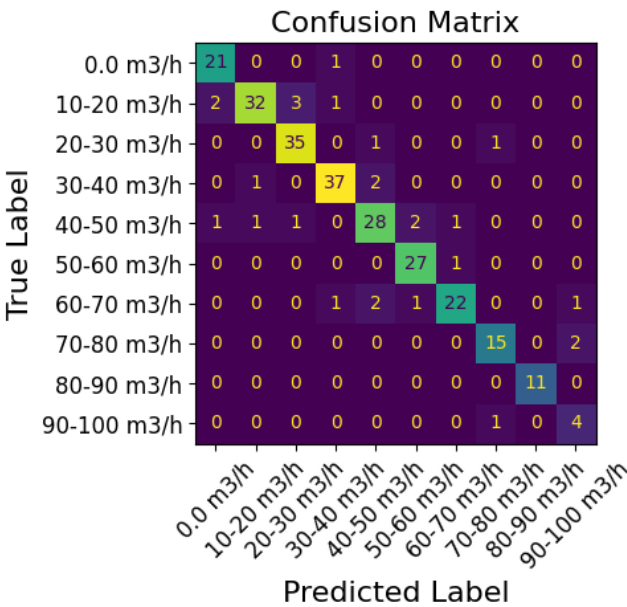


Figure 177 – Random Forest Confusion Matrix - V

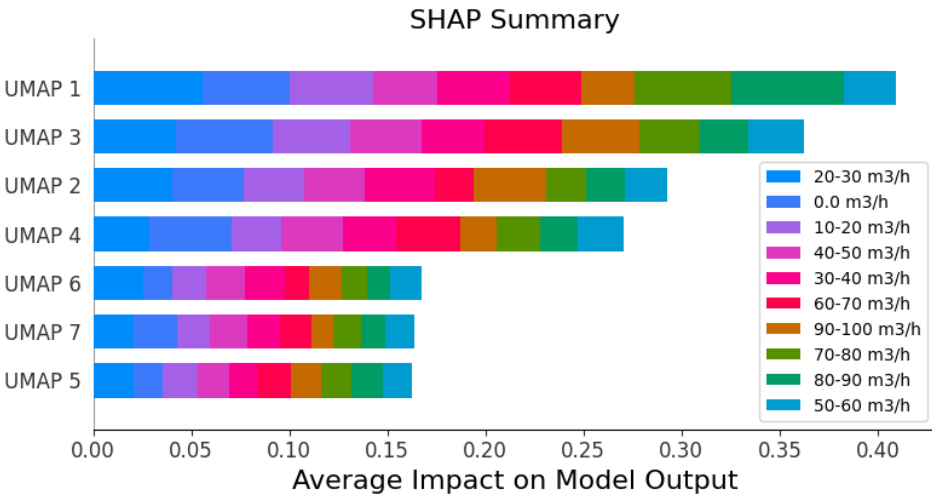


Figure 178 – Random Forest SHAP Values Summary - V

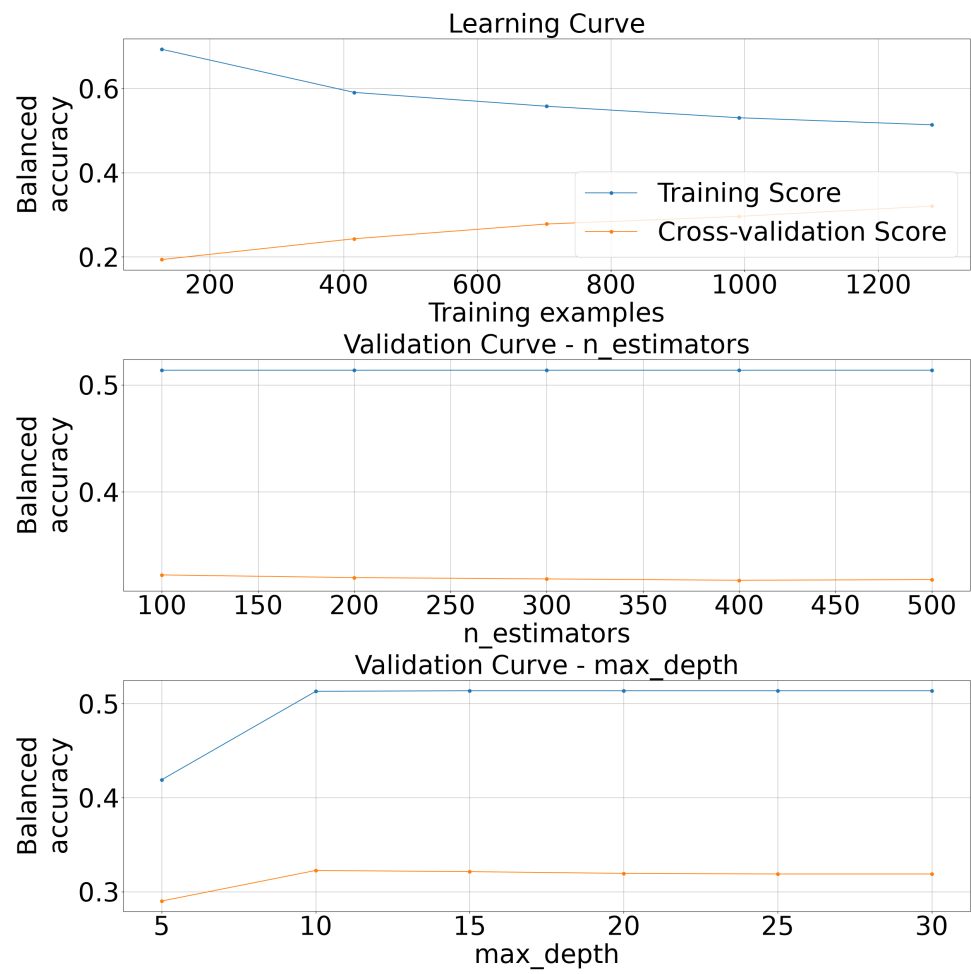


Figure 179 – Random Forest Learning & Validation Curves - V

Figure 180 presents the algorithm's ability to learn each individual class, from which is clear that the Logistic Regression algorithm was able to learn all classes quite well, having more trouble learning the classes "20-30 m^3/h " and "30-40 m^3/h ", both with a total of 11 misclassifications. Figure 181 presents how much each projected feature contributes to the Logistic Regression classification process, the first feature had the most impact followed by the fifth, second, third, and fourteenth with most impact, while the fourth, ninth, twelfth and thirteenth are the projected features with least impact on the learning process. Finally, Figure 182 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

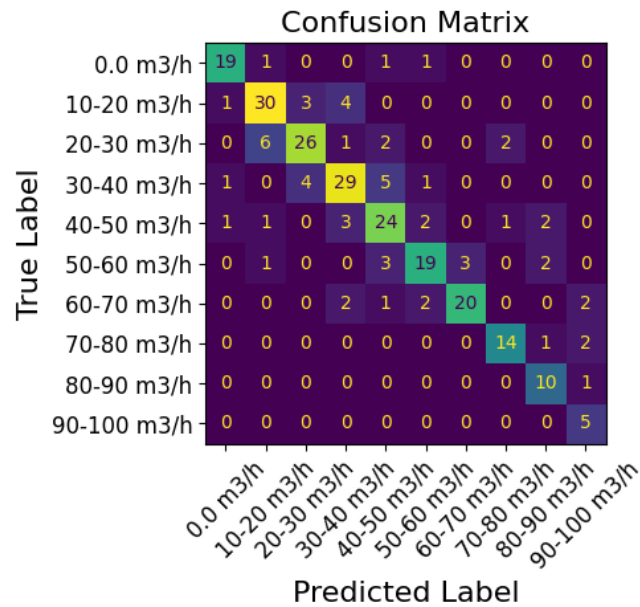


Figure 180 – Logistic Regression Confusion Matrix - V

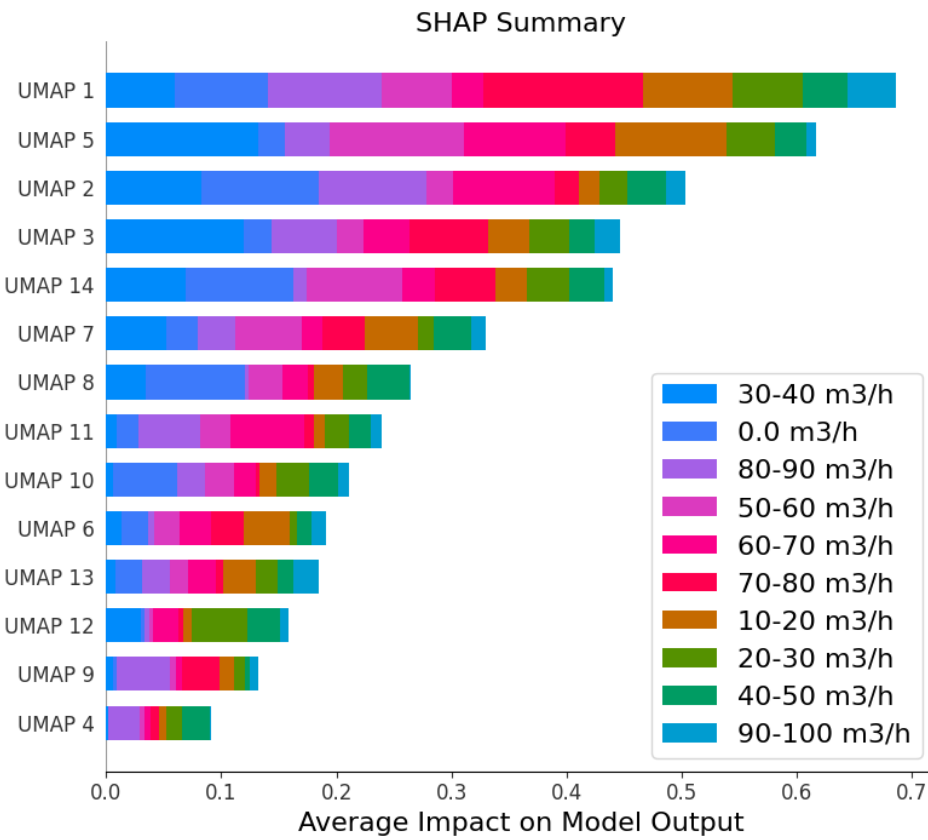


Figure 181 – Logistic Regression SHAP Values Summary - V

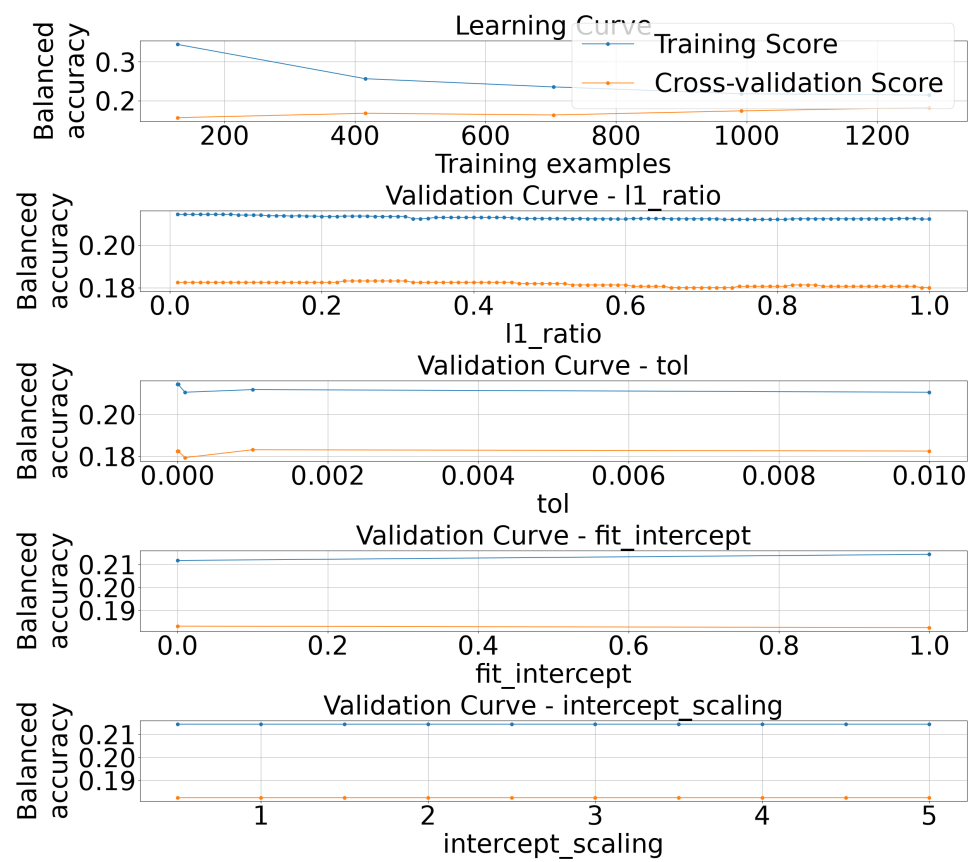


Figure 182 – Logistic Regression Learning & Validation Curves - V

Figure 183 presents the algorithm's ability to learn each individual class, from which is clear that the GNB algorithm was able to learn all classes quite well, having a little more trouble learning "20-30 m^3/h " class, mistaking it for the "10-20 m^3/h " class. Figure 184 presents how much each projected feature contributes to the Gaussian Naive Bayes classification process, the first feature had the most impact followed by the third, tenth and fourteenth, while the ninth, twelfth, eighth and eleventh projected features had the least impact on the learning process. Finally, Figure 185 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

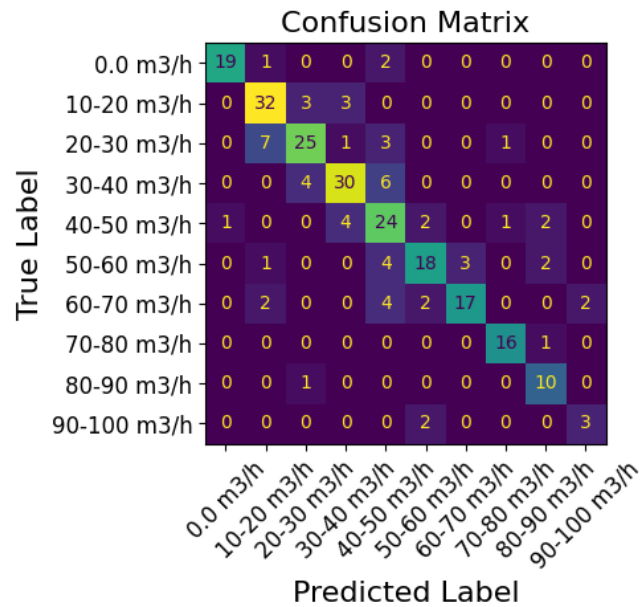


Figure 183 – GNB Confusion Matrix - V

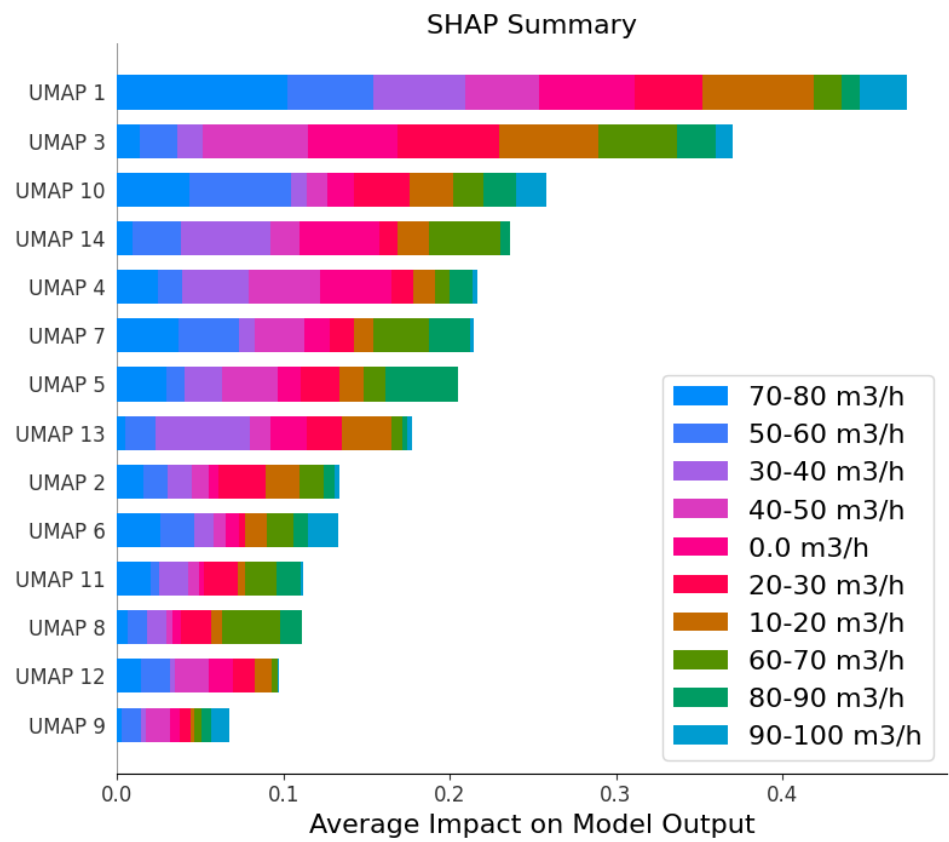


Figure 184 – GNB SHAP Values Summary - V

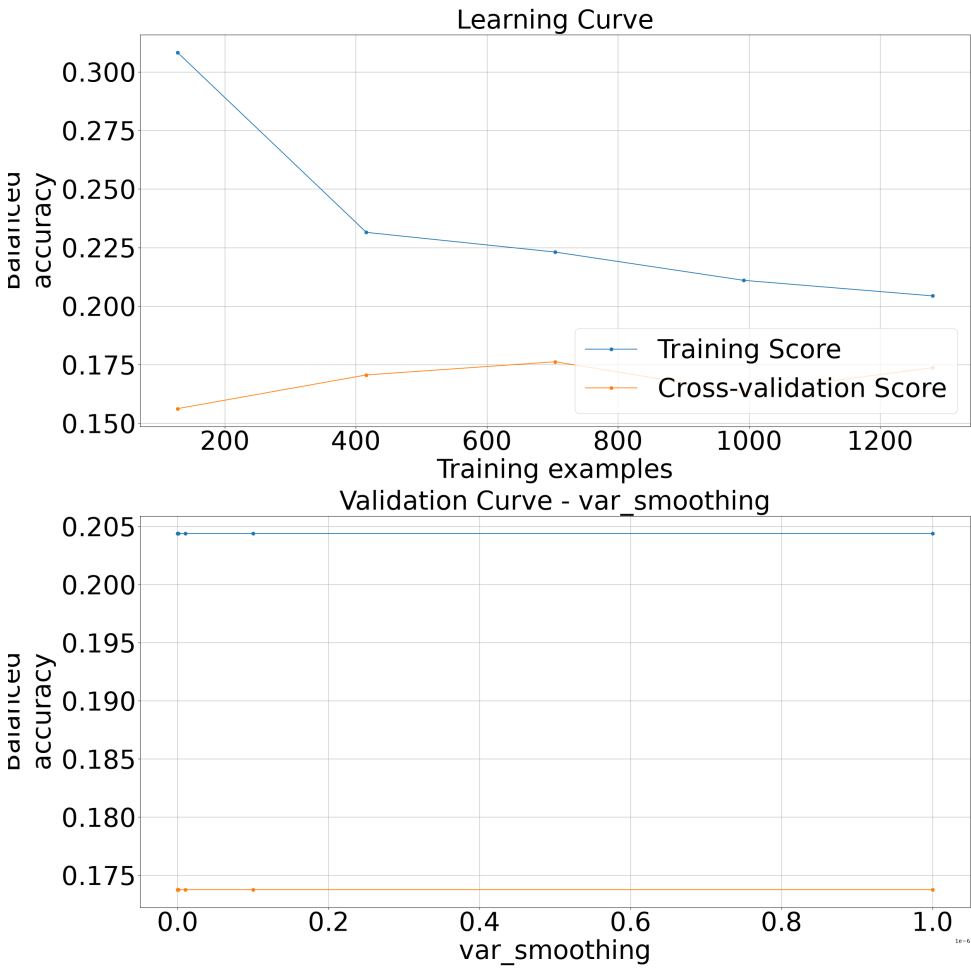


Figure 185 – GNB Learning & Validation Curves - V

Figure 186 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "40-50 m^3/h " class, mistaking it for the "50-60 m^3/h ". Figure 187 presents how much each projected feature contributes to the XGBoost classification process, the first feature had the most impact followed by the fourth, and second, while the tenth, sixth, seventh and eighth projected features had the least impact on the learning process. Finally, Figure 188 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

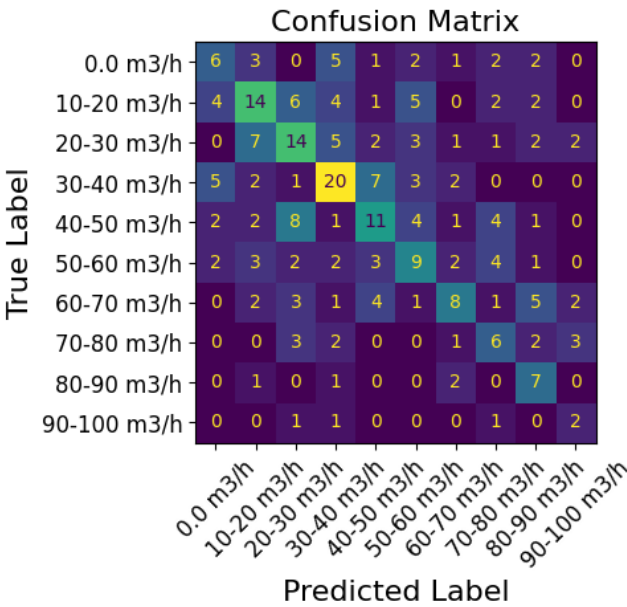


Figure 186 – XGBoost Confusion Matrix - V

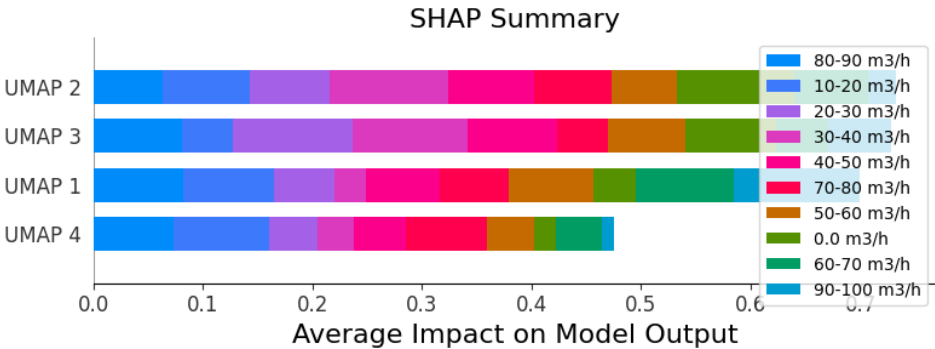


Figure 187 – XGBoost SHAP Values Summary - V

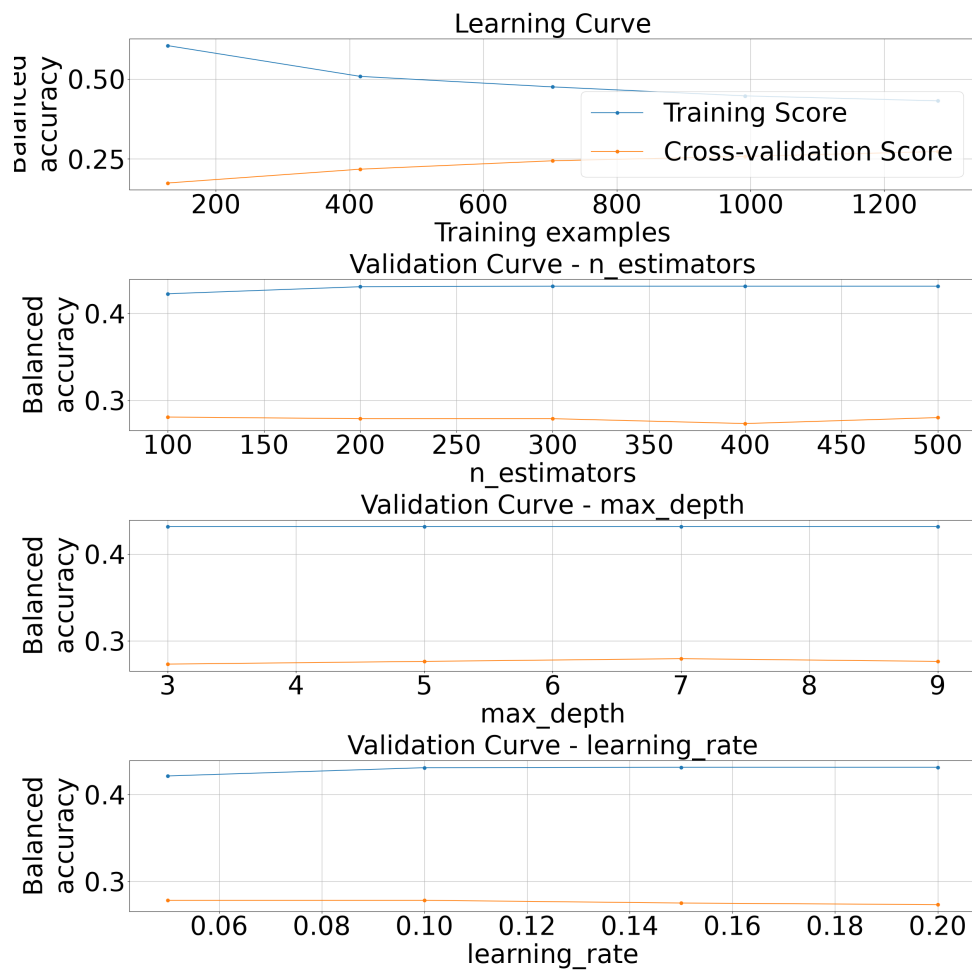


Figure 188 – XGBoost Learning & Validation Curves - V

Figure 189 presents the algorithm's ability to learn each individual class, from which is clear that the MLP algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "20-30 m^3/h ". Figure 190 presents how much each projected feature contributes to the MLP classification process, the eighth feature had the most impact followed by the sixth, first, and fourth, while the ninth, third, and tenth had a least impact on the learning process. Finally, Figure 191 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

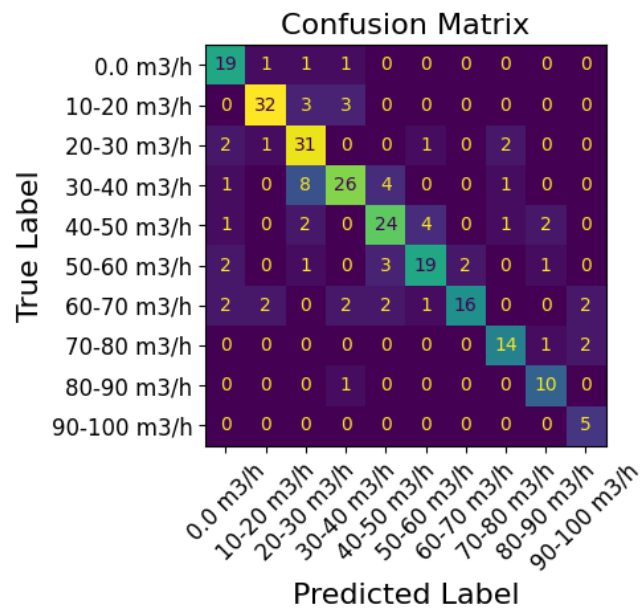


Figure 189 – MLP Confusion Matrix - V

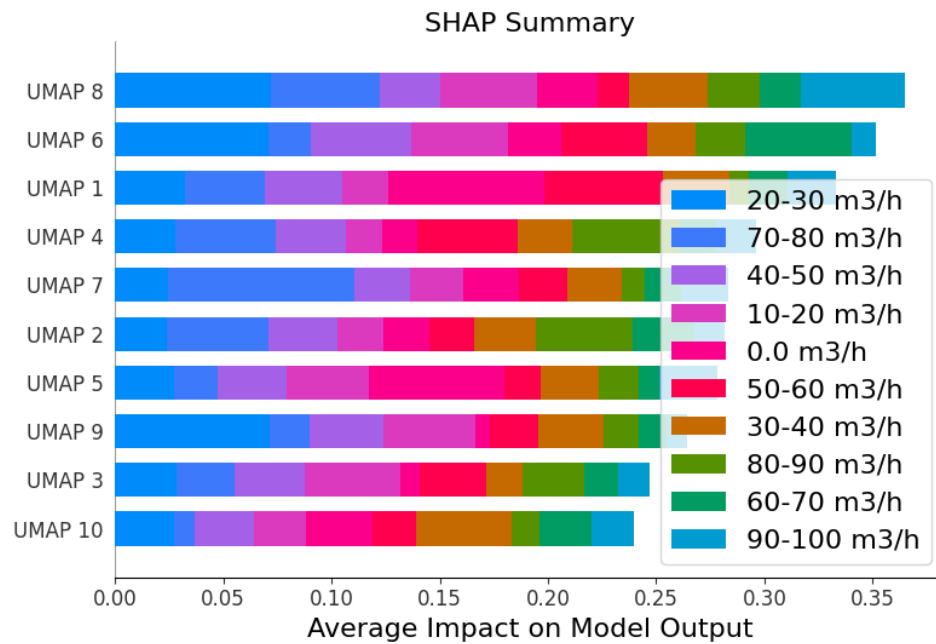


Figure 190 – MLP SHAP Values Summary - V

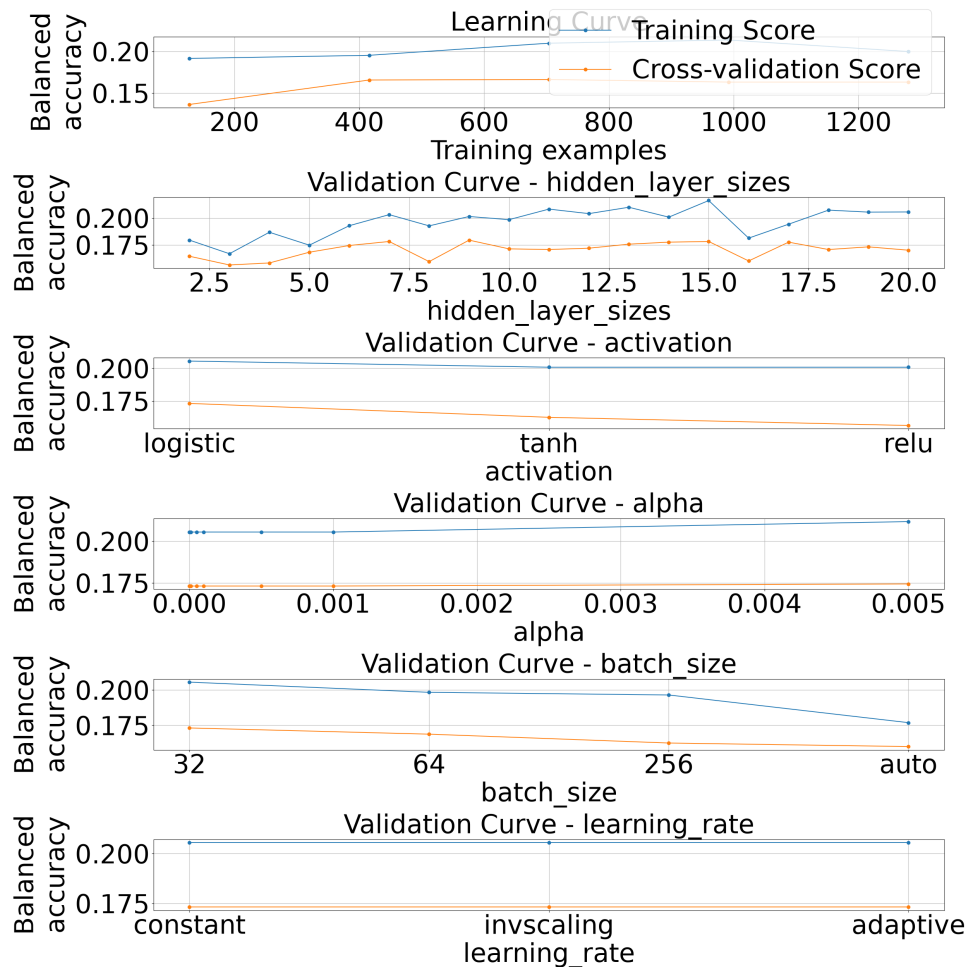


Figure 191 – MLP Learning & Validation Curves - V

	GMM	K-Means	HDBSCAN	Agglomerative	Spectral	Mean Shift
Silhouette	0.60	0.53	0.68	0.65	-0.06	0.96
ARI	0.13	0.11	0.06	0.11	0.07	0.02
DBI	0.64	0.78	1.02	0.53	7.1	0.05
CH	623.21	423.18	137.6	700.92	18.8	36170.85

Table 13 – Unsupervised Learning Results - V

4.2.2.2 Unsupervised Learning

Table 13 summarize the overall performance of all tested unsupervised algorithms. The consistently low ARI scores indicate poor alignment between the discovered clusters and the ground-truth flow rate labels, suggesting that the algorithms were not able to recover the flow rate structure in the feature manifold. However, the remaining metrics show that several methods still identified a clear and coherent structure in the data. This structure is primarily associated with the rotational speed, which is an expected outcome since the dominant spectral and dynamical components of the motor signals are directly tied to shaft rotation and its harmonics. As a result, the extracted Koopman features naturally exhibit larger variance with respect to RPM than to flow variations, leading the algorithms to form well-defined RPM-driven clusters.

The table shows that GMM’s Silhouette score of 0.60, indicating that the algorithm formed well defined clusters – however, there’s still some mixture with the other metrics re-enforcing that. The low ARI score tells the formed clusters don’t align well with the true classes, which is further confirmed by its confusion matrix in [Figure 192](#). [Figure 193](#) presents how much each projected feature contributes to the GMM classification process, the second feature had the most impact followed by the first, and sixth, while the eighth, fourth and fifth had the least impact on the learning process.

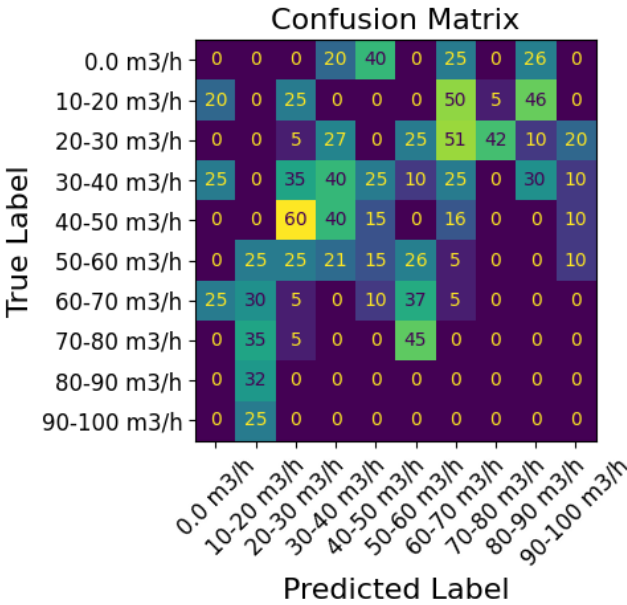


Figure 192 – GMM Confusion Matrix - V

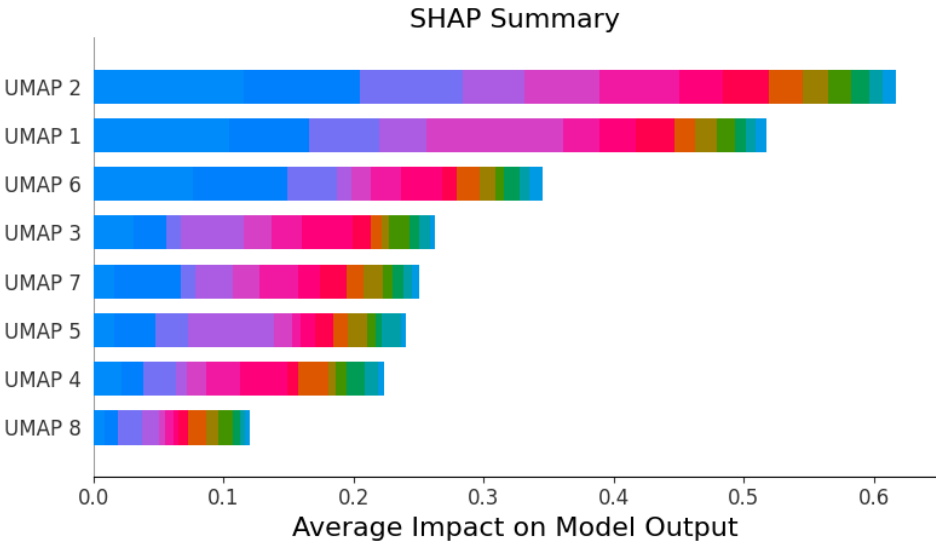


Figure 193 – GMM SHAP Values Summary - V

Kmeans reached a Silhouette score of 0.53, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 194. Figure 195 presents how much each projected feature contributes to the Kmeans classification process, the first feature had the most impact followed by the second, and fifth, while the third, fourth and seventh had the least impact on the learning process.

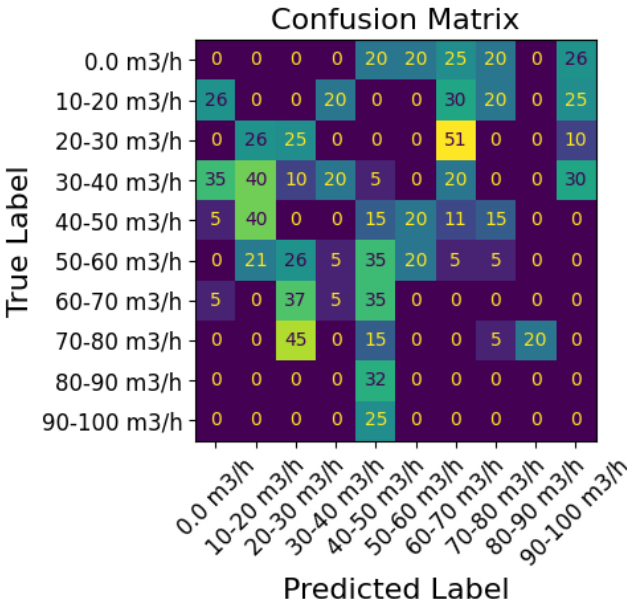


Figure 194 – KMeans Confusion Matrix - V

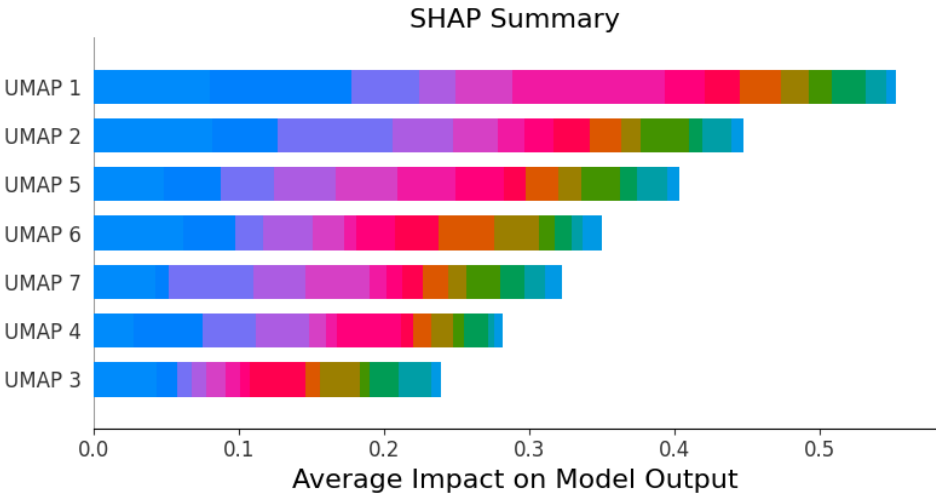


Figure 195 – KMeans SHAP Values Summary - V

HDBSCAN reached a Silhouette score of 0.68, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 196. Figure 197 presents how much each projected feature contributes to the HDBSCAN classification process, the third feature had the most impact followed by the second, first, and fourth, with all having a great impact in the learning process.

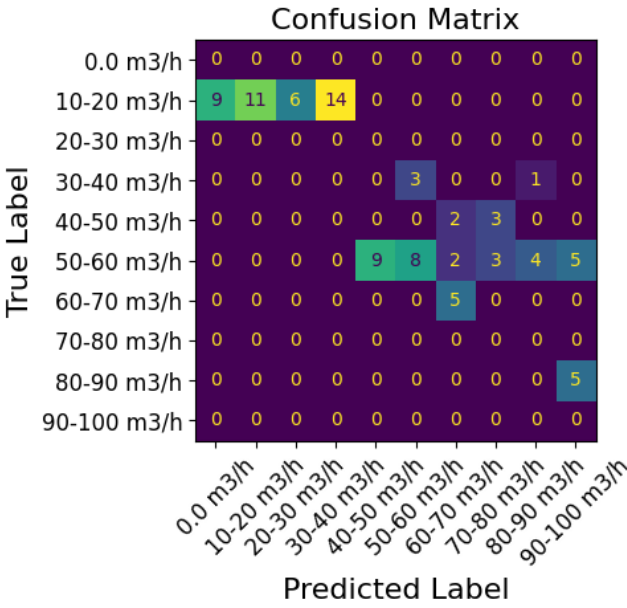


Figure 196 – HDBSCAN Confusion Matrix - V

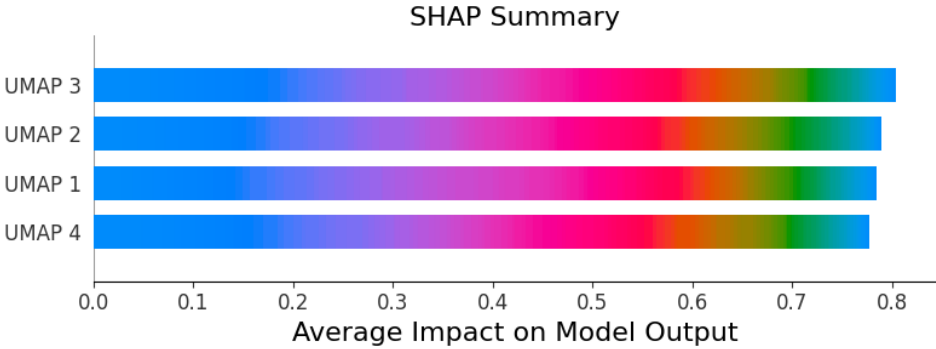


Figure 197 – HDBSCAN SHAP Values Summary - V

Agglomerative Clustering got a Silhouette score of 0.65, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 198. Figure 199 presents how much each projected feature contributes to the Agglomerative Clustering classification process, the first feature had the most impact followed by the second, and third, while the fifth, ninth, thirteenth, and eleventh had the least impact on the learning process.

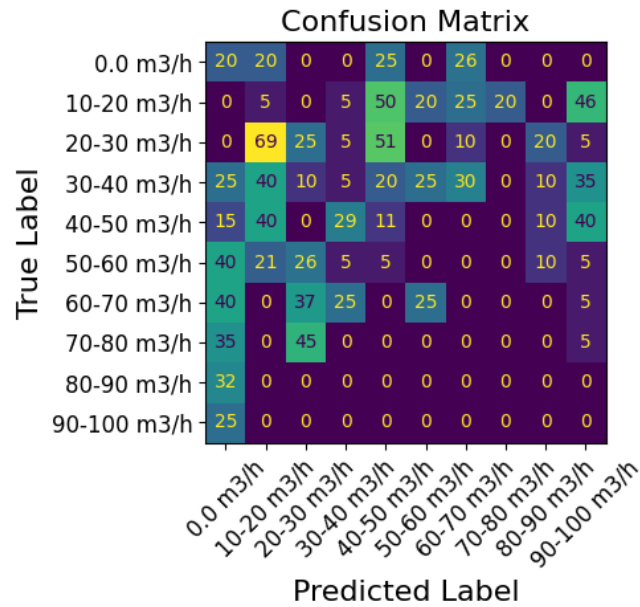


Figure 198 – Agglomerative Clustering Confusion Matrix - V

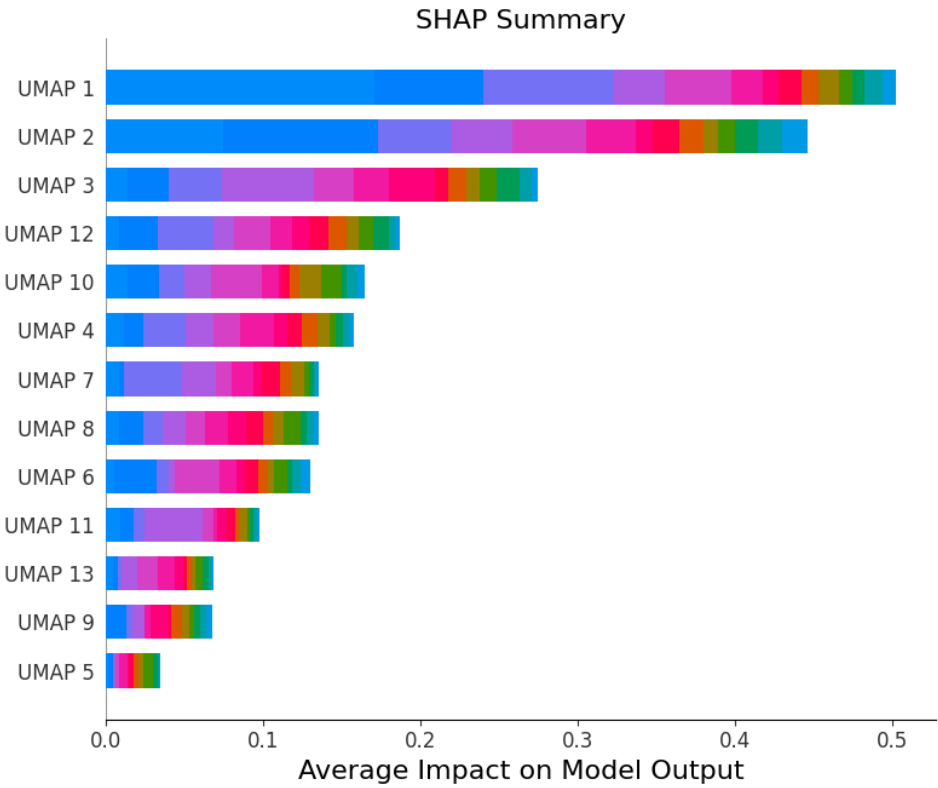


Figure 199 – Agglomerative Clustering SHAP Values Summary - V

The Spectral Clustering’s Silhouette score of -0.06, indicating that the algorithm that the algorithm barely could form clusters, meaning that this method isn’t suited to cluster signals with this mixture, with other metrics re-enforcing that. The low ARI score tells the formed clusters don’t align well with the true classes, which is further confirmed by its confusion matrix in [Figure 200](#). [Figure 201](#) presents how much each projected feature contributes to the Spectral Clustering classification process, the fourth feature had the most impact followed by the second, third, and first, with all having a great impact on the learning process.

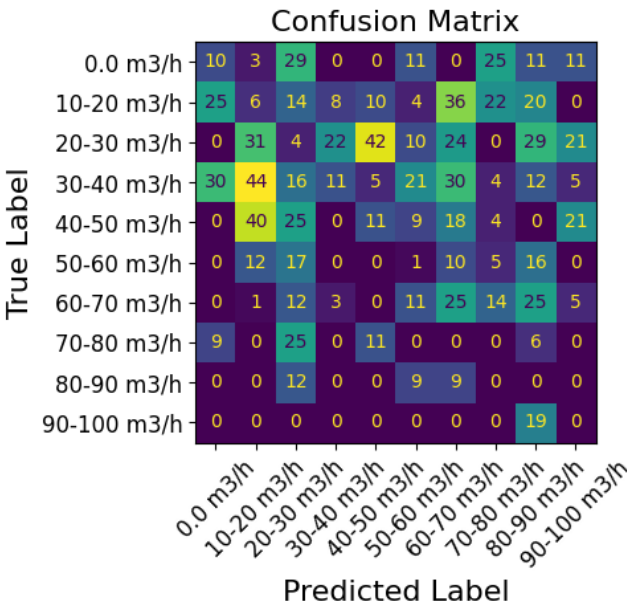


Figure 200 – Spectral Clustering Confusion Matrix - V

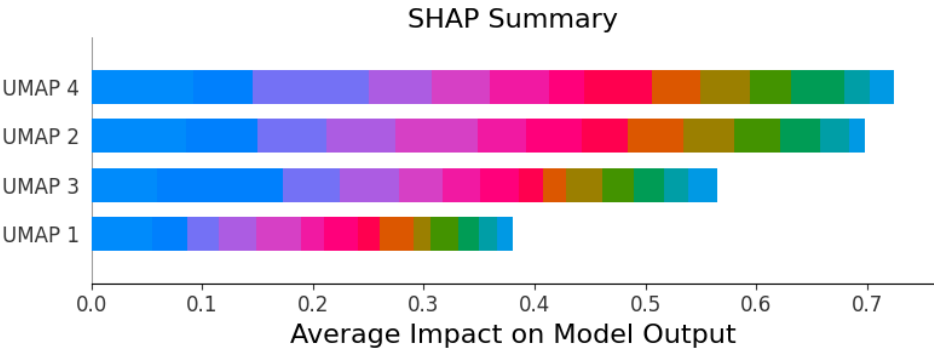


Figure 201 – Spectral Clustering SHAP Values Summary - V

Finally, the Mean Shift algorithm reached a Silhouette score of 0.96, indicating that the algorithm formed almost perfect clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in [Figure 202](#). [Figure 203](#) presents how much each projected feature contributes to the Mean Shift classification process, the third feature had the most impact followed by the second, first, with all having a great impact in the learning process.

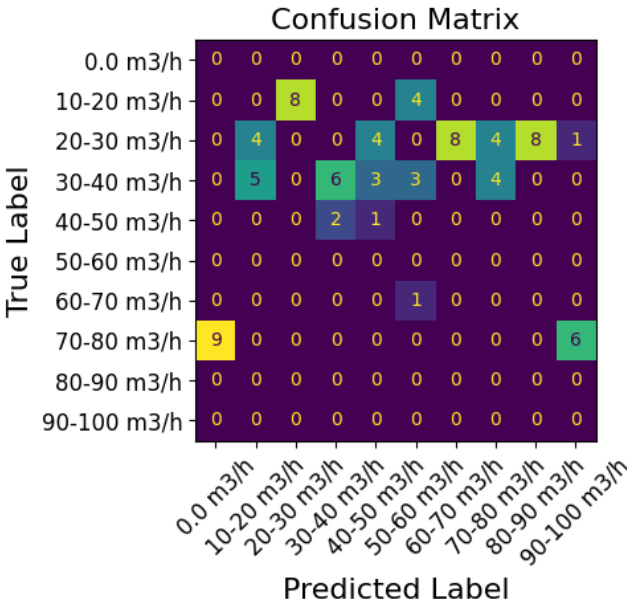


Figure 202 – Mean Shift Confusion Matrix - V

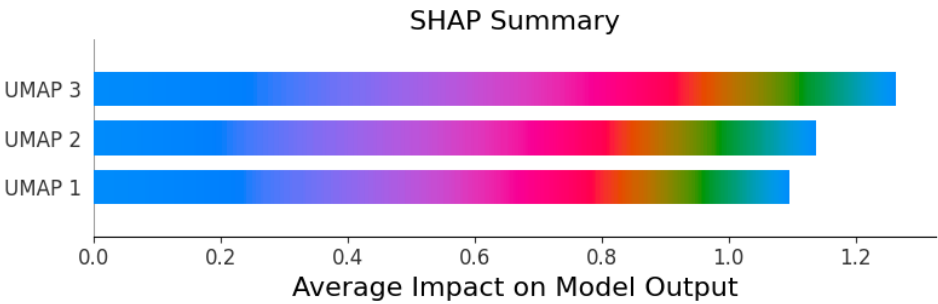


Figure 203 – Mean Shift SHAP Values Summary - V

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	97.8%	94.3%	91.3%	71.1%	85.6%	97.9%	95.8%
Test Accuracy	91.5%	92.6%	91.7%	67.7%	71.2%	81.5%	83.1%
F1 Score	0.92	0.92	0.92	0.87	0.81	0.82	0.80
ROC-AUC	0.97	0.96	0.95	0.80	0.84	0.85	0.96

Table 14 – Supervised Learning Results - CV

4.2.3 3 Currents + 2 Vibrational Axes (CV)

4.2.3.1 Supervised Pipeline

Table 14 summarize the overall performance of all tested supervised algorithms, showing that MLP was the best performing algorithm followed by XGBoost, k-NN SVM, MLP and Random Forest. The Logistic Regression and Gaussian Naive Bayes had lower accuracies due to the algorithm requirement being broken. Logistic Regression, for example, uses linear combinations of the inputs to classify the data, but if the data is nonlinear, then the algorithm will struggle more to learn the feature manifold and make correct classifications. Likewise, GNB assumes Gaussian distributed features so, if this assumption is broken, it would lead to a lower classification accuracy. However, the UMAP + SMOTE preprocessing step partially mitigates this limitations. By enforcing local metric linearity and introducing neighbourhood-based interpolations, the resulting feature space becomes more compatible with linear decision boundaries and Gaussian class-conditional modeling, allowing these algorithms achieve relatively high accuracies despite their theoretical limitations.

Figure 204 presents the algorithm's ability to learn each individual class, from which is clear that the k-NN algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 205 presents how much each projected feature contributes to the k-NN classification process, the first feature had the most impact followed by the fourth, then second, the tenth and fifth were the ones with least impact but still having meaningful contribution for the classification. Finally, Figure 206 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

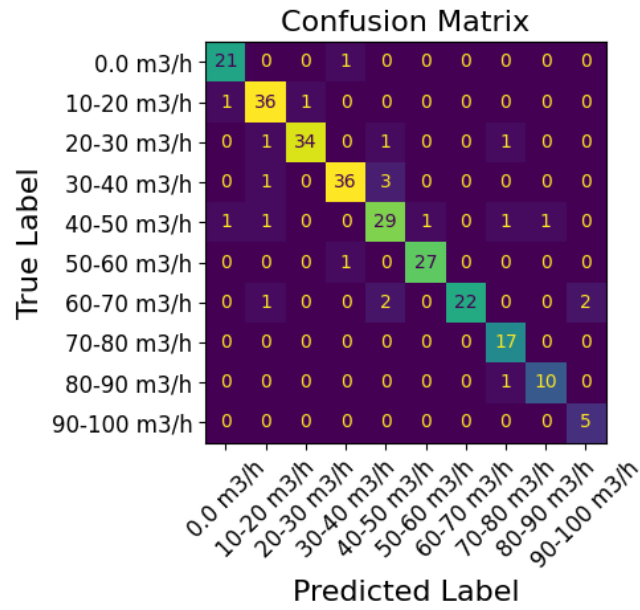


Figure 204 – k-NN Confusion Matrix - CV

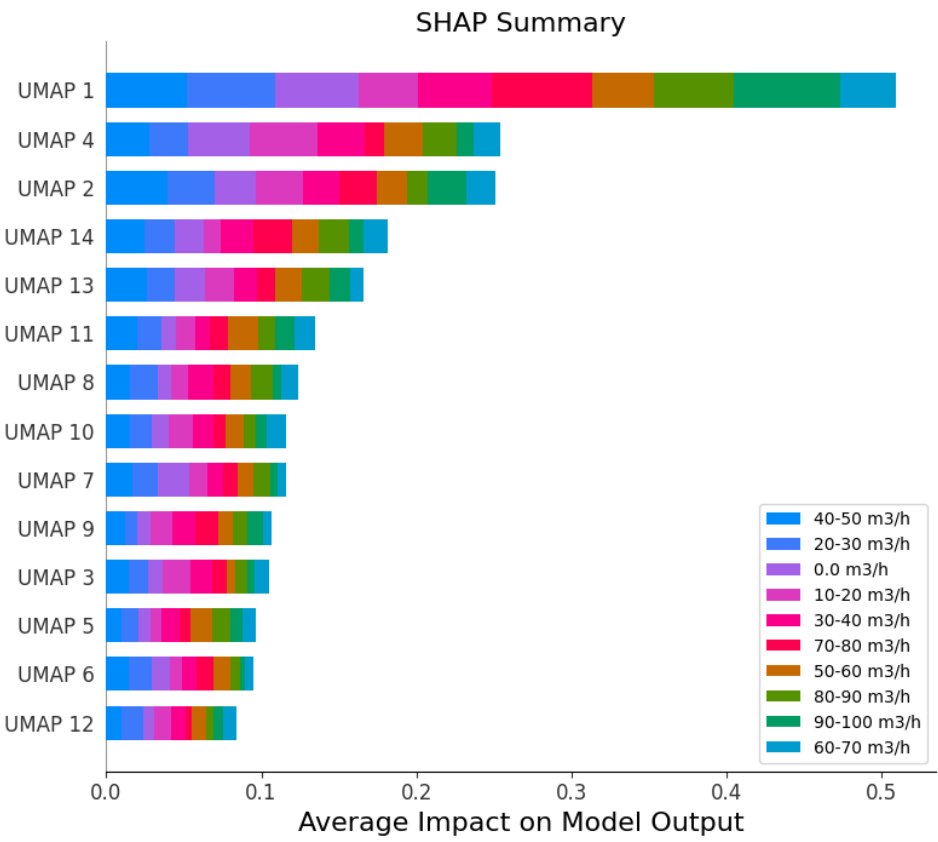


Figure 205 – k-NN SHAP Values Summary - CV

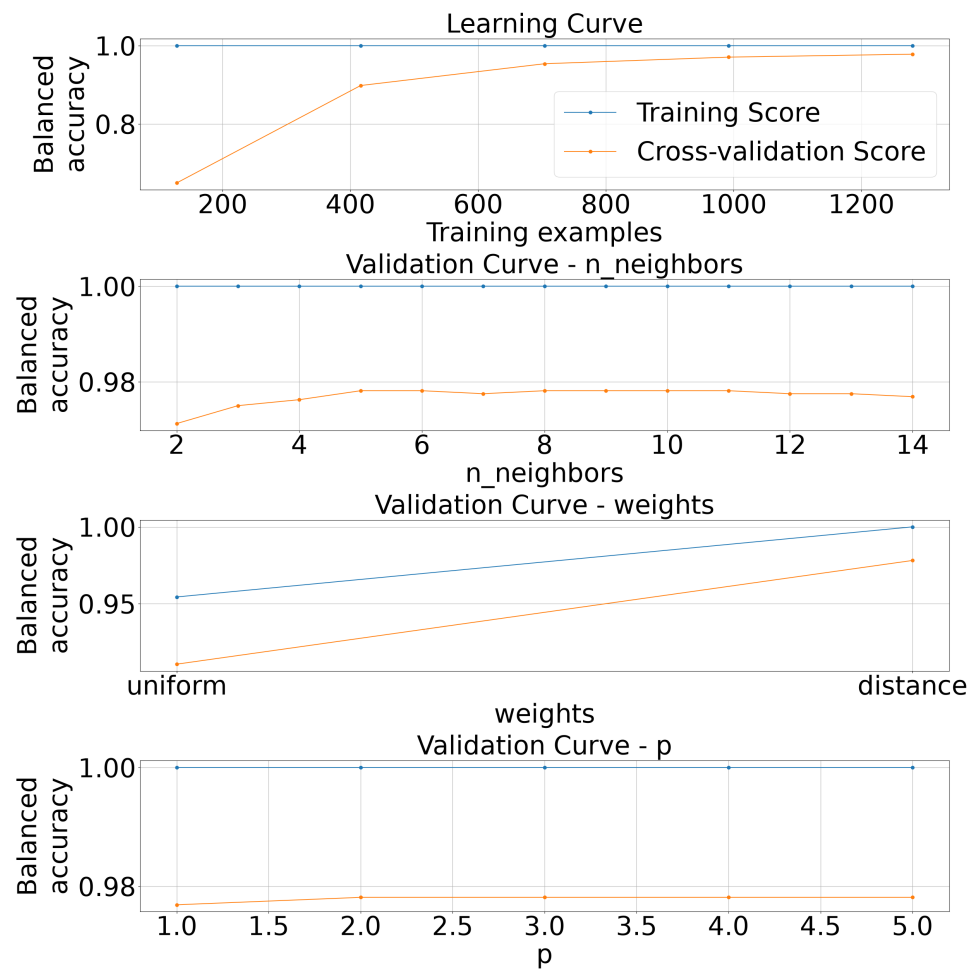


Figure 206 – k-NN Learning & Validation Curves - CV

Figure 207 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 208 presents how much each projected feature contributes to the SVM classification process, the first feature had the most impact followed by the second, and third, with all having a great impact on the learning process. Finally, Figure 209 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

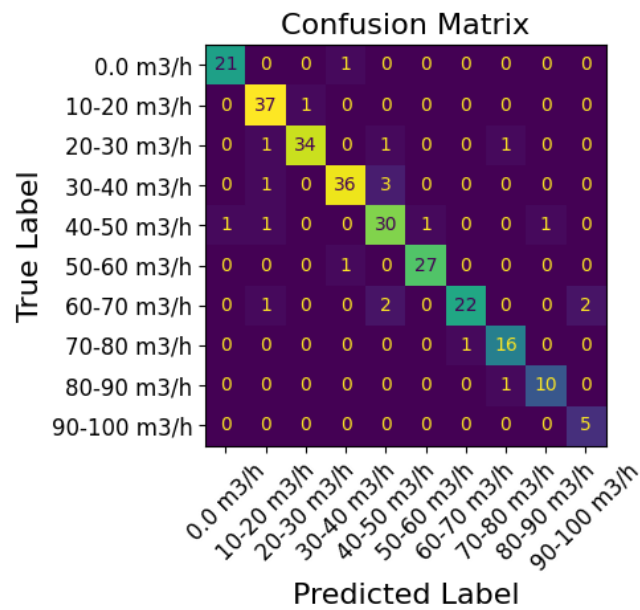


Figure 207 – SVM Confusion Matrix - CV

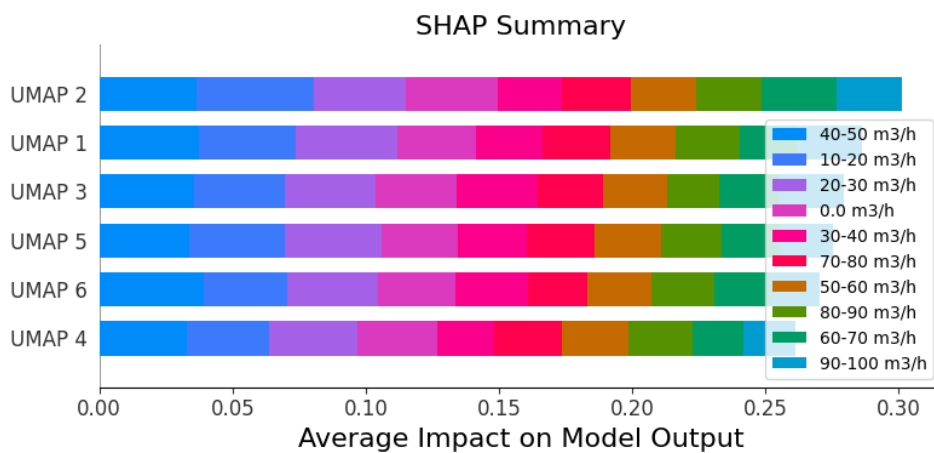


Figure 208 – SVM SHAP Values Summary - CV

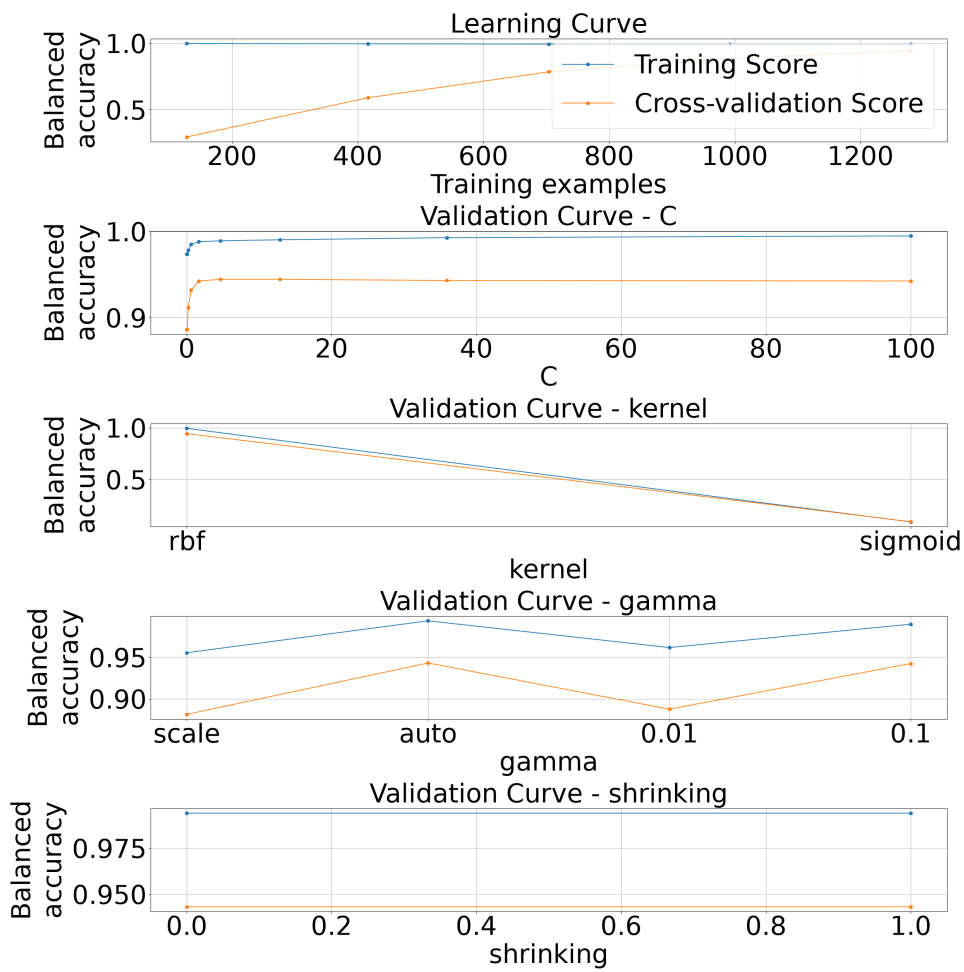


Figure 209 – SVM Learning & Validation Curves - CV

Figure 210 presents the algorithm's ability to learn each individual class, from which is clear that the Random Forest algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 211 presents how much each projected feature contributes to the Random Forest classification process, the first feature had the most impact followed by the third, and second, with all having a great impact on the learning process. Finally, Figure 212 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

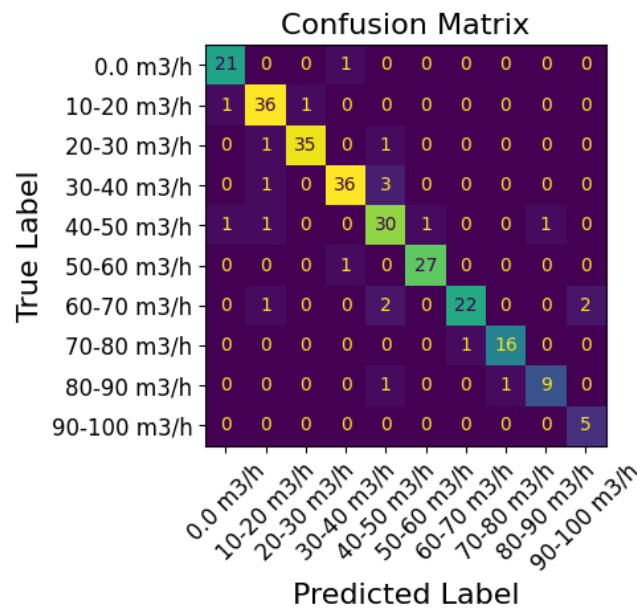


Figure 210 – Random Forest Confusion Matrix - CV

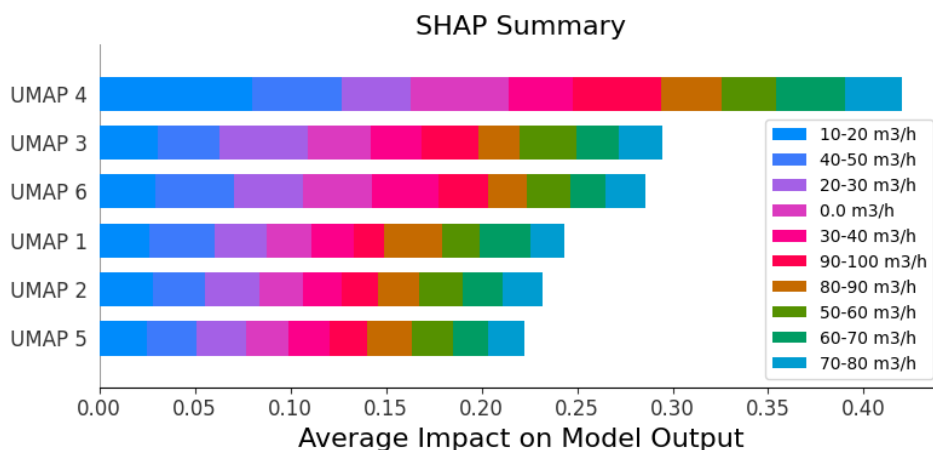


Figure 211 – Random Forest SHAP Values Summary - CV

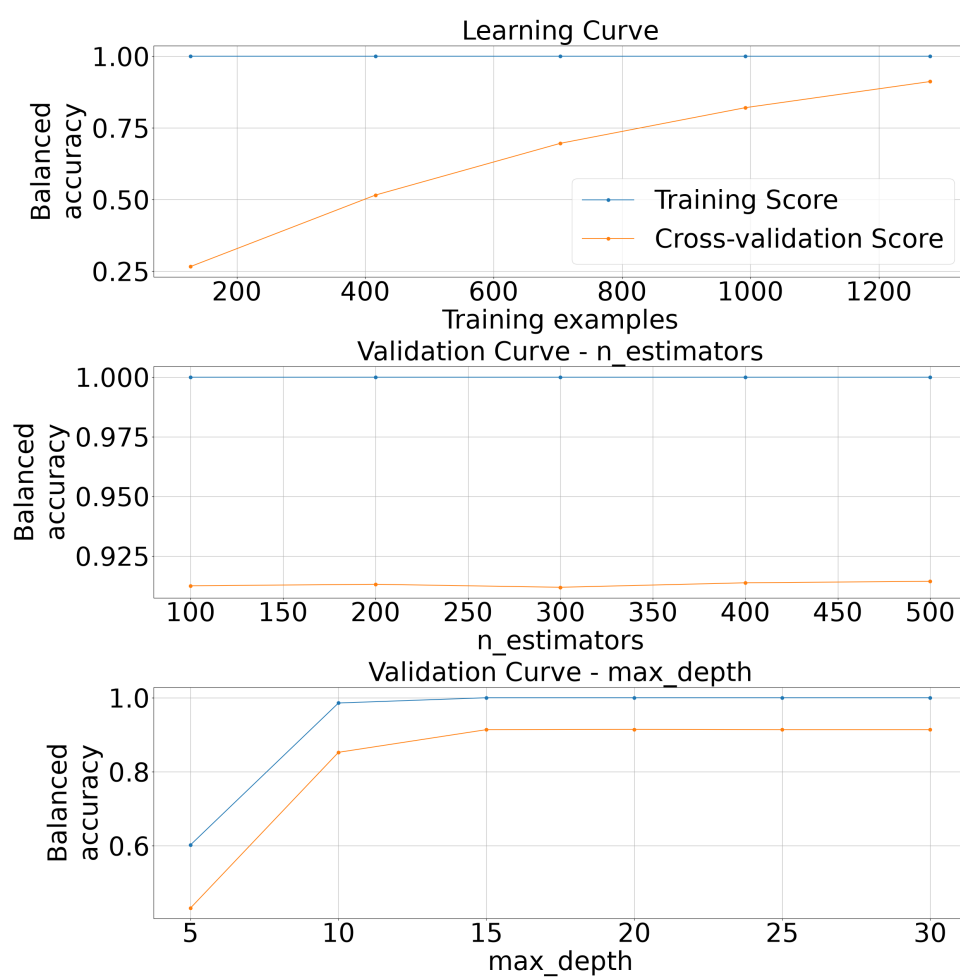


Figure 212 – Random Forest Learning & Validation Curves - CV

Figure 213 presents the algorithm's ability to learn each individual class, from which is clear that the Logistic Regression algorithm was able to learn all classes quite well, having more trouble learning the classes "50-60 m^3/h " and "30-40 m^3/h ", with a total of 20 and 16 misclassifications respectively. Figure 214 presents how much each projected feature contributes to the Logistic Regression classification process, the first feature had the most impact followed by the thirteenth, third, second, and fifth with most impact, while the eleventh, tenth, eighth and sixth are the projected features with least impact on the learning process. Finally, Figure 215 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

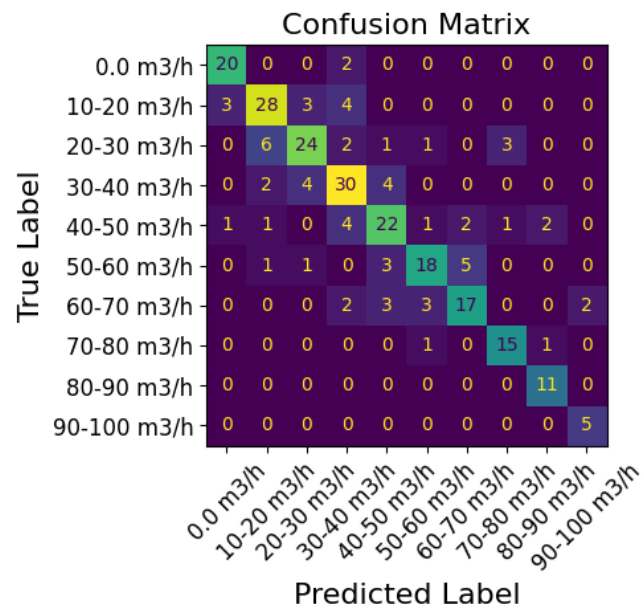


Figure 213 – Logistic Regression Confusion Matrix - CV

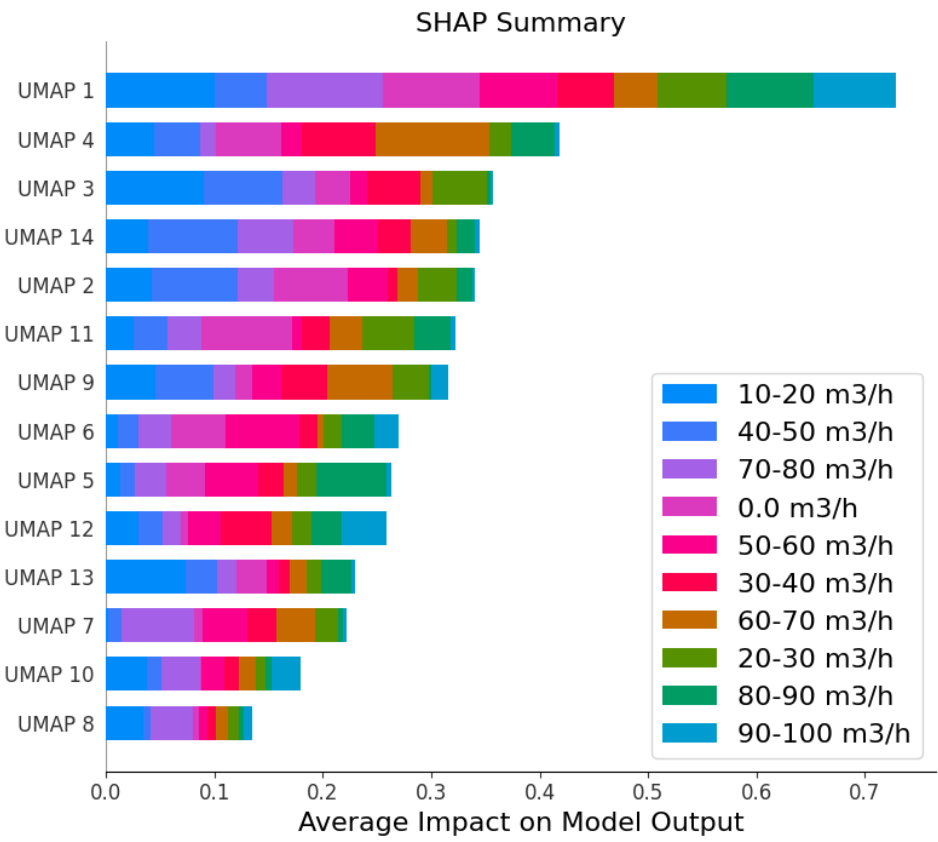


Figure 214 – Logistic Regression SHAP Values Summary - CV

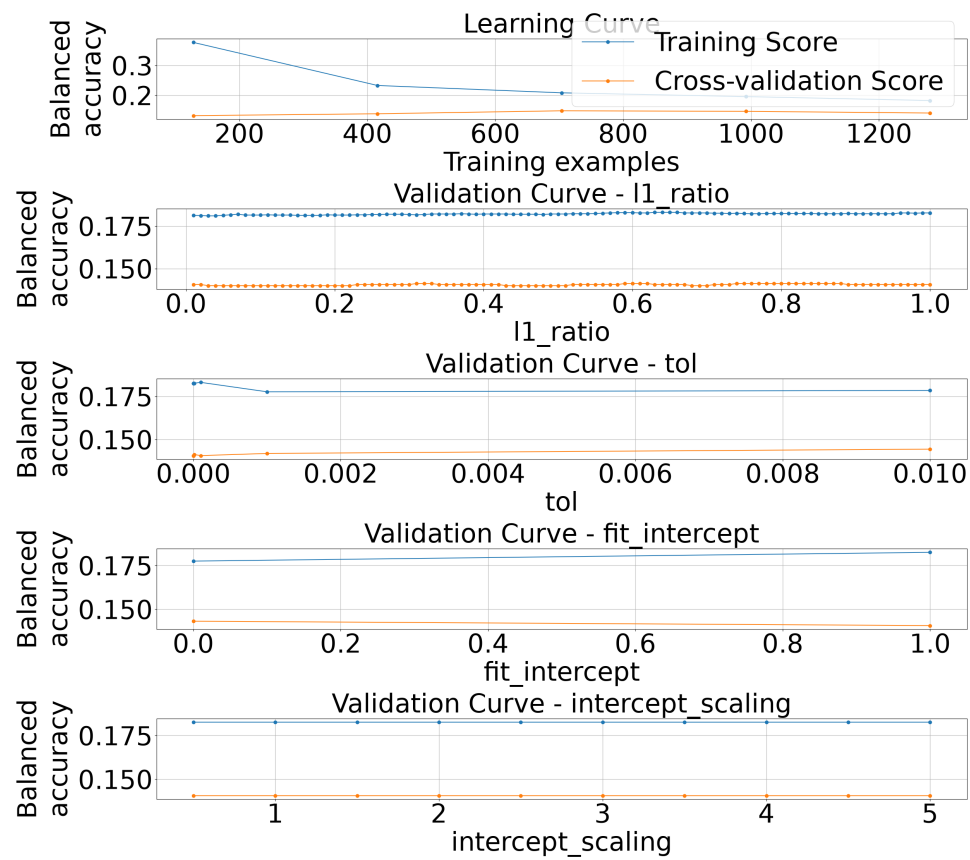


Figure 215 – Logistic Regression Learning & Validation Curves - CV

Figure 216 presents the algorithm's ability to learn each individual class, from which is clear that the GNB algorithm was able to learn all classes quite well, having a little more trouble learning "30-40 m^3/h " and "50-60 m^3/h " classes, mistaking them for the "40-50 m^3/h ", with 16 and 17 mistakes respectively. Figure 217 presents how much each projected feature contributes to the Gaussian Naive Bayes classification process, the third feature had the most impact followed by the second, while the tenth, sixth, eighth and twelfth projected features had the least impact on the learning process. Finally, Figure 218 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

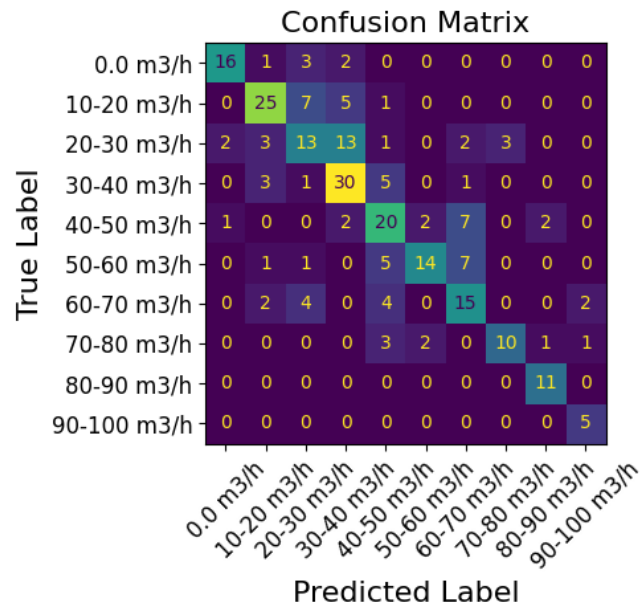


Figure 216 – GNB Confusion Matrix - CV

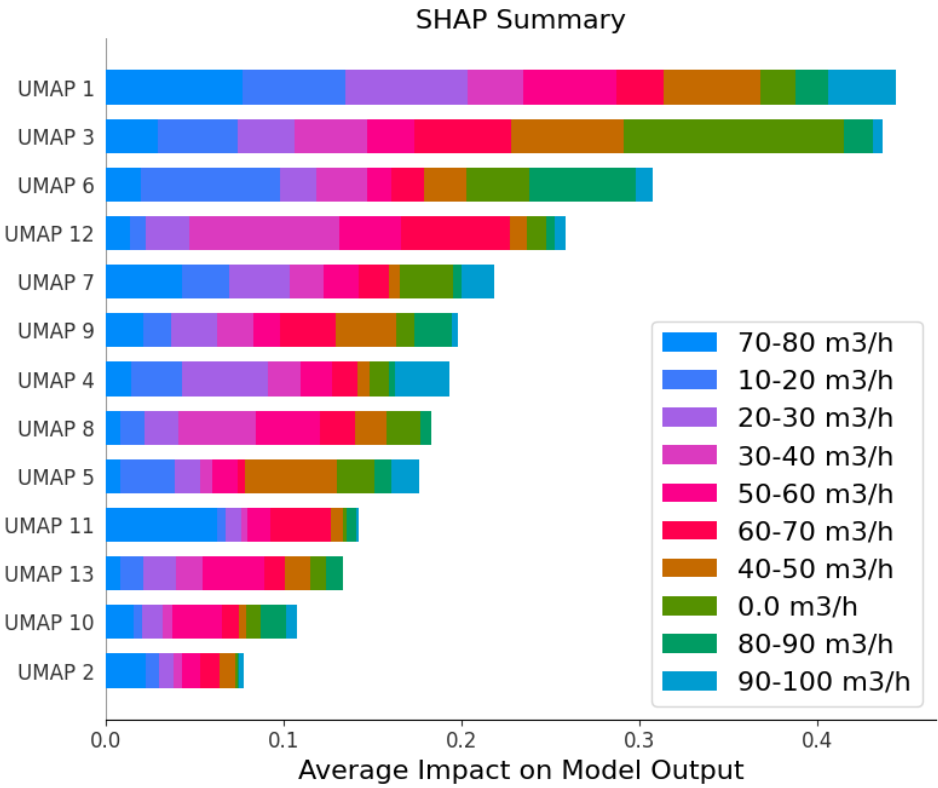


Figure 217 – GNB SHAP Values Summary - CV

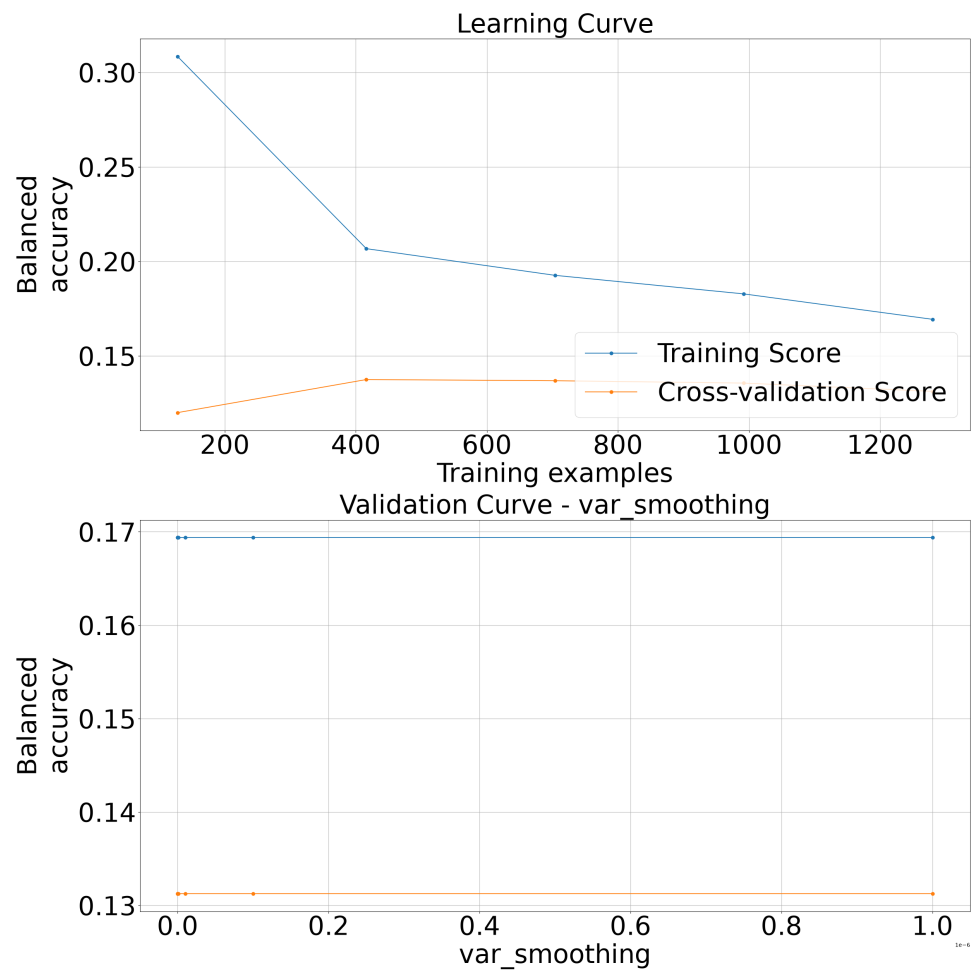


Figure 218 – GNB Learning & Validation Curves - CV

Figure 219 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "40-50 m^3/h " class, mistaking it for the "50-60 m^3/h ". Figure 220 presents how much each projected feature contributes to the XGBoost classification process, the first feature had the most impact followed by the fourth, and second, while the tenth, sixth, seventh and eighth projected features had the least impact on the learning process. Finally, Figure 221 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

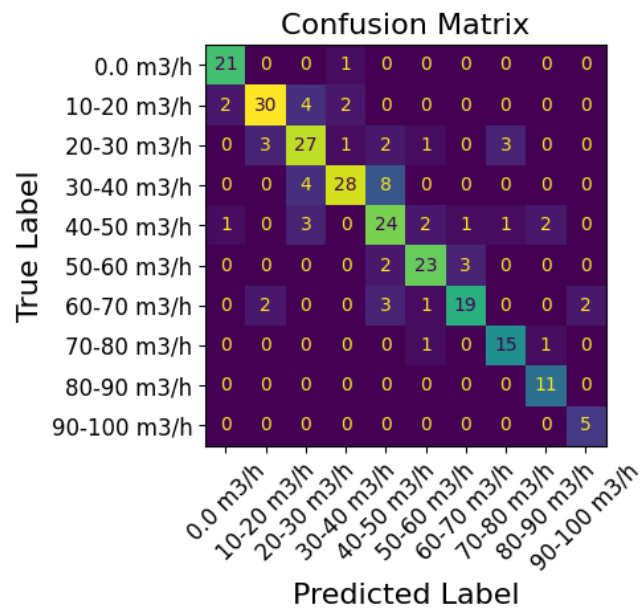


Figure 219 – XGBoost Confusion Matrix - CV

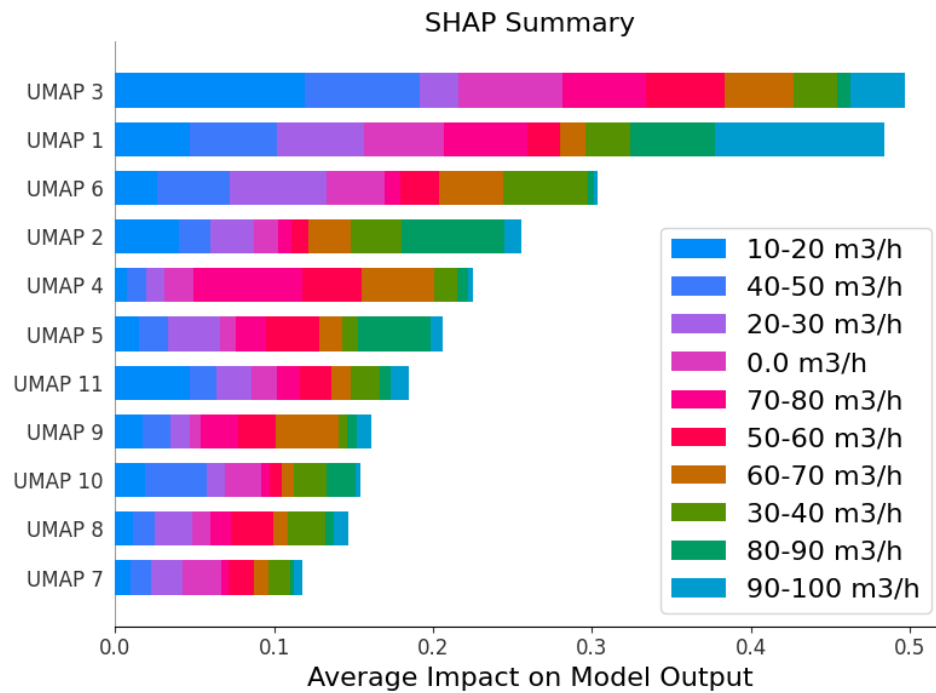


Figure 220 – XGBoost SHAP Values Summary - CV

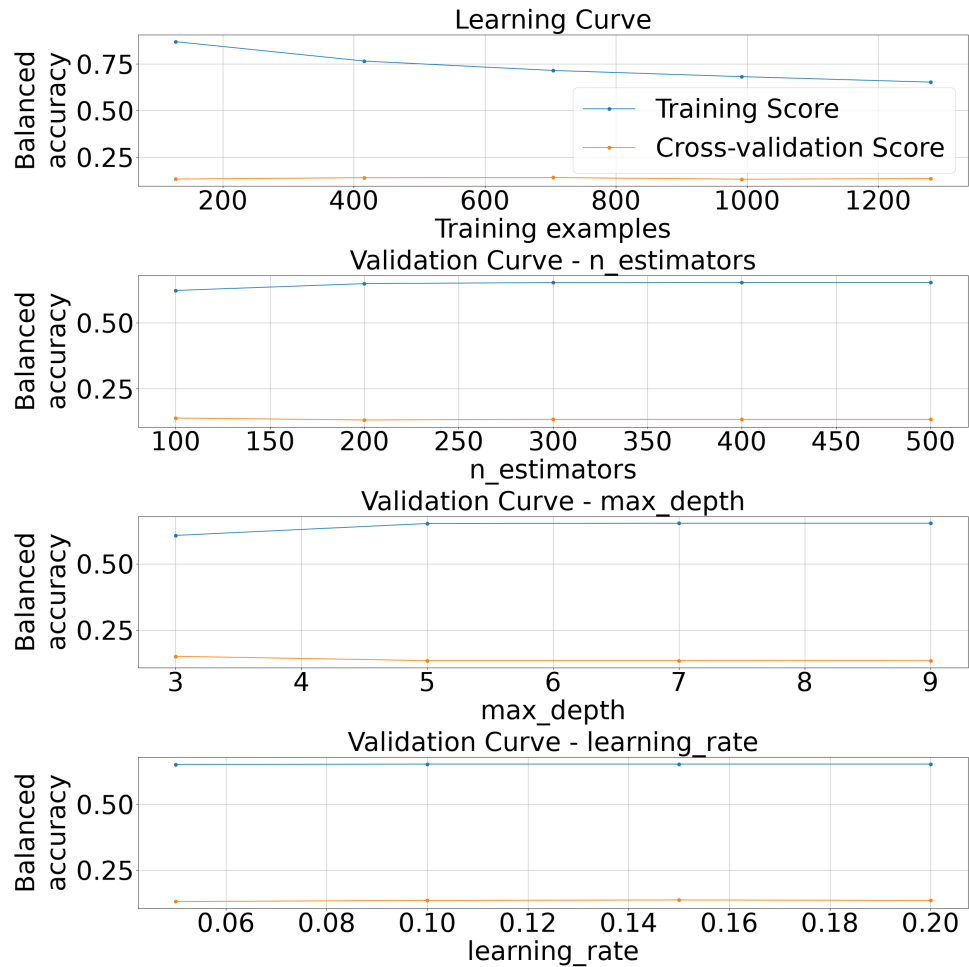


Figure 221 – XGBoost Learning & Validation Curves - CV

Figure 222 presents the algorithm's ability to learn each individual class, from which is clear that the MLP algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 223 presents how much each projected feature contributes to the MLP classification process, the first feature had the most impact followed by the fourth, and third, while the second, seventh, eighth and eleventh had a least impact on the learning process. Finally, Figure 224 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

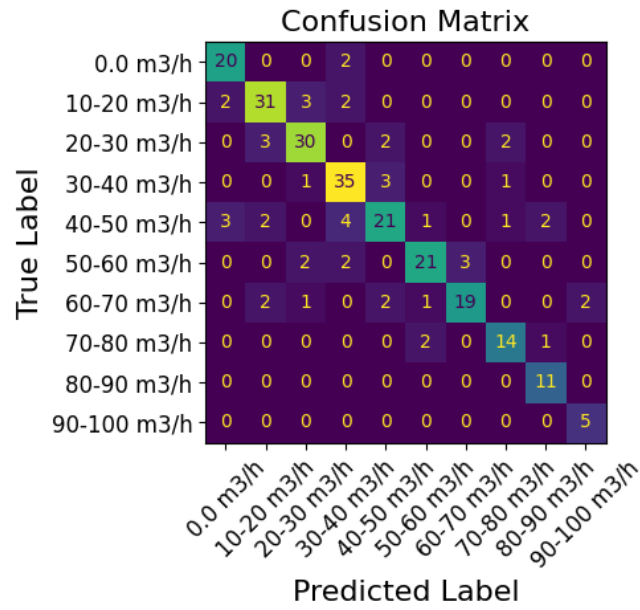


Figure 222 – MLP Confusion Matrix - CV

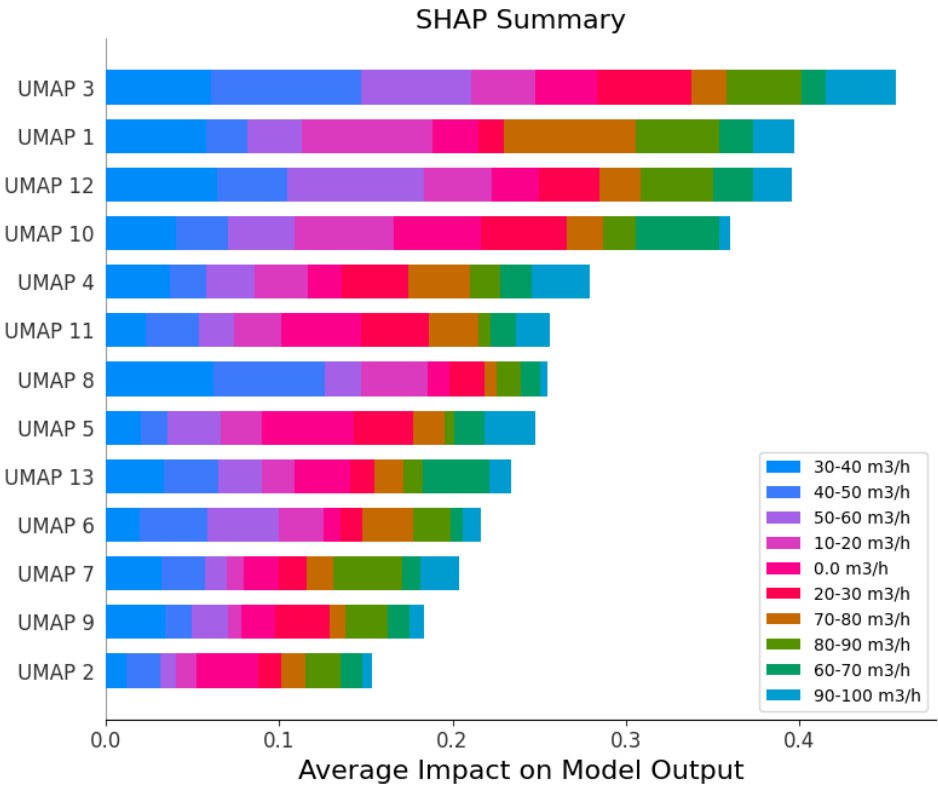


Figure 223 – MLP SHAP Values Summary - CV



Figure 224 – MLP Learning & Validation Curves - CV

	GMM	K-Means	HDBSCAN	Agglomerative	Spectral	Mean Shift
Silhouette	0.60	0.68	0.82	0.61	0.42	0.90
ARI	0.12	0.14	0.06	0.13	0.11	0.02
DBI	0.71	0.54	0.79	0.50	1.92	0.13
CH	640.99	1059.36	621.03	1148.90	188.89	6066.88

Table 15 – Unsupervised Learning Results - CV

4.2.3.2 Unsupervised Learning

Table 15 summarize the overall performance of all tested unsupervised algorithms. The consistently low ARI scores indicate poor alignment between the discovered clusters and the ground-truth flow rate labels, suggesting that the algorithms were not able to recover the flow rate structure in the feature manifold. However, the remaining metrics show that several methods still identified a clear and coherent structure in the data. This structure is primarily associated with the rotational speed, which is an expected outcome since the dominant spectral and dynamical components of the motor signals are directly tied to shaft rotation and its harmonics. As a result, the extracted Koopman features naturally exhibit larger variance with respect to RPM than to flow variations, leading the algorithms to form well-defined RPM-driven clusters.

The table shows that GMM’s Silhouette score of 0.74, indicating that the algorithm formed well defined clusters – however, there’s still some mixture with the other metrics re-enforcing that. The low ARI score tells the formed clusters don’t align well with the true classes, which is further confirmed by its confusion matrix in [Figure 225](#). [Figure 226](#) presents how much each projected feature contributes to the GMM classification process, the first feature had the most impact followed by the third, and second, while the eighth, ninth and seventh had the least impact on the learning process.

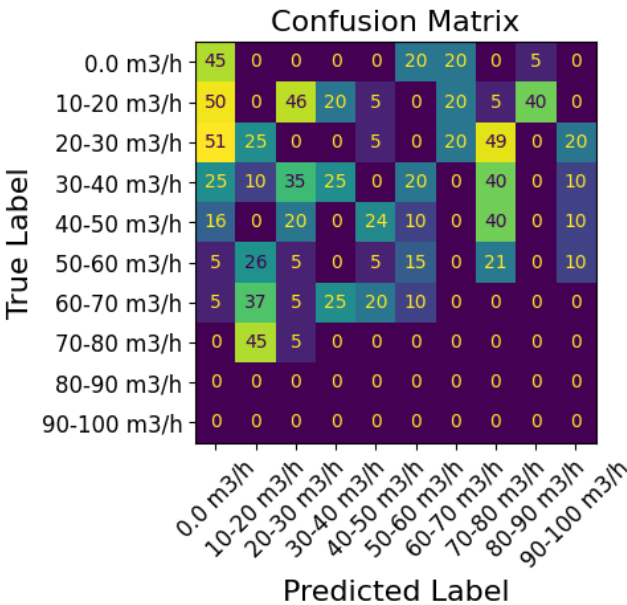


Figure 225 – GMM Confusion Matrix - CV

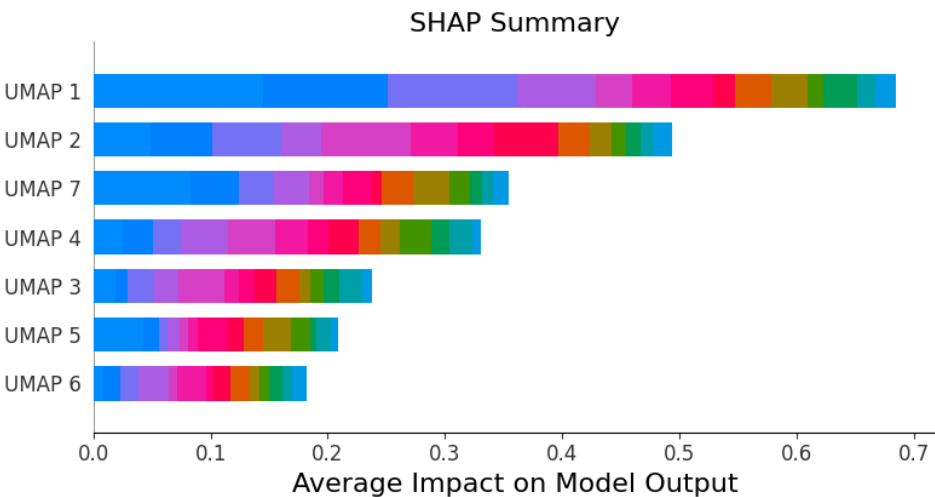


Figure 226 – GMM SHAP Values Summary - CV

Kmeans reached a Silhouette score of 0.68, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 227. Figure 228 presents how much each projected feature contributes to the Kmeans classification process, the first feature had the most impact followed by the second, and third, while the fifth, fourth and sixth had the least impact on the learning process.

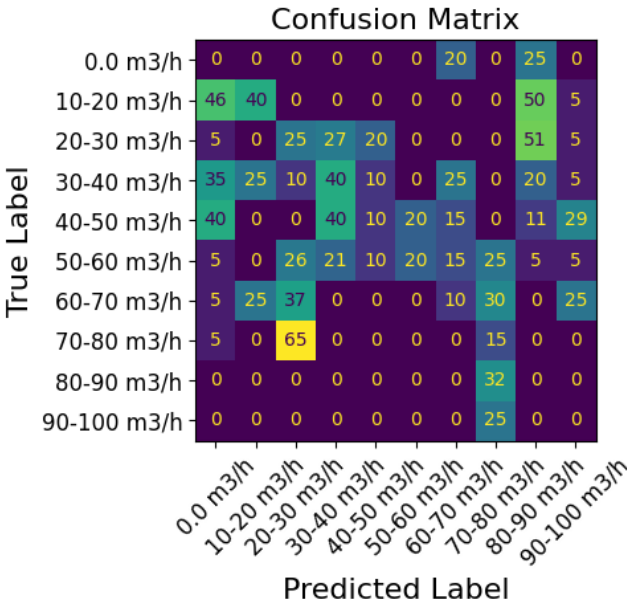


Figure 227 – KMeans Confusion Matrix - CV

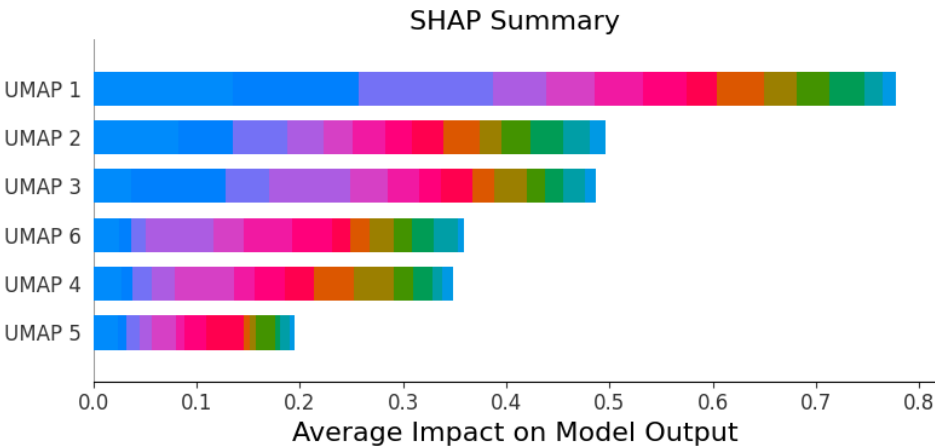


Figure 228 – KMeans SHAP Values Summary - CV

HDBSCAN reached a Silhouette score of 0.79, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 229. Figure 230 presents how much each projected feature contributes to the HDBSCAN classification process, the optimization process picked only 2 projected features and both greatly contributed on the learning process.

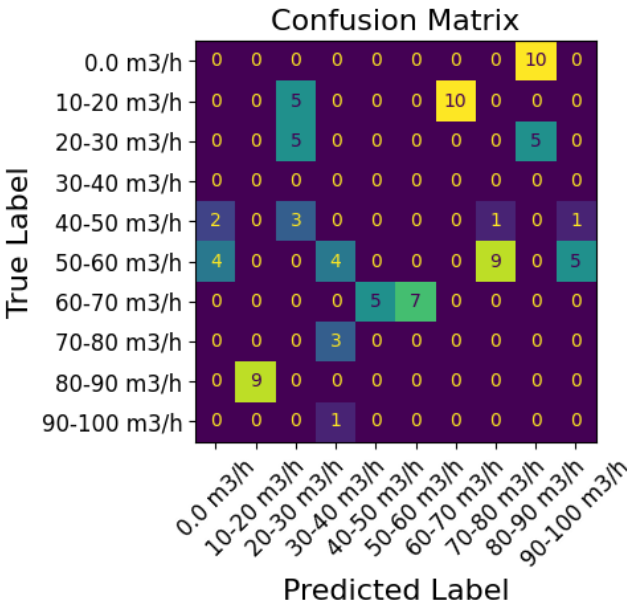


Figure 229 – HDBSCAN Confusion Matrix - CV

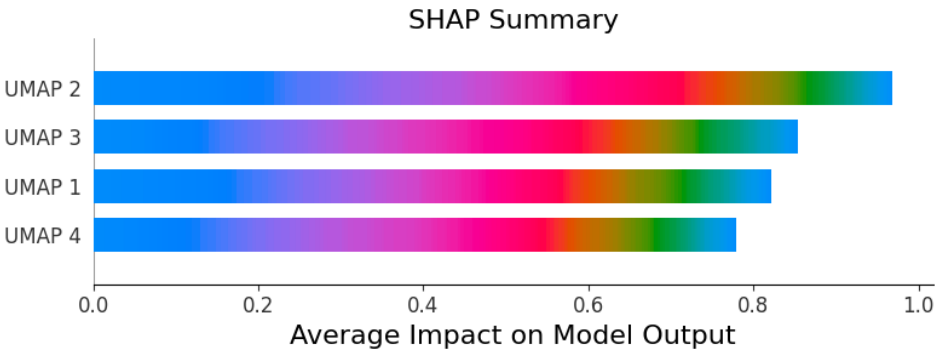


Figure 230 – HDBSCAN SHAP Values Summary - CV

Agglomerative Clustering got a Silhouette score of 0.61, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 231. Figure 232 presents how much each projected feature contributes to the Agglomerative Clustering the optimization process picked only 3 projected features and both greatly contributed on the learning process.

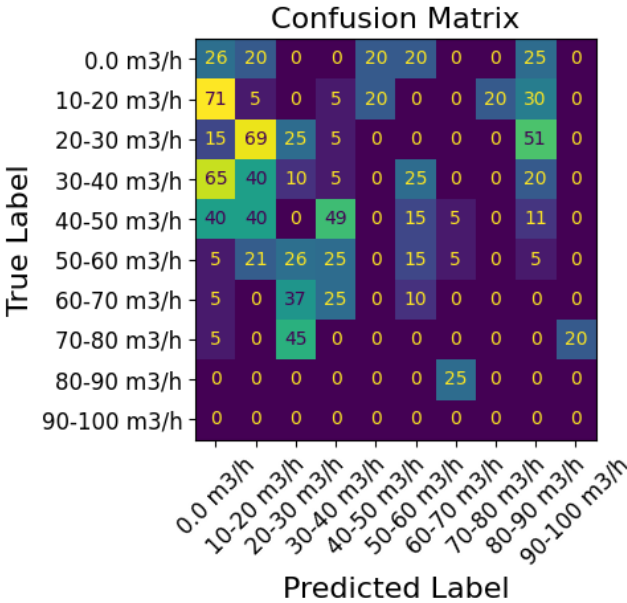


Figure 231 – Agglomerative Clustering Confusion Matrix - CV

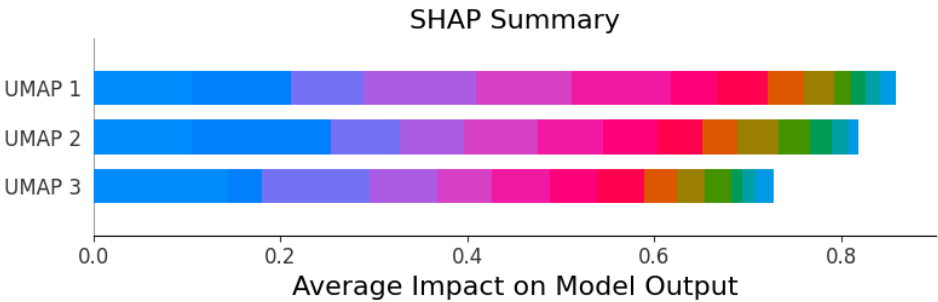


Figure 232 – Agglomerative Clustering SHAP Values Summary - CV

The Spectral Clustering's Silhouette score of 0.56, indicating that the algorithm that the algorithm had more trouble forming clusters, meaning that this method isn't suited to cluster signals with this mixture, with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 233. Figure 234 presents how much each projected feature contributes to the Spectral Clustering classification process, the first feature had the most impact followed by the fourth, eleventh, third, and second, while the eighth, twelfth, fifth and ninth had the least impact on the learning process.

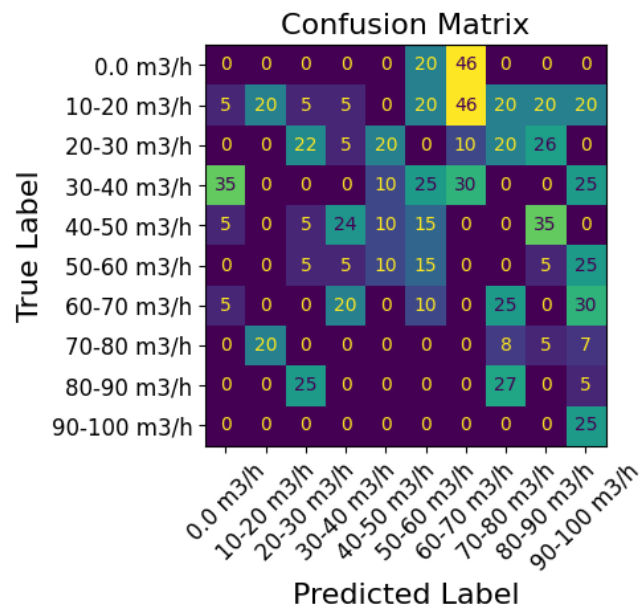


Figure 233 – Spectral Clustering Confusion Matrix - CV

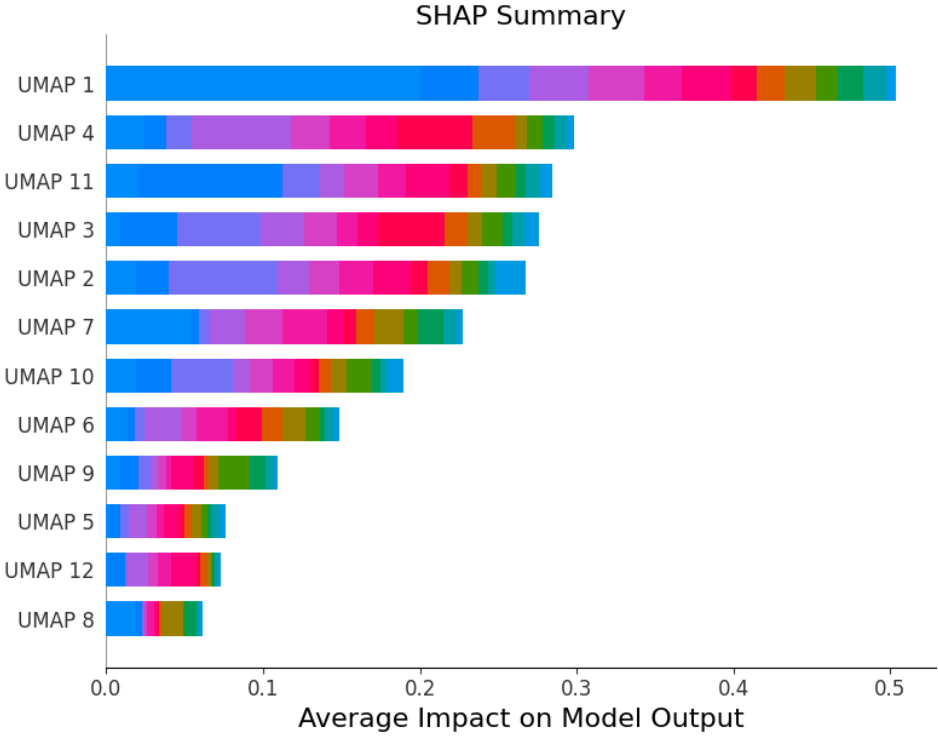


Figure 234 – Spectral Clustering SHAP Values Summary - CV

Finally, the Mean Shift algorithm reached a Silhouette score of 0.97, indicating that the algorithm formed great clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 235. Figure 236 presents how much each projected feature contributes to the Mean Shift classification process, the optimization process picked only 2 projected features and both greatly contributed on the learning process.

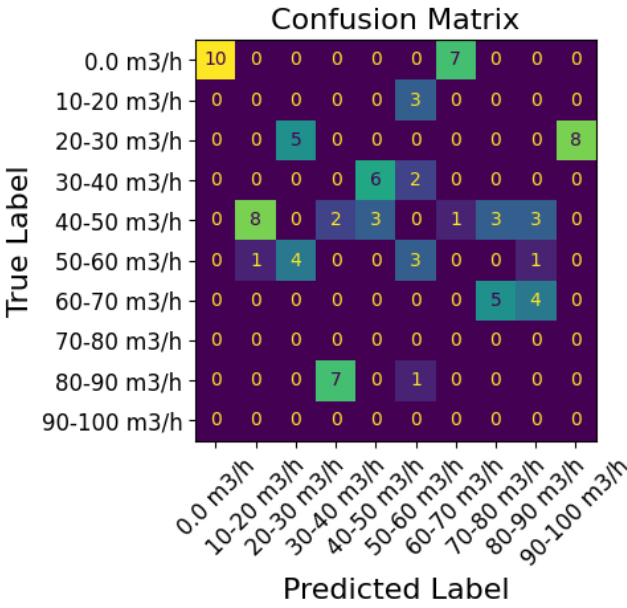


Figure 235 – Mean Shift Confusion Matrix - CV

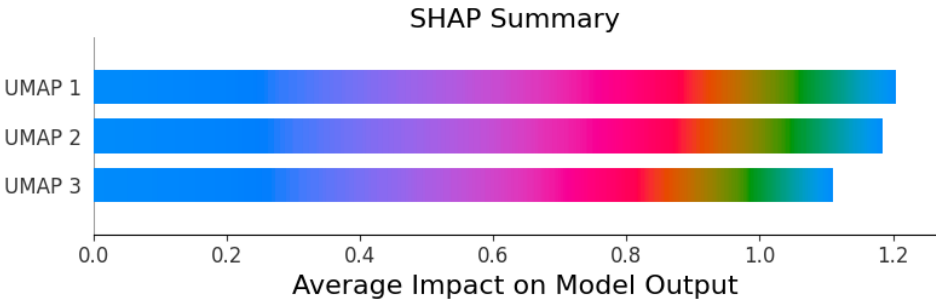


Figure 236 – Mean Shift SHAP Values Summary - CV

	k-NN	SVM	Random Forest	LR	GNB	XGBoost	MLP
CV Accuracy	96.3%	92.6%	80.9%	74.1%	61.6%	97.4%	90.7%
Test Accuracy	89.3%	89%	89%	74.2%	70.5%	84.2%	80.5%
F1 Score	0.90	0.90	0.90	0.67	0.63	0.81	0.76
ROC-AUC	0.96	0.96	0.95	0.92	0.90	0.93	0.93

Table 16 – Supervised Learning Results - CTV

4.2.4 3 Currents + 3 Voltages + 2 Vibrational Axes (CTV)

4.2.4.1 Supervised Pipeline

Table 16 summarize the overall performance of all tested supervised algorithms, showing that k-NN was the best performing algorithm followed by SVM, Random Forest, XGBoost, and MLP. The Logistic Regression and Gaussian Naive Bayes had lower accuracies due to the algorithm requirement being broken. Logistic Regression, for example, uses linear combinations of the inputs to classify the data, but if the data is nonlinear, then the algorithm will struggle more to learn the feature manifold and make correct classifications. Likewise, GNB assumes Gaussian distributed features so, if this assumption is broken, it would lead to a lower classification accuracy. However, the UMAP + SMOTE preprocessing step partially mitigates this limitations. By enforcing local metric linearity and introducing neighbourhood-based interpolations, the resulting feature space becomes more compatible with linear decision boundaries and Gaussian class-conditional modeling, allowing these algorithms achieve relatively high accuracies despite their theoretical limitations.

Figure 237 presents the algorithm's ability to learn each individual class, from which is clear that the k-NN algorithm was able to learn all classes quite well, having a little more trouble learning the "30-40 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 238 presents how much each projected feature contributes to the k-NN classification process, the first feature had the most impact followed by the fourth, then second, the seventh, sixth and fifth were the ones with least impact but still having meaningful contribution for the classification. Finally, Figure 239 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

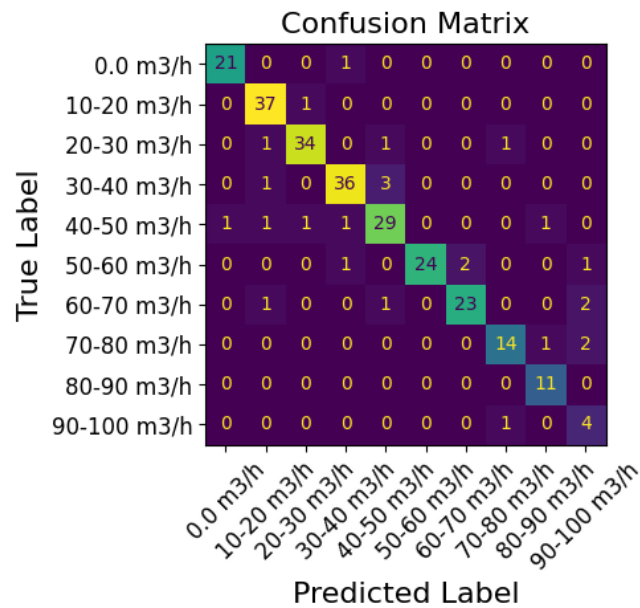


Figure 237 – k-NN Confusion Matrix - CTV

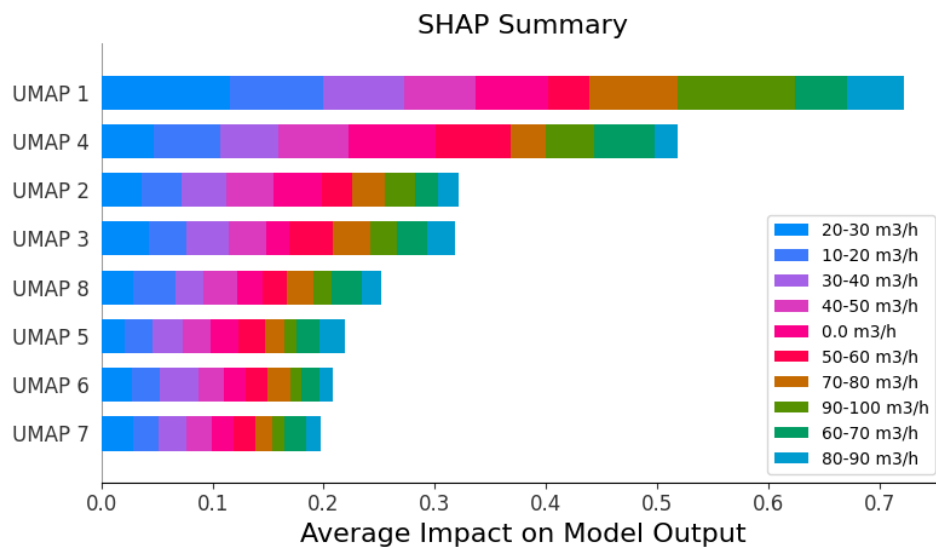


Figure 238 – k-NN SHAP Values Summary - CTV

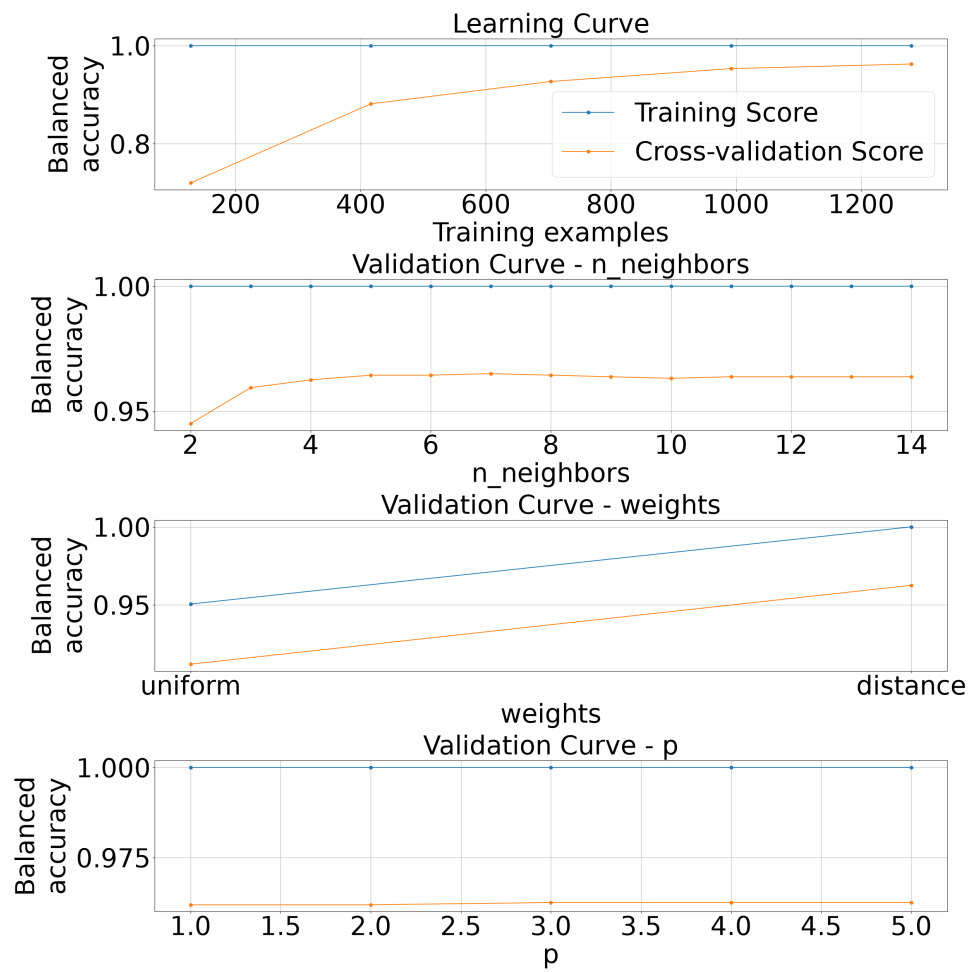


Figure 239 – k-NN Learning & Validation Curves - CTV

Figure 240 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "40-50 m^3/h " class, mistaking it for the "30-40 m^3/h ". Figure 241 presents how much each projected feature contributes to the SVM classification process, the first feature had the most impact followed by the third, and second, with all having a great impact on the learning process. Finally, Figure 242 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

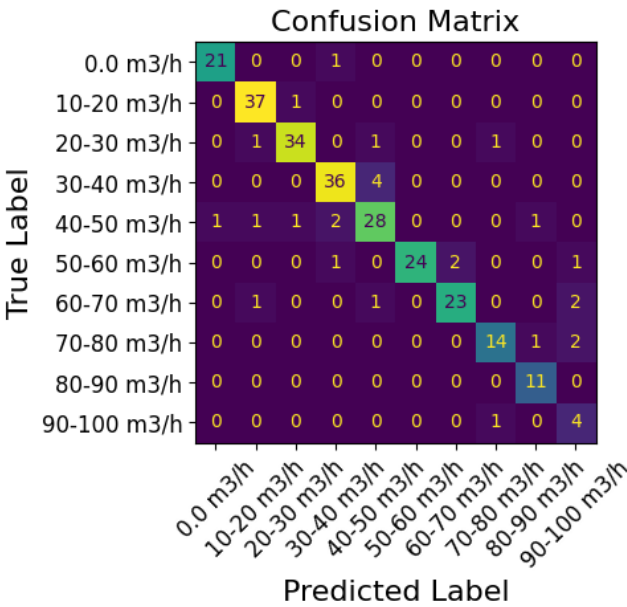


Figure 240 – SVM Confusion Matrix - CTV

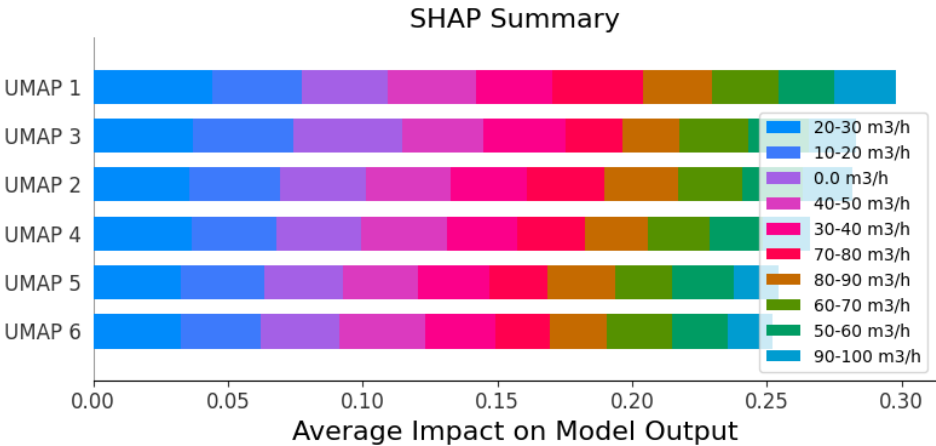


Figure 241 – SVM SHAP Values Summary - CTV

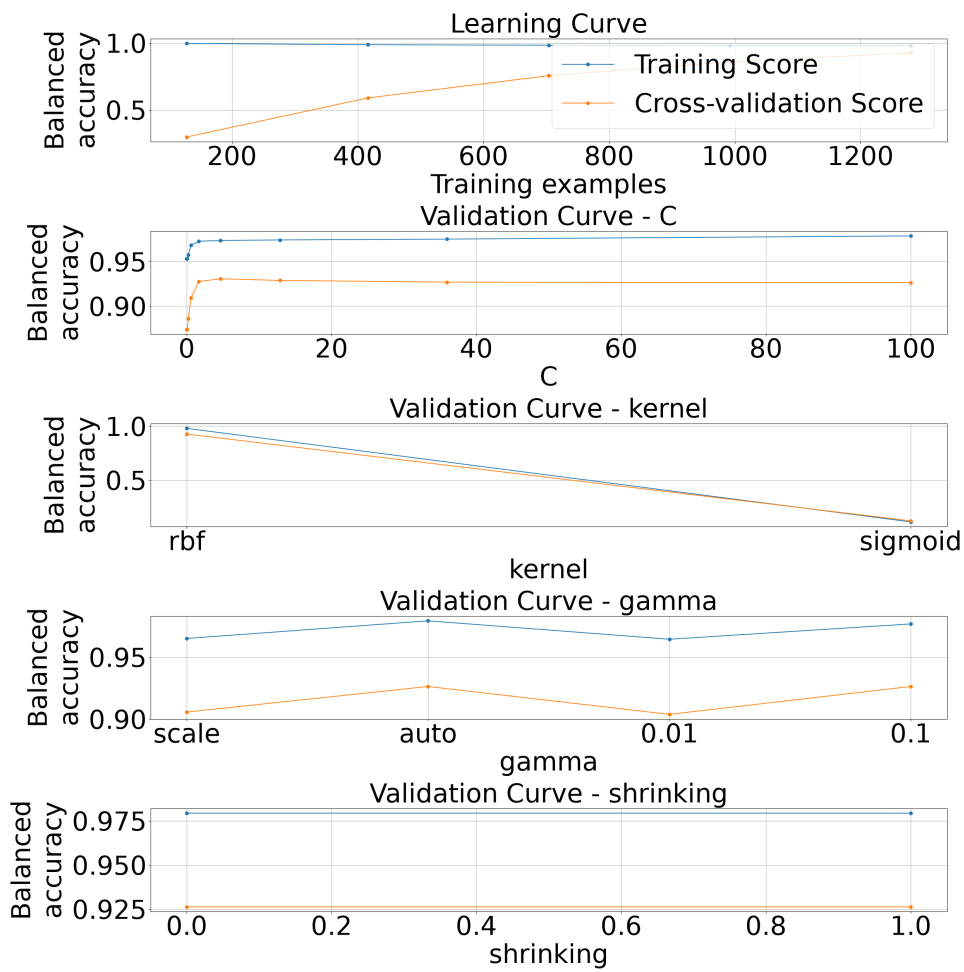


Figure 242 – SVM Learning & Validation Curves - CTV

Figure 243 presents the algorithm's ability to learn each individual class, from which is clear that the Random Forest algorithm was able to learn all classes quite well, having a little more trouble learning the "40-50 m^3/h " class, with a total of 6 mistakes. Figure 244 presents how much each projected feature contributes to the Random Forest classification process, the third feature had the most impact followed by the first, then second, while the fourth and fifth were the ones with least impact but still having meaningful contribution for the classification. Finally, Figure 245 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

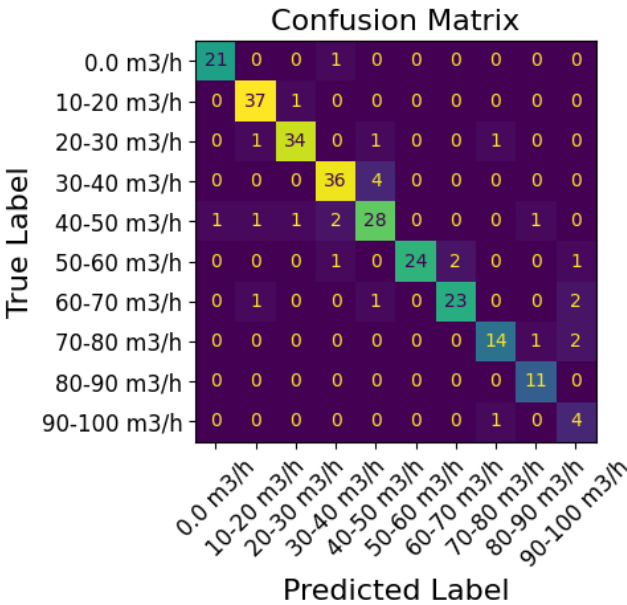


Figure 243 – Random Forest Confusion Matrix - CTV

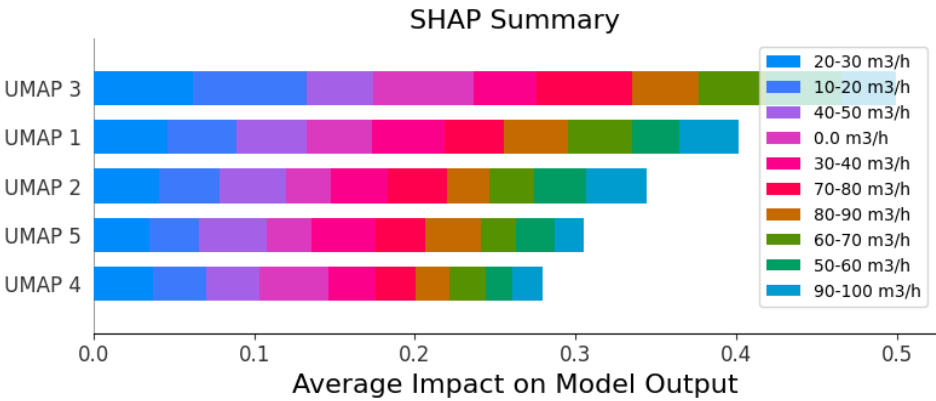


Figure 244 – Random Forest SHAP Values Summary - CTV

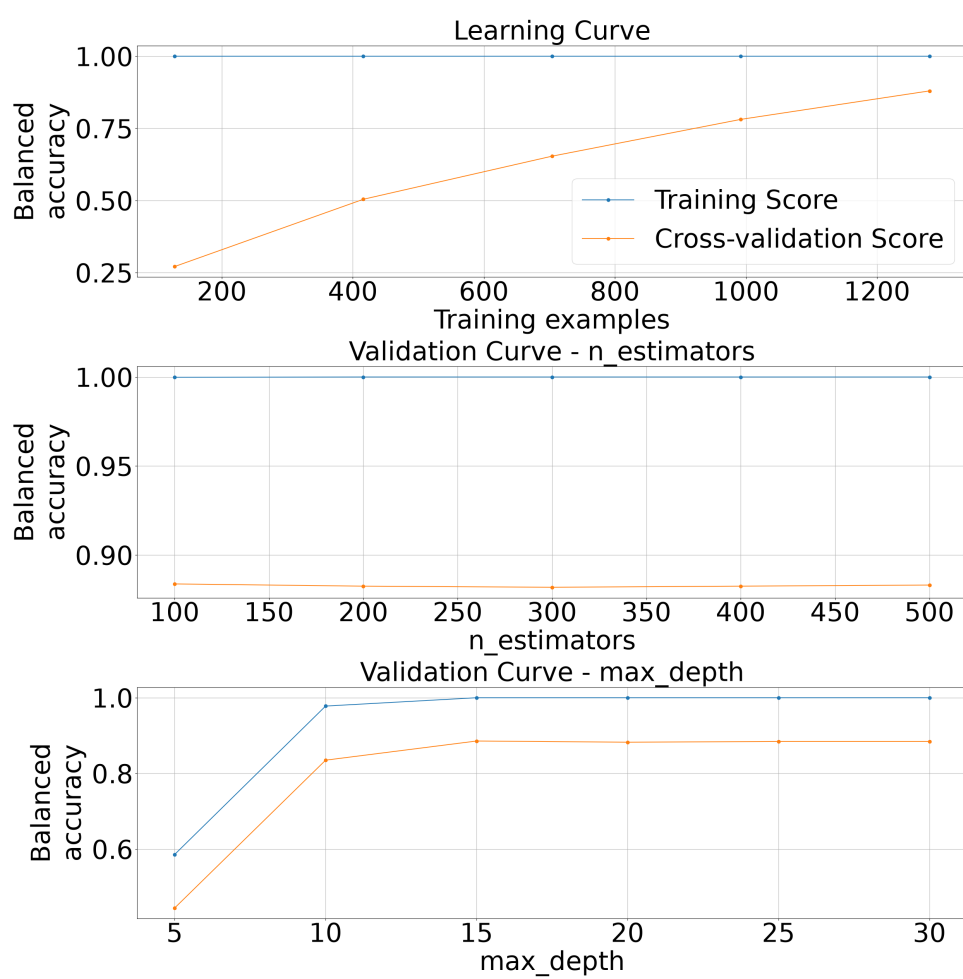


Figure 245 – Random Forest Learning & Validation Curves - CTV

Figure 246 presents the algorithm's ability to learn each individual class, from which is clear that the Logistic Regression algorithm was able to learn all classes quite well, having more trouble learning the classes "30-40 m^3/h ", mistaking it for the "10-20 m^3/h " class. Figure 247 presents how much each projected feature contributes to the Logistic Regression classification process, the first feature had the most impact followed by the fourth, fourteenth, and third with most impact, while the eleventh, eighth, twelfth, fifth and ninth are the projected features with least impact on the learning process. Finally, Figure 248 demonstrates that there a little indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

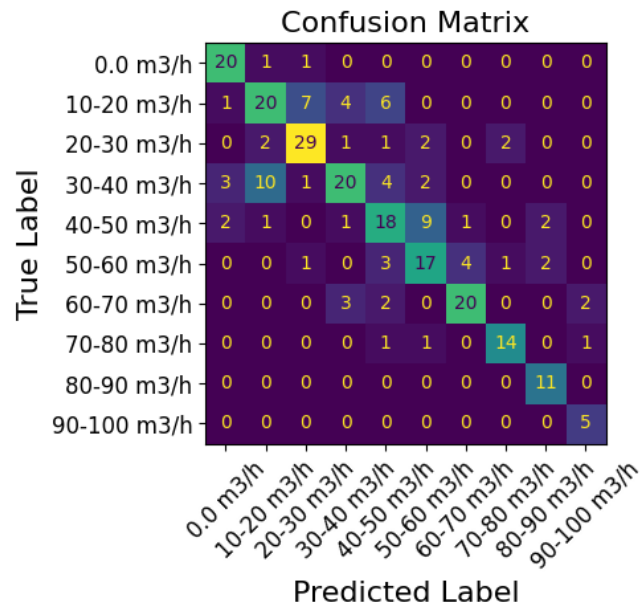


Figure 246 – Logistic Regression Confusion Matrix - CTV

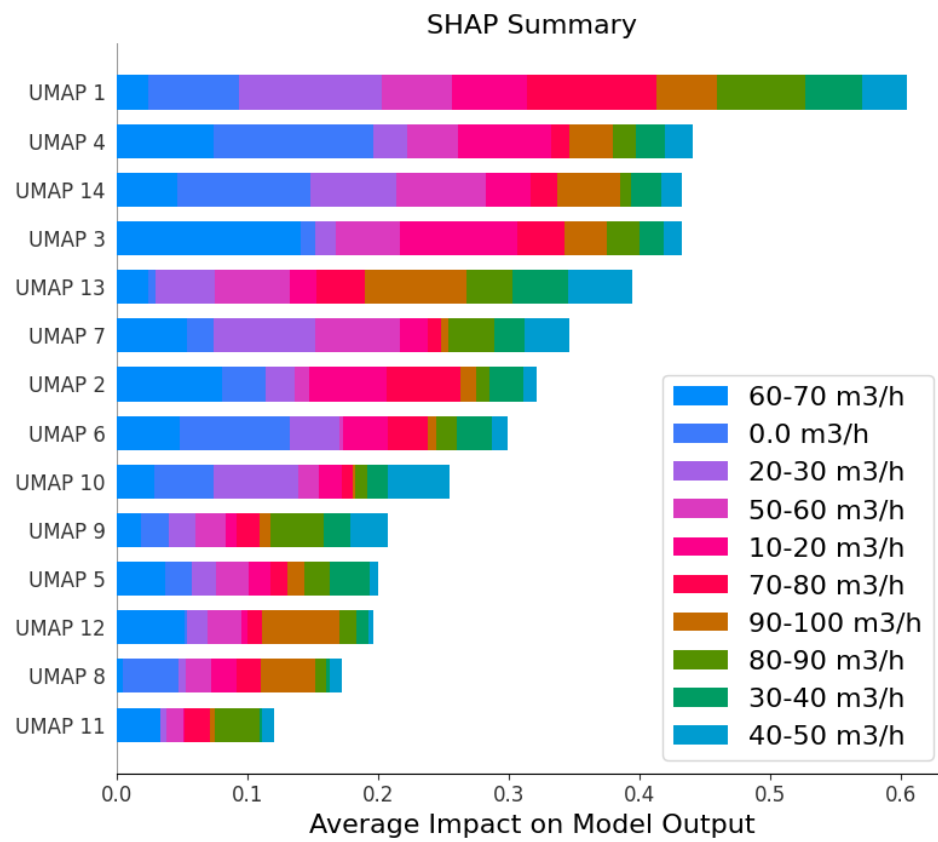


Figure 247 – Logistic Regression SHAP Values Summary - CTV

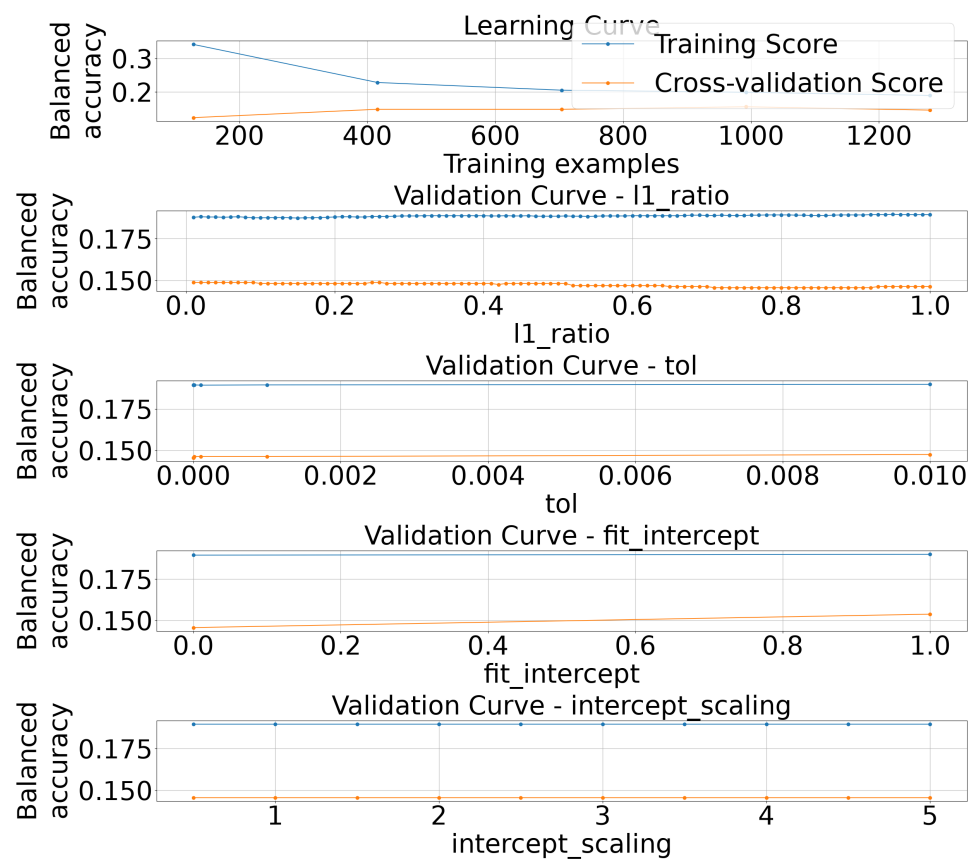


Figure 248 – Logistic Regression Learning & Validation Curves - CTV

Figure 249 presents the algorithm's ability to learn each individual class, from which is clear that the GNB algorithm was able to learn all classes quite well, having a little more trouble learning "10-20 m^3/h " class, with a total of 23 mistakes. Figure 250 presents how much each projected feature contributes to the Gaussian Naive Bayes classification process, the second feature had the most impact followed by the third, while the ninth, eighth, thirteenth, sixth and twelfth projected features had the least impact on the learning process. Finally, Figure 251 demonstrates that there is a little indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

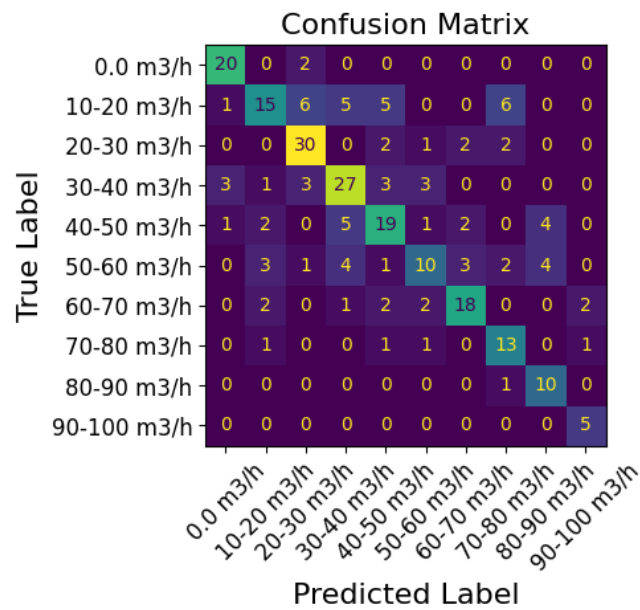


Figure 249 – GNB Confusion Matrix - CTV

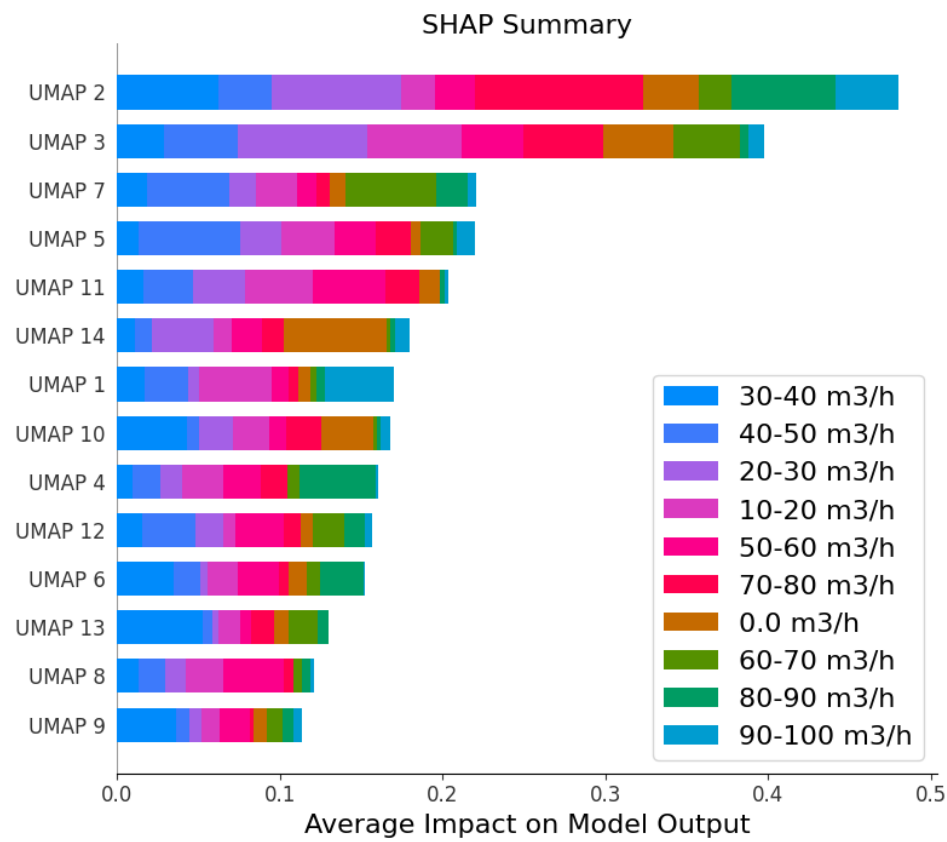


Figure 250 – GNB SHAP Values Summary - CTV

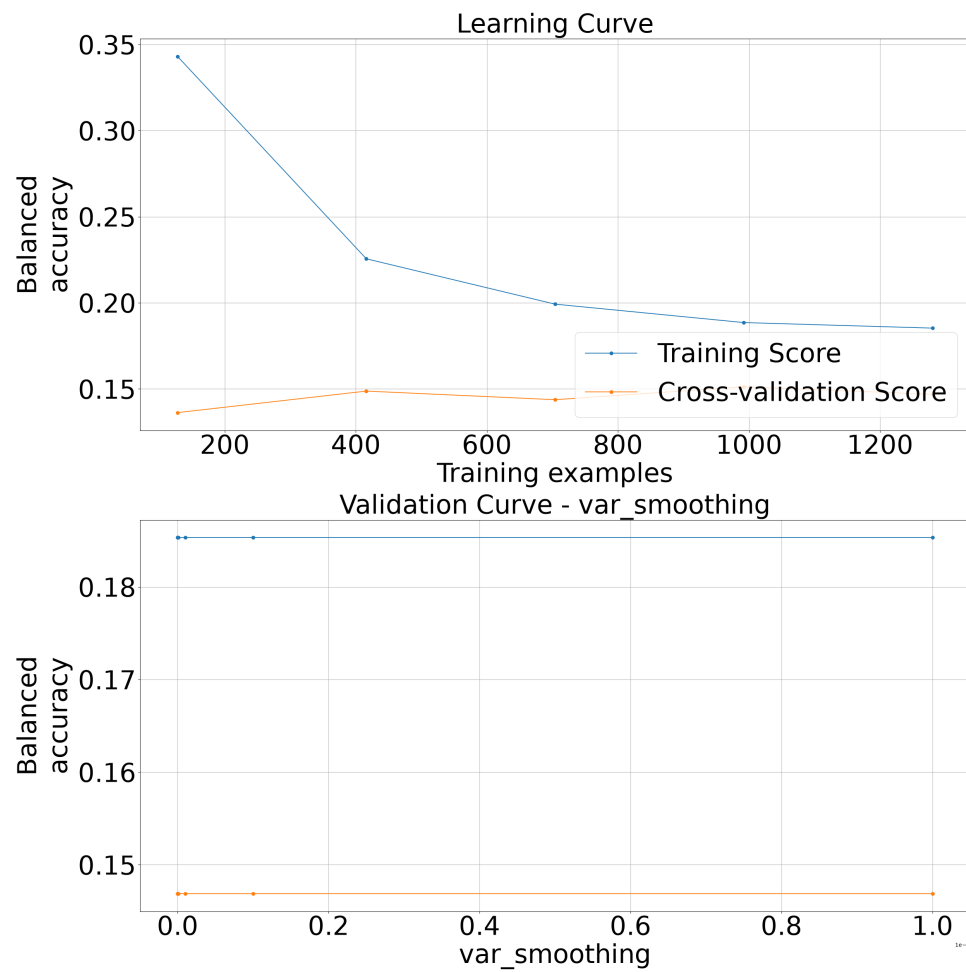


Figure 251 – GNB Learning & Validation Curves - CTV

Figure 252 presents the algorithm's ability to learn each individual class, from which is clear that the SVM algorithm was able to learn all classes quite well, having a little more trouble learning the "40-50 m^3/h " class, with a total of 11 mistakes. Figure 253 presents how much each projected feature contributes to the XGBoost classification process, the third feature had the most impact followed by the first, and fifth, while the ninth, tenth, and eighth projected features had the least impact on the learning process. Finally, Figure 254 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

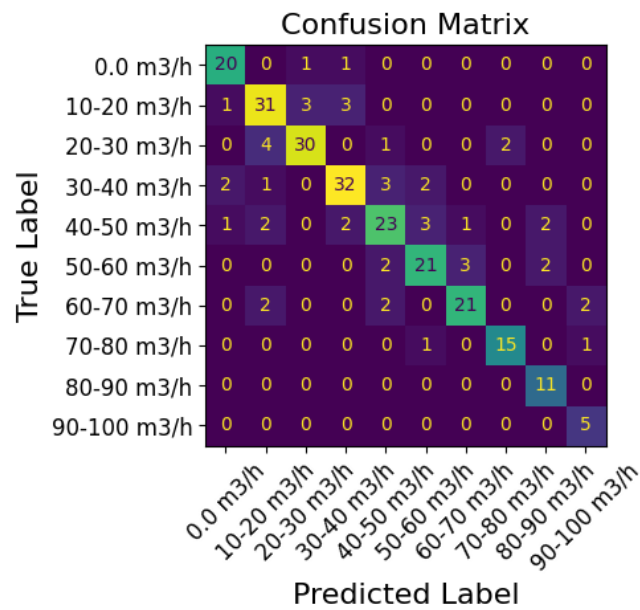


Figure 252 – XGBoost Confusion Matrix - CTV

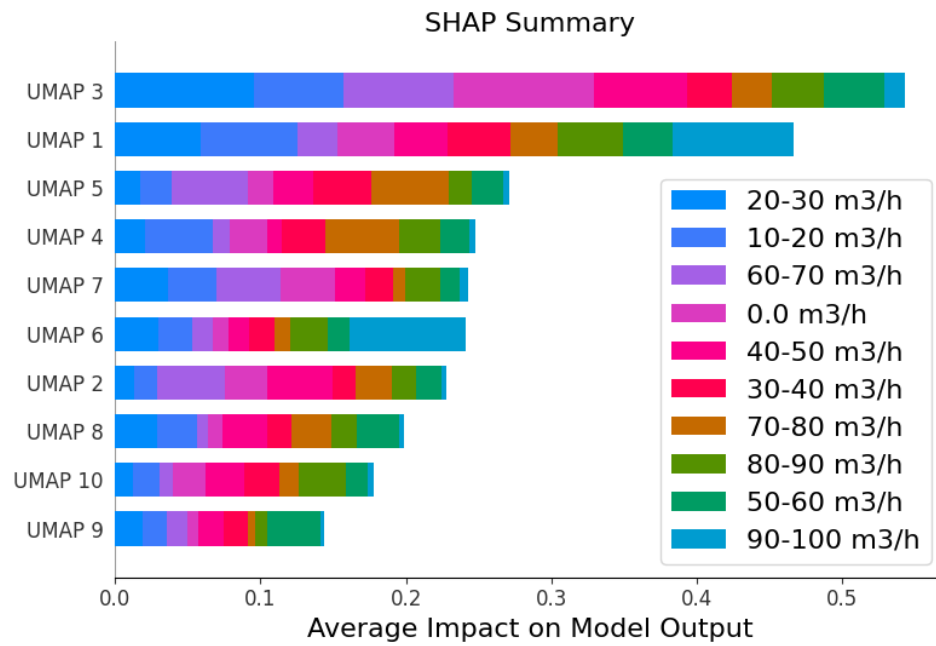


Figure 253 – XGBoost SHAP Values Summary - CTV

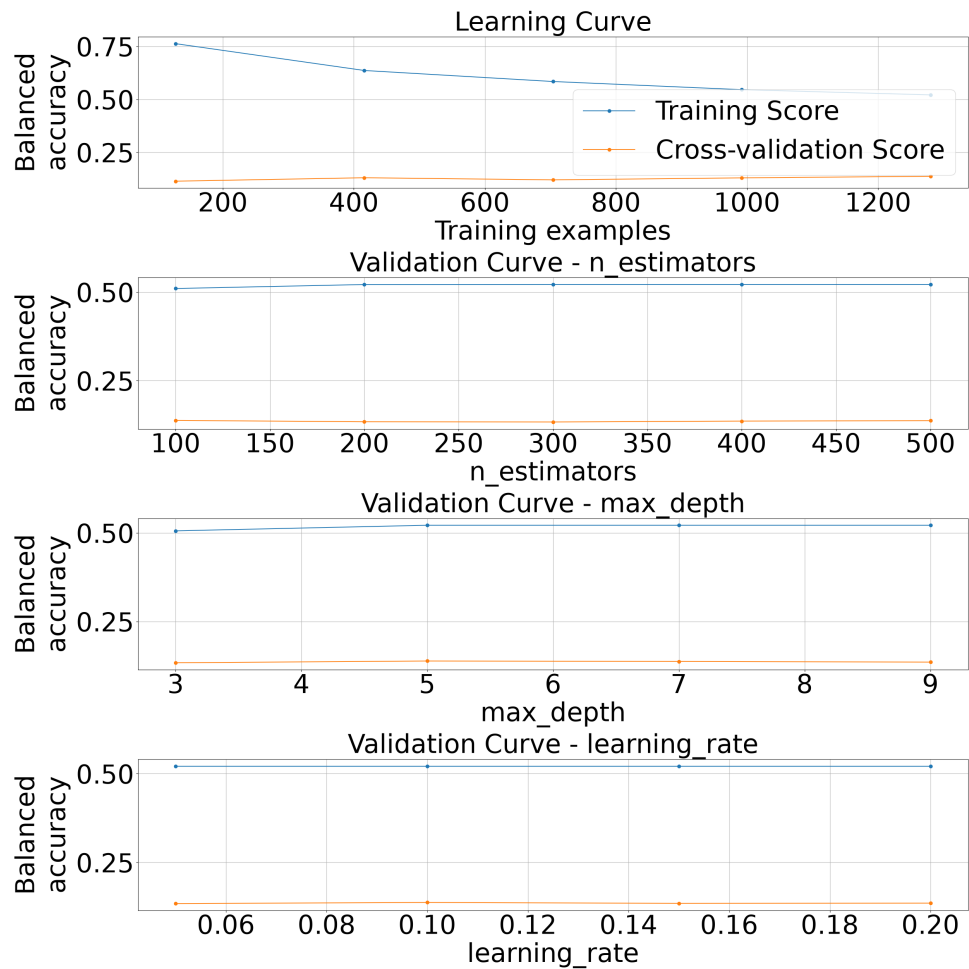


Figure 254 – XGBoost Learning & Validation Curves - CTV

Figure 255 presents the algorithm's ability to learn each individual class, from which is clear that the MLP algorithm was able to learn all classes quite well, having a little more trouble learning the "20-30 m^3/h " class, mistaking it for the "40-50 m^3/h ". Figure 256 presents how much each projected feature contributes to the MLP classification process, the fifth feature had the most impact followed by the eighth, third, and first, while the fourth, tenth, ninth and second had a least impact on the learning process. Finally, Figure 257 demonstrates that there is no indicative of overfitting and how the hyperparameters affects the method capability of learning the dataset.

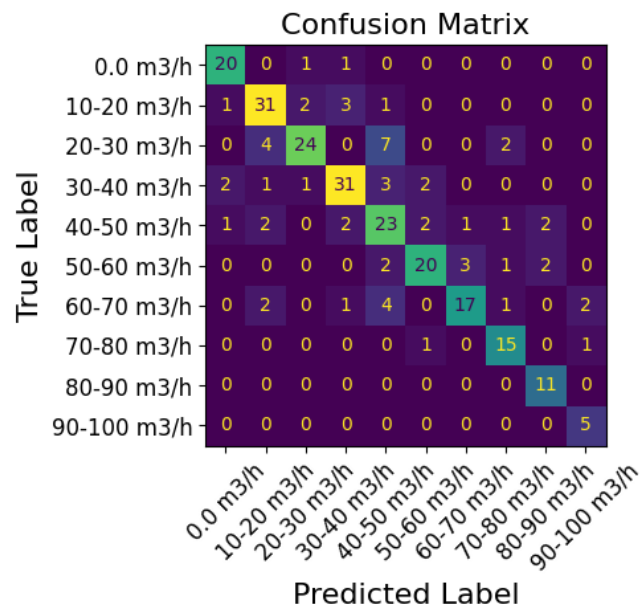


Figure 255 – MLP Confusion Matrix - CTV

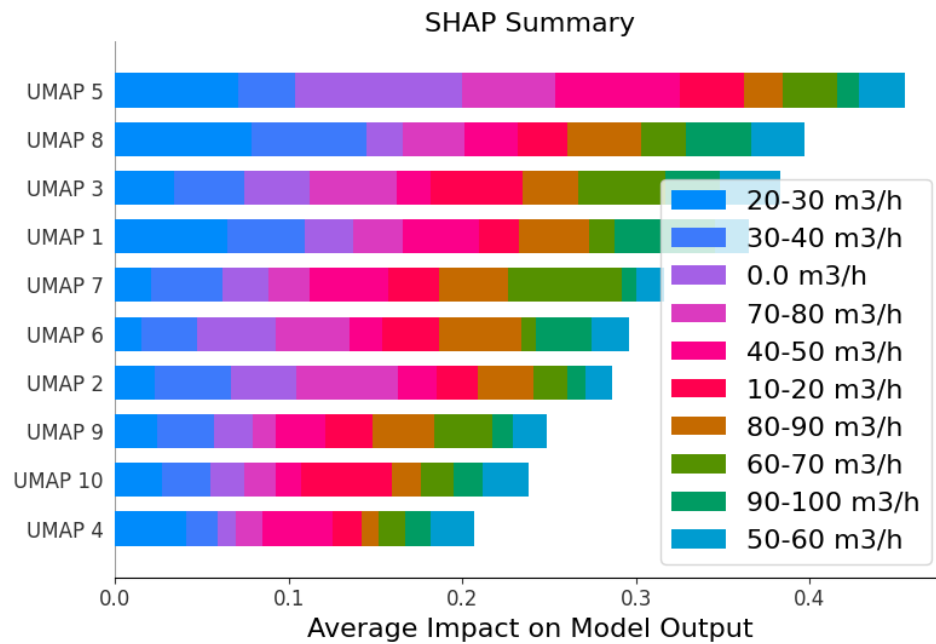


Figure 256 – MLP SHAP Values Summary - CTV

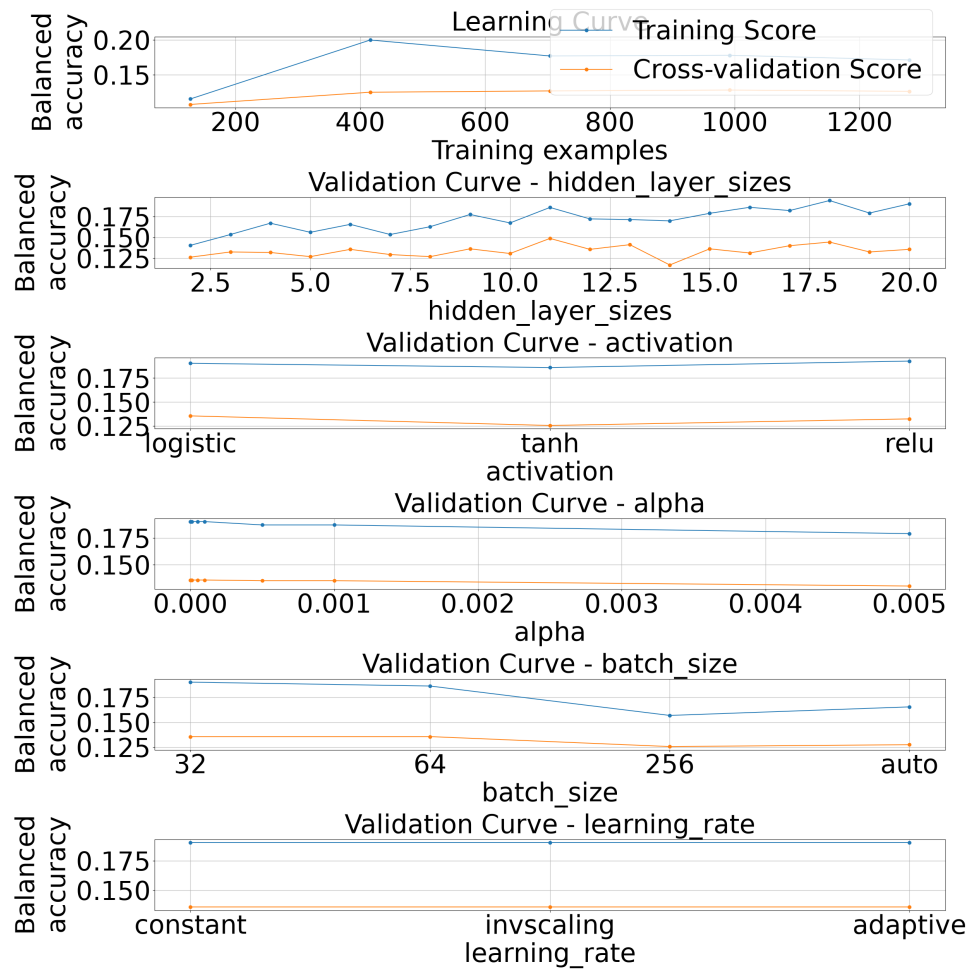


Figure 257 – MLP Learning & Validation Curves - CTV

	GMM	K-Means	HDBSCAN	Agglomerative	Spectral	Mean Shift
Silhouette	0.67	0.68	0.69	0.62	0.14	0.995
ARI	0.15	0.14	0.06	0.12	0.08	0.02
DBI	0.63	0.56	0.98	0.12	2.76	0.02
CH	964.74	1334.99	517.73	673.21	79.13	320648.38

Table 17 – Unsupervised Learning Results - CTV

4.2.4.2 Unsupervised Learning

Table 17 summarize the overall performance of all tested unsupervised algorithms. The consistently low ARI scores indicate poor alignment between the discovered clusters and the ground-truth flow rate labels, suggesting that the algorithms were not able to recover the flow rate structure in the feature manifold. However, the remaining metrics show that several methods still identified a clear and coherent structure in the data. This structure is primarily associated with the rotational speed, which is an expected outcome since the dominant spectral and dynamical components of the motor signals are directly tied to shaft rotation and its harmonics. As a result, the extracted Koopman features naturally exhibit larger variance with respect to RPM than to flow variations, leading the algorithms to form well-defined RPM-driven clusters.

The table shows that GMM's Silhouette score of 0.67, indicating that the algorithm formed well defined clusters – however, there's still some mixture with the other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in [Figure 258](#). [Figure 259](#) presents how much each projected feature contributes to the GMM classification process, the first feature had the most impact followed by the second, fourth, and twelfth, while the tenth, ninth and thirteenth had the least impact on the learning process.

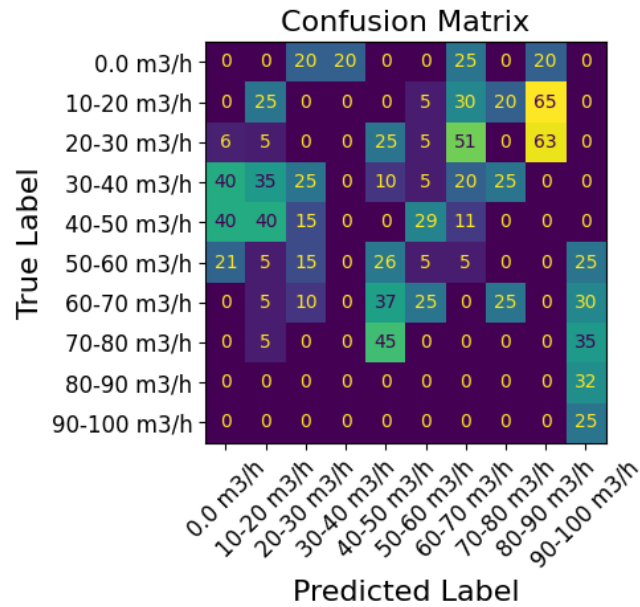


Figure 258 – GMM Confusion Matrix - CTV

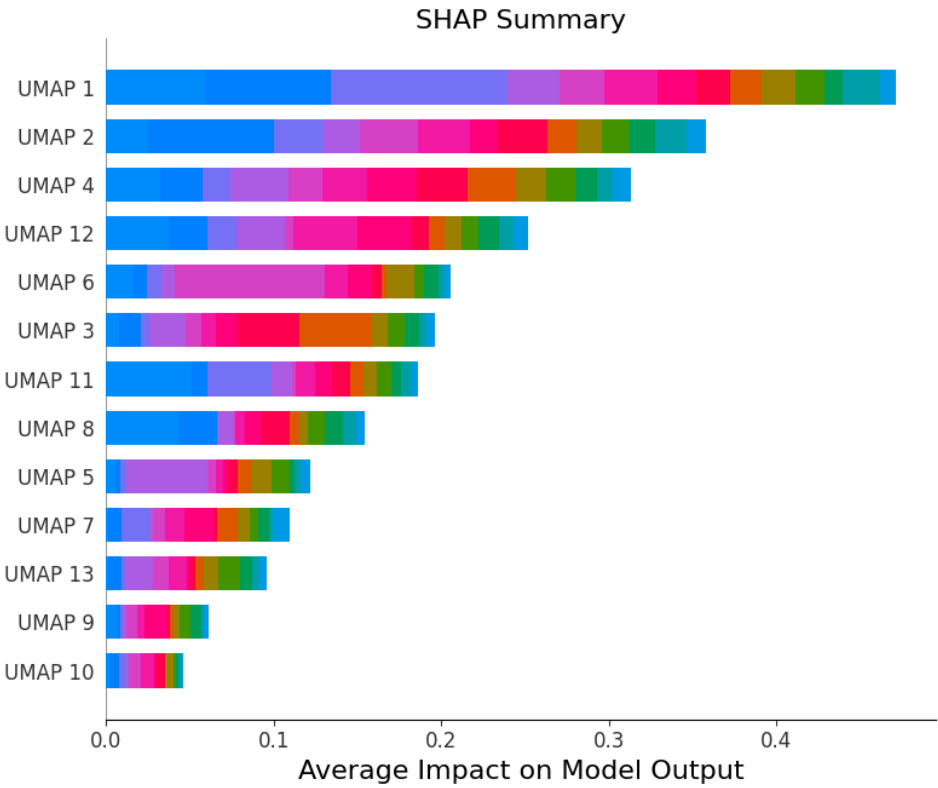


Figure 259 – GMM SHAP Values Summary - CTV

Kmeans reached a Silhouette score of 0.68, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 260. Figure 261 presents how much each projected feature contributes to the Kmeans classification process, the first feature had the most impact followed by the second, while the fifth, third and fourth had the least impact on the learning process.

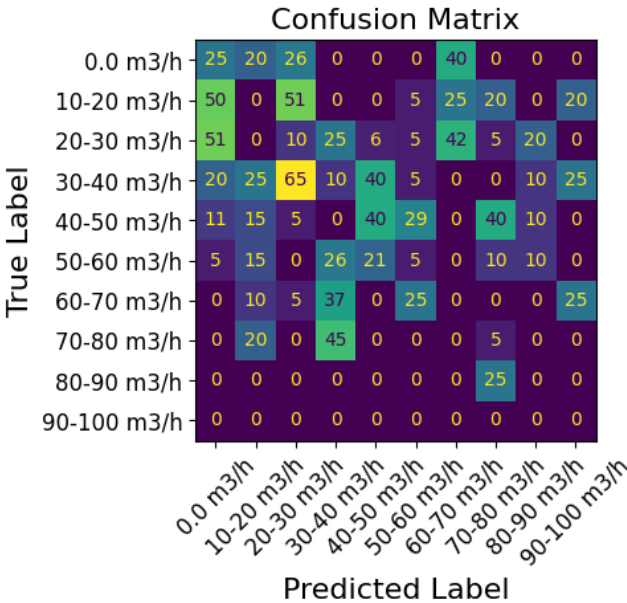


Figure 260 – KMeans Confusion Matrix - CTV

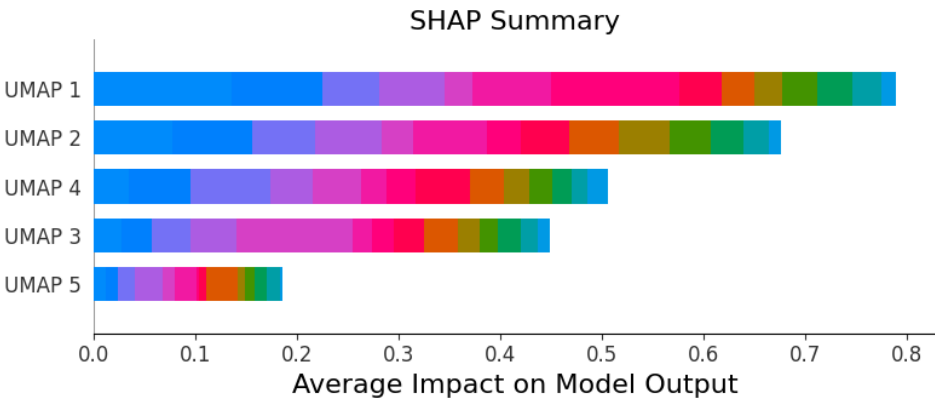


Figure 261 – KMeans SHAP Values Summary - CTV

HDBSCAN reached a Silhouette score of 0.69, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 262. Figure 263 presents how much each projected feature contributes to the HDBSCAN classification process, the optimization process picked only 4 projected features and both greatly contributed on the learning process.

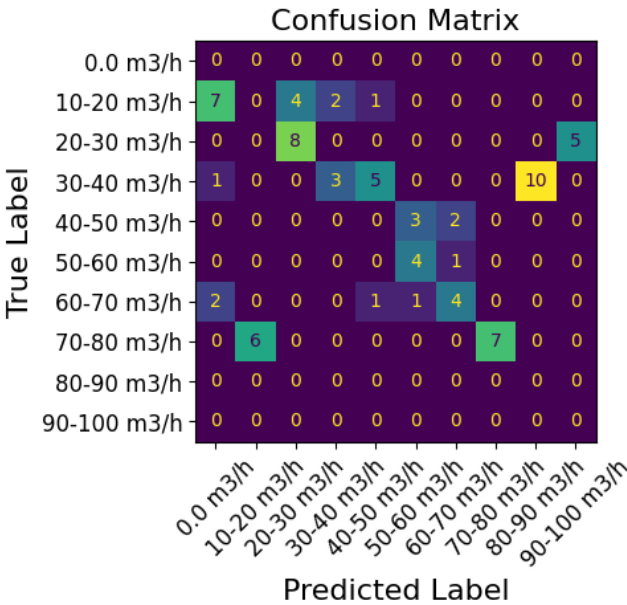


Figure 262 – HDBSCAN Confusion Matrix - CTV

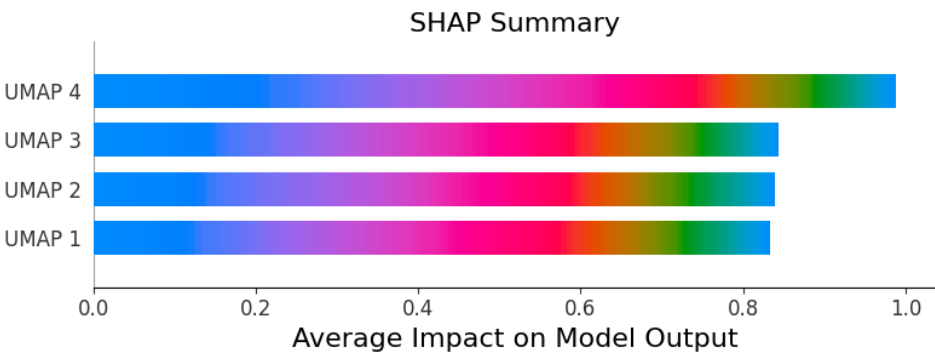


Figure 263 – HDBSCAN SHAP Values Summary - CTV

Agglomerative Clustering got a Silhouette score of 0.62, indicating that the algorithm formed well defined clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 264. Figure 265 presents how much each projected feature contributes to the Agglomerative Clustering classification process, the first feature had the most impact followed by the third, and second, while the sixth, eighth and fifth had the least impact on the learning process.

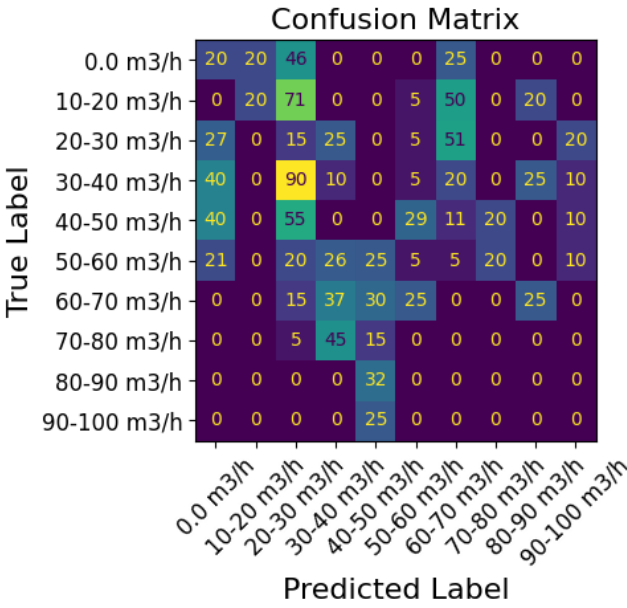


Figure 264 – Agglomerative Clustering Confusion Matrix - CTV

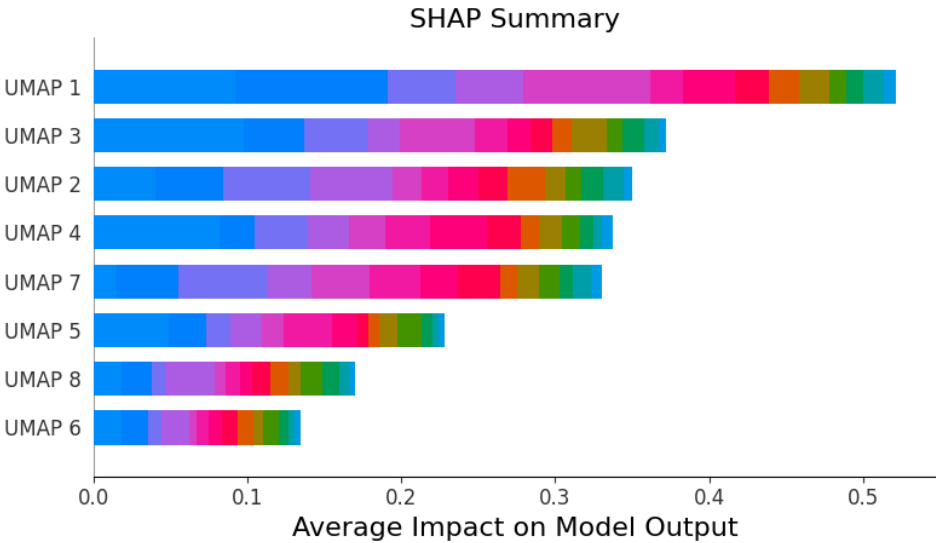


Figure 265 – Agglomerative Clustering SHAP Values Summary - CTV

The Spectral Clustering’s Silhouette score of 0.14, indicating that the algorithm that the algorithm had more trouble forming clusters, meaning that this method isn’t suited to cluster signals with this mixture, with other metrics re-enforcing that. The low ARI score tells the formed clusters don’t align well with the true classes, which is further confirmed by its confusion matrix in Figure 266. Figure 267 presents how much each projected feature contributes to the Spectral Clustering classification process, the first feature had the most impact followed by the second, third, and sixth, while the fourth, and fifth had the least impact on the learning process.

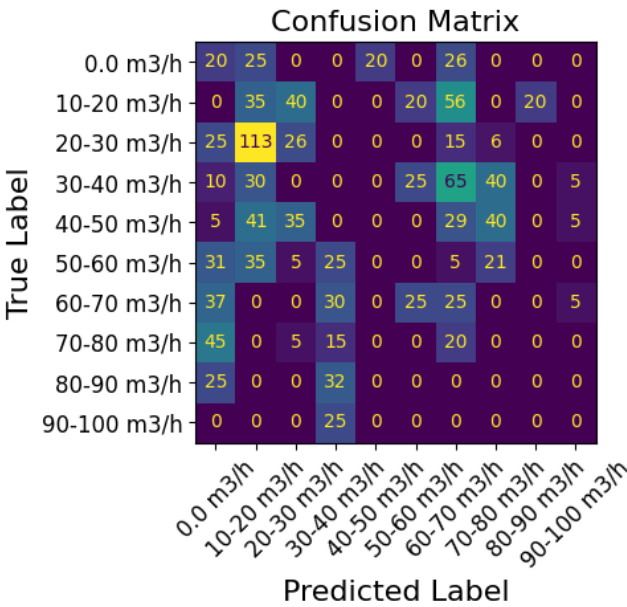


Figure 266 – Spectral Clustering Confusion Matrix - CTV

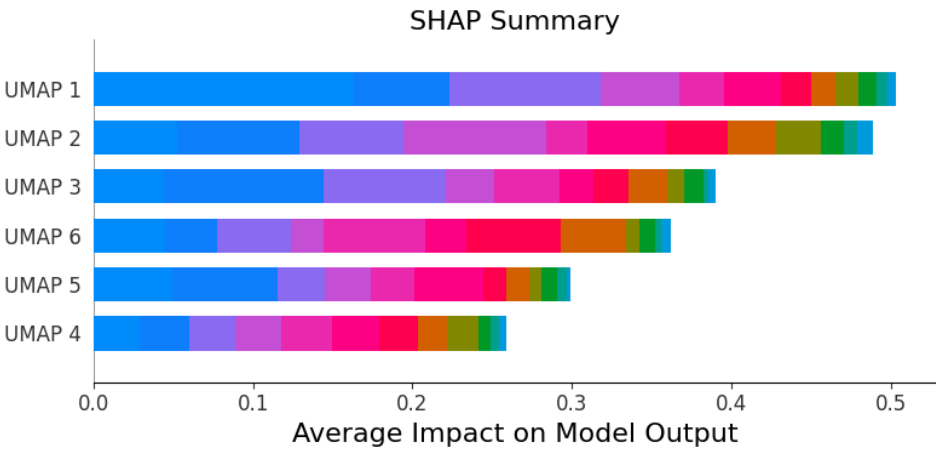


Figure 267 – Spectral Clustering SHAP Values Summary - CTV

Finally, the Mean Shift algorithm reached a Silhouette score of 0.995, indicating that the algorithm formed almost perfect clusters with other metrics re-enforcing that. The low ARI score tells the formed clusters don't align well with the true classes, which is further confirmed by its confusion matrix in Figure 268. Figure 269 presents how much each projected feature contributes to the Mean Shift classification process, the optimization process picked only 4 projected features and both greatly contributed on the learning process.

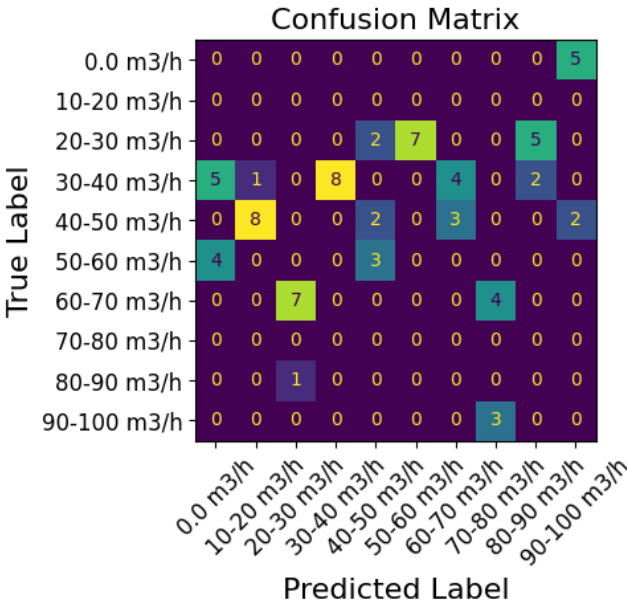


Figure 268 – Mean Shift Confusion Matrix - CTV

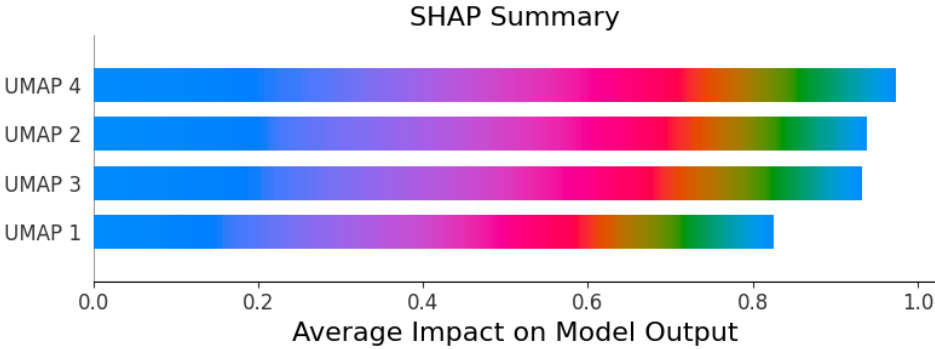


Figure 269 – Mean Shift SHAP Values Summary - CTV

5 Conclusion and Discussion

This chapter summarizes the main findings of this dissertation and discusses their implications for motor fault detection and flow rate classification using the proposed Koopman-based framework. The conclusions are presented progressively. First, the effectiveness of Koopman-derived features for motor fault detection is revisited and compared with conventional raw-signal and Fourier-based representations. Next, the performance of the framework for flow rate classification is examined across different sensing configurations, considering both supervised and unsupervised approaches to assess its ability to distinguish operating conditions and reveal the underlying structure of the extracted feature space. Finally, the main contributions, limitations, and directions for future research are discussed.

In the motor fault detection pipeline, the results show that, despite different amplitudes due to the different ways the Fourier and Koopman approaches measure the spectrum, both identify the same frequencies present in a signal and preserve the ratios between them so it guarantees a profound inspection and fault recognition by the fault frequencies methods established in the literature. While training the algorithms with the raw, Fourier Spectrum, and Koopman feature separately, the Koopman fed algorithms consistently outperform the others, followed by the Fourier and raw inputs, getting as high as 96.8% balanced cross validation accuracy and 80.7% balanced test accuracy, making it a quite reliable resource in fault detection.

In the fluid rate classification pipeline, experimental results demonstrated that the extracted Koopman features enable reliable discrimination of flow-rate classes under highly variable operation conditions. For only the 3 current signals, supervised classifiers achieved balanced cross-validation accuracies up to 96.1% and balanced test accuracies up to 90.2%, indicating strong generalization despite the complexity of the dataset. For the vibration signals, supervised classifiers achieved balanced cross-validation accuracies up to 88.2% and balanced test accuracies up to 89.2%, showing that the vibration signals capture almost all information captured by current signals, making possible to interchange the type of signal used in monitoring without much loss of precision. For the 3 current + 2 vibration signals, supervised classifiers achieved balanced cross-validation accuracies up to 94.3% and balanced test accuracies up to 92.6%, show a complementary characteristic between the two types of signals, meaning one type captures some information that the other don't so, when put together, the new dynamics added improves the overall accuracy of the classification algorithms. Now for only the combination of all measured signals, supervised classifiers achieved balanced cross-validation accuracies up to 96.3% and balanced test accuracies up to 89.3%, indicating that new information added by the voltage signals is

redundant and doesn't contribute much to the system's description, instead, these extra redundant features lead to overfitting and ultimately reduce overall accuracy. For all input combinations, the unsupervised algorithms further showed that the feature space preserves meaningful dynamical structure, revealing inherent organization related to pump operating regimes and rotational speeds, as can be seen by the Mean Shift's consistent higher or equal to 0.90 Silhouette score.

The SHAP analysis further revealed that only specific regions of the Koopman-UMAP latent manifold consistently drive classification and clustering performance. This suggests that future improvements in motor fault and flow rate discrimination might be achieved by explicitly optimizing the Koopman feature selection and dimensionality reduction stages to enhance the separability of these regions, potentially improving both classification accuracy and lower influence of the VFD dynamics. It is important to note that, due to the nonlinear and nonparametric nature of UMAP, the contribution of individual Koopman spectral components to specific regions of the latent manifold cannot be directly quantified. Consequently, while SHAP analysis reveals which latent dimensions are most relevant for downstream models, tracing these effects back to specific Koopman features remains an open challenge.

A point worth highlighting is the adequacy of the available dataset size. Although the motor fault database comprises only 106 samples of operating conditions, the sample-versus-learning curves obtained for the proposed classifiers exhibited a clear tendency toward performance saturation, indicating diminishing gains as additional samples are incorporated into the training process. This behaviour suggests that the current dataset already captures a significant portion of the variability associated with the investigated operating conditions and that the models are approaching their practical learning capacity under the present feature representation. Therefore, while larger datasets would always be beneficial for improving generalization and robustness, the observed learning plateaus indicate that the available number of samples is sufficient to support the conclusions drawn in this work.

An additional observation concerns the relatively limited contribution of voltage measurements to the flow classification task. Since stator voltages primarily act as externally imposed excitation inputs and remain comparatively insensitive to load variations, they naturally contain weaker flow-dependent signatures than stator currents and vibration measurements, both of which directly respond to changes in hydraulic loading and torque demand. Consequently, incorporating voltage channels increase the amount of correlated and redundant information within the feature space. From a conventional machine learning perspective, such redundancy could potentially increase model complexity, promote overfitting, and degrade generalization performance. However, the experimental results indicate that classification performance remains remarkably stable even after intro-

ducing voltage measurements. This behaviour confirms that the proposed Koopman-based feature extraction pipeline effectively acts as a dynamical information filtering mechanism, preserving the physically relevant system dynamics while suppressing redundant or weakly informative components. Therefore, the relatively unchanged performance across sensor configurations provides additional evidence that the extracted Koopman representations are robust physically meaningful, and largely insensitive to increases in measurement redundancy.

From a field deployment perspective, the proposed methodology is well suited for practical implementation, as it relies exclusively on motor related measurements that are already available in most industrial motor drive systems. Feature extraction can be performed offline during model development and deployed online as a lightweight inference module and a computational complexity of $O(dN) + O(d^2N) + O(r^2N) + O(r^3N)$ with d being the embedding dimension, r the retained rank, and N the signal length (worth to mention that the expression can be reduced to its dominant term $O(d^2N)$), enabling real-time or near-real-time monitoring without additional sensing infrastructure. The resulting models can be integrated into existing condition monitoring or supervisory control systems to provide continuous assessment of motor fault detection, flow regimes, and operating states, particularly in applications where direct flow measurement is impractical, unreliable, or cost-prohibitive.

These findings confirm that Koopman spectral analysis provides a robust and interpretable representation of the system spectrum, enabling fault detection, pump dynamics, capable of capturing stationary, transient, and multiphase flow effects within a unified framework. The proposed approach offers a viable alternative to conventional Fourier-based monitoring techniques, particularly in scenarios where operating conditions vary significantly or direct flow measurements are impractical. Overall, this work highlights the potential of Koopman-based feature extraction as a foundation for advanced condition monitoring and flow regime identification in complex industrial pumps.

5.1 Study Limitations & Future Works

One important limitation of the present study is associated with the intrinsic spectral characteristics of inverter-fed induction motors. In such systems, the measured electrical and mechanical signals are strongly dominated by the fundamental excitation frequency and its harmonics components, which typically contain a significant portion of the total signal energy. Conversely, fault-related signatures and flow-induced load variations generally appear as secondary dynamical modulations superimposed onto these dominant components, often exhibiting substantially lower amplitudes.

This disparity in energy distribution introduces an inherent signal-to-background

challenge. Since the fundamental frequency and its harmonics represent the primary source of variance in the measured signals, they naturally dominate both the Fourier and Koopman spectra. As operating conditions change, particularly under variable-speed regimes, these dominant components shift across the frequency domain, potentially masking or distorting the weaker signatures associated with faults or hydraulic load variations. Consequently, subtle dynamical patterns become increasingly difficult to separate and identify, especially when their spectral contributions overlap with harmonics of the fundamental excitation.

Importantly, this limitation should not be interpreted as a deficient of the proposed Koopman framework. Rather, it reflects an inherent characteristic of the sensing modality and of induction motor operation itself. The extracted Koopman features preserve the natural dynamical organization of the system, including the dominance of rotational dynamics, as evidenced by the unsupervised analysis performed in this work. Therefore, future research should focus on developing methodologies capable of attenuating the influence of dominant excitation components while preserving physically meaningful fault and flow-related information.

Another promising research direction concerns the application of Design of Experiments (DOE ([Design of Experiments](#))) methodologies to the data acquisition stage. Rather than relying on empirically selected operating conditions, future studies could employ statistically designed experiments to systematically determine the combinations of rotational speed, load conditions, gas injection rates, and fault severities that maximize information content while minimizing experimental effort. Such approaches may help identify redundant operating points, improve dataset representativeness, and provide a more rigorous assessment of the minimum number of samples required for reliable model training and validation. Another strategy could be focused on exploring band rejection to filter out the inverter's fundamental frequency and harmonics, or speed normalization techniques to mitigate the speed dependence in the Koopman spectrum. From a learning perspective, future studies focused on hierarchical and multi-task classification strategies and coarsed/physics informed clusterization or feature construction algorithms to explicitly decouple speed from the wanted information, incorporating physical constraints, operating regimes, and process continuity could provide additional insights on pump and motor fault behaviour; as well as deeper and more specialized neural networks architectures trained directly on Koopman-derived representations is also recommended. Finally, validation on larger, more diverse datasets, covering broader operating envelopes and transient regimes, will be essential to assess scalability, generalization, and practical deployment in real industrial monitoring systems.

Also, future work investigating algorithmic strategies to directly optimize Koopman spectral selection and manifold embedding based on latent-space interpretability

analysis. In particular, methods that stabilize and enhance the most discriminative regions of the Koopman-UMAP manifold may help further distinguishably regions of the additional sensing hardware. Developing methodologies to bridge this interpretability gap between Koopman spectral features and nonlinear manifold embeddings represents a promising direction for future research.

Bibliography

- [1] SUN, T.; WANG, R.; ZENG, Z. Investigations of rotational speed and flow rate on centrifugal pump performance using cfd. *IEEE 13th International Conference on Mechanical and Intelligent Manufacturing Technologies*, 2016. [21](#), [22](#)
- [2] XU, W.; WEI, D.; LEI, C. Control for the centrifugal pump in the simulation platform of power plants. *International Conference on Industrial Informatics*, 2016. [21](#), [22](#)
- [3] NIU, G.; CHEN, Y. Motor fault diagnostics based on current signatures: A review. *IEEE Trans. on Instrumentation and Measurement*, v. 72, 2023. [21](#), [23](#), [27](#), [28](#)
- [4] RECH, E. V. F.; OLIVEIRA, L. E. de L. de; SANT'ANA, W. C. Detection of nonstationary events in motor signals using nonparametric statistical hypothesis testing. *IEEE Access*, v. 13, 2025. [21](#), [27](#), [48](#)
- [5] HURTADO, Z. Y. M.; TELLO, C. P.; SARDUY, J. G. A review on detection and fault diagnosis in induction machines. *Publicaciones en Ciencias y Tecnología*, v. 8, p. 11–30, 2014. [21](#), [23](#)
- [6] KUMAR, S. et al. A comprehensive review of condition based prognostic maintenance (cbpm) for induction motor. *IEEE Access*, 2019. [21](#), [27](#), [28](#)
- [7] SALOMON, C. P. et al. Electrical signature analysis for condition monitoring of permanent magnet synchronous machine. *Advances in Electrical and Computer Engineering*, v. 18, 2018. [21](#), [27](#), [28](#)
- [8] MORINIGO-SOTELO, D. et al. Reliable detection of rotor bars breakage in induction motors via music and zsc. *IEEE Trans. on Industry Applications*, v. 54, 2018. [21](#)
- [9] QIN, H.; NI, H.; JIN, J. Differential evolutionary modeling and starting-up cavitation characteristics analysis for multistage centrifugal pump. *IEEE 4th Information and Mechatronics Engineering Conference (ITOEC)*, 2018. [21](#), [22](#)
- [10] JIANG, Y. et al. A flow rate estimation method for gas-liquid two-phase flow based on transformer neural networks. *IEEE Sensors Journal*, v. 24, 2024. [21](#), [22](#), [23](#)
- [11] GEVORKOV, L.; VODOVOZOV, V. Mixed pressure control system for a centrifugal pump. *11th IEEE International Conference on Compatibility, Power Electronics and Power Engineering*, 2017. [21](#), [22](#)
- [12] JINASENA, A.; SHARMA, R. Model based real-time flow rate estimation in open channels with application to conventional drilling. *International Conference on Control, Automation and Systems*, 2018. [22](#)
- [13] GARCIA-BRACAMONTE, J. E. et al. An approach on mcsa-based fault detection using independent component analysis and neural networks. *IEEE Transactions on Instrumentation and Measurement*, v. 68, 2019. [22](#)

- [14] SONG, X.; WANG, Z.; HU, J. Detection of bearing outer race fault in induction motors using motor current signature analysis. *International Conference on Electrical Machines and Systems*, v. 22, 2019. [22](#), [27](#)
- [15] ZAMUDIO-RAMIREZ, I. et al. Detection of uniform gearbox wear in induction motors based on the analysis of stray flux signals through statistical time-domain features and dimensionality reduction techniques. *IEEE Transactions on Industry Applications*, v. 58, 2022. [22](#)
- [16] YELPALE, P. D.; JARIAL, R. K.; PATIL, A. J. Fuzzy-based induction motor fault diagnosis decision-making system for motor current signature analysis. *International Conference for Innovation in Technology*, 2024. [22](#), [27](#)
- [17] LIU, D.; INOUE, H.; KANEMARU, M. Robust motor current signature analysis (mcsa)-based fault detection under varying operation conditions. *International Conference on Electrical Machines and Systems*, 2022. [22](#), [27](#), [28](#)
- [18] SHEN, Z.; SCHMOLL, R.; KROLL, A. Measurement of fluid flow velocity by using infrared and visual cameras: Comparison and evaluation of optical flow estimation algorithms. *IEEE Sensors*, 2023. [22](#)
- [19] DINARDO, G.; FABBIANO, L.; VACCA, G. Fluid flow rate estimation using acceleration sensors. *Seventh International Conference on Sensing Technology*, 2013. [22](#)
- [20] LI, H.; LI, H.; CHEN, Y. Fusion of doppler and pair-wise optical flow with motion compensation for ultrasound velocity field estimation. *IEEE International Ultrasonics Symposium (IUS)*, 2025. [23](#)
- [21] KOOPMAN, B. O. Hamiltonian systems and transformations in hilbert space. *Department of Mathematics, Columbia University*, 1931. [23](#), [30](#)
- [22] KORDA, M.; MEZIC, I. Learning koopman eigenfunction for prediction and control: the transient case. *arXiv:1810.08733v1*, 2022. [23](#), [30](#)
- [23] BRUCE, A. L.; ZEIDAN, V. M.; BERNSTEIN, D. S. What is the koopman operator? a simplified treatment for discrete-time systems. *American Control Conference (ACC)*, 2019. [23](#), [30](#)
- [24] RAJENDRAN, S. et al. Dynamic mode decomposition based fault diagnosis in three-phase electrical machines. *Results in Engineering ScienceDirect*, 2024. [23](#), [31](#), [32](#)
- [25] NETTO, M.; MILI, L. A robust data-driven koopman kalman filter for power systems dynamic state estimation. *IEEE Transactions on Power Systems*, v. 33, 2018. [23](#), [31](#), [32](#)
- [26] WANG, E. et al. Koopman-informed neural network for machinery's nonlinear dynamics learning and remaining useful life prediction. *IEEE Transactions on Instrumentation and Measurement*, v. 74, 2025. [23](#), [31](#), [32](#)
- [27] CLAINCHE, S. L.; VEGA, J. M. Higher order dynamic mode decomposition. *SIAM*, 2017. [23](#), [32](#), [33](#)
- [28] ZHANG, Y.; GAO, S.; JIANG, L. High-order dynamic mode decomposition for multidimensional harmonic retrieval. *Asia-Pacific International Symposium and Exhibition on Electromagnetic Compatibility (APEMC)*, 2025. [23](#), [32](#), [33](#)

- [29] CLAINCHE, S. L.; HAN, Z.-H.; FERRER, E. An alternative method to study cross-flow instabilities based on high order dynamic mode decomposition. *Physics of Fluids* 31, 2019. [23](#), [32](#), [33](#)
- [30] RECH, E. V. F.; OLIVEIRA, L. E. de L. de; SANT'ANA, W. C. Flow rate classification in industrial pump systems using koopman-based feature extraction and machine learning. *IEEE Access*, v. 14, 2026. [23](#), [29](#)
- [31] SUSUKI, A. C. Y. Introduction to koopman mode decomposition for data-based technology of power system nonlinear dynamics. *International Federation of Automatic Control (IFAC)*, 2018. [23](#), [30](#), [31](#)
- [32] MASBAHI, J. B. A.; MESBAHI, M. Nonlinear observability via koopman analysis: Characterizing the role of symmetry. *Elsevier*, 2021. [23](#)
- [33] BAROCIO, B. C. B. E.; NINA, F. A dynamic mode decomposition framework for global power system oscillation analysis. *IEEE Transactions on Power Systems*, 2015. [23](#)
- [34] SUSUKI, R. H. Y.; ISHIGAME, A. Estimation of power system inertia using nonlinear koopman modes. *IEEE Power & Energy Society General Meeting (PESGM)*, 2018. [23](#)
- [35] CHENG, J. S. H.-L.; LEHTONEN, M. Data-driven modal characterization of converter-driven oscillation from synchronized waveform measurement using dynamic mode decomposition. *Elsevier*, 2026. [23](#)
- [36] RAJENDRAN R. SREEHESH, V. S. K. D. D. J. S.; BANJERDPONGCHAI, D. Dynamic mode decomposition based fault diagnosis in three-phase electrical machines. *Elsevier*, 2025. [24](#)
- [37] SALOMON, C. P. et al. A study of fault diagnosis based on electrical signature analysis for synchronous generators predictive maintenance in bulk electric system. *MDPI*, 2019. [27](#)
- [38] BROER, H. W.; TAKENS, F. Mixed spectra and rotational symmetry. *University of Groningen*, 1993. [29](#)
- [39] BARANSKI, K.; GUTMAN, Y.; SPIEWAK, A. A probabilistic takens theorem. *London Mathematical Society*, 2020. [29](#)
- [40] WANG, Y. et al. The hankel matrix solution to a system of quaternion matrix equations. *Chinese Control And Decision Conference (CCDC)*, 2020. [29](#)
- [41] LU, H. et al. A low rank hankel matrix reconstruction method for ultrafast magnetic resonance spectroscopy. *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017. [29](#)
- [42] DANGO I. KAMWA, H. G. A. N. A. A.; MASOOM, A. Higher-order dynamic mode decomposition to identify harmonics in power systems. *MDPI*, 2025. [30](#), [31](#)
- [43] LEE, T.-W.; PARK, J. E. Entropy and turbulence structure. *MDPI*, 2022. [31](#)
- [44] JAEN-CUELLAR J. J. SAUCEDO-DORANTES, D. A. E.-O. A. Y.; ROMERO-TRONCOSO, R. de J. Evaluation of entropy analysis as a fault-related feature for detecting faults in induction motors and their kinematic chain. *MDPI*, 2024. [31](#)

- [45] GUNAWAN, A. B. M.; HASMAWATI; IHSAN, A. F. Implementation of the k-nearest neighbor algorithm for forecasting the operational condition of natural gas pipeline transmission networks. *International Conference on Data Science and Its Applications (ICoDSA)*, 2024. 37
- [46] MLADENOVA, T.; VALOVA, I. Comparative analysis between the traditional k-nearest neighbor and modifications with weight-calculation. *International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2022. 37
- [47] PIETRZAK, P.; WOLKIEWICZ, M. Application of support vector machine to stator winding fault detection and classification of permanent magnet synchronous motor. *19th International Power Electronics and Motion Control Conference (PEMC)*, 2021. 37
- [48] KUYORO, A. O.; ALIM, S.; AWODELE, O. Comparative analysis of the performance of various support vector machine kernels. *5th Information Technology for Education and Development (ITED)*, 2022. 37
- [49] YILING, H.; SHAOFENG, H. A short-term load forecasting model based on improved random forest algorithm. *5th International Forum on Electrical Engineering and Automation (IFEAA)*, 2020. 38
- [50] MARTINS, W. et al. Fpga-based implementation of random forest classifier for semg signal classification. *46th Annual Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2024. 38
- [51] BHUVANA, R.; MAHESHWARI, S.; SASIKALA, S. Predict the heart disease using a logistic regression classifier algorithm. *2th International Conference on System Modeling & Advancement in Research Trends*, 2023. 38
- [52] KUMAR, S. Adaptive logistic regression: Leveraging arctan in active learning. *10th International Conference on Signal Processing and Communication (ICSC)*, 2025. 38
- [53] PRADHAN, R. et al. Fault detection in ac microgrid using gaussian naive bayes: A probabilistic approach. *4th International Conference on Sustainable Energy and Future Transportation (SEFET)*, 2024. 38
- [54] KELLY, A.; JOHNSON, M. A. Investigating the statistical assumptions of naive bayes classifiers. *55th Annual Conference of Information Sciences and Systems (CISS)*, 2021. 38
- [55] HUSSIAN, R.; SALEH, M. A.; REFAAT, S. S. Induction motor multi incipient fault detection based on gradient boosting algorithms. *50th Annual Conference of the IEEE Industrial Electronics Society*, 2024. 39
- [56] KONKIN, N. A. Determination of optimal maximum usable frequency prediction periods for ionospheric radio channels using the xgboost machine learning algorithm. *System of Signals Generating and Processing in the Field of on Board Communications*, 2023. 39
- [57] RAHMANI, O.; ILTARABIAN, M.; SADROSSADAT, S. A. Modeling and simulation of speed and efficiency of bldc motor as a starter motor based on multilayer perceptron (mlp) neural network. *IEEE International Transportation Electrification Conference & EXPO Asia-Pacific*, 2018. 39

- [58] RAY, S.; MATHEW, E. C. Mlp neural network aided power plant controller to mitigate oscillations in renewable energy plant. *International Joint Conference on Neural Networks (IJCNN)*, 2025. [39](#)
- [59] OUACHTOUK, I.; HANI, S. E.; DAHI, K. Detection and classification of broken rotor bars faults in induction machine using k-means classifier. *2nd International Conference on Electrical and Information Technologies (ICEIT)*, 2016. [39](#), [40](#)
- [60] BHADANA, A.; SINGH, M. Fusion of k-means algorithm with dunn's index for improved clustering. *2nd International Conference on Electrical and Information Technologies for Sustainable Solutions*, 2017. [39](#), [40](#)
- [61] ZENG, Y. et al. Probabilistic power flow method for dc distribution systems based on gaussian mixture model. *10th Asia Conference on Power and Electrical Engineering (ACPEE)*, 2025. [40](#)
- [62] WANG, Z. et al. Fault assessment of hydraulic system based on gaussian mixture model. *Global Reliability and Prognostics and Health Management (PHM-Shanghai)*, 2020. [40](#)
- [63] TANG, Z. et al. Short-term wind power prediction based on hdbscan with outlier factor method and cnn-bilstm. *IEEE IAS Industrial and Commercial Power System Asia*, 2023. [40](#), [41](#)
- [64] ZHOU, H. et al. Exploring hdbscan for effort-aware defect prediction: A density-based clustering method. *6th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*, 2024. [40](#), [41](#)
- [65] BABICHEV, S.; TAIF, M. A.; LYTVYNENKO, V. Inductive model of data clustering based on the agglomerative hierarchical algorithm. *IEEE First International Conference on Data Stream Mining & Processing*, 2016. [41](#)
- [66] DAI, W.; ZHANG, M. Hierarchical agglomerative clustering optimization for massive data. *IEEE Smart World Congress (SWC)*, 2024. [41](#)
- [67] HUANG, M. et al. Modeling and analysis for power substation load data based on spectral clustering. *4th International Electrical and Energy Conference (CIEEC)*, 2021. [41](#), [42](#)
- [68] GAO, Y.; LUO, R.; YANG, W. Fast spectral clustering based on fusion density peaks. *7th International Conference on Big Data and Artificial Intelligence*, 2024. [41](#), [42](#)
- [69] AHMAD, A. et al. Vehicle recognition using multi-layer perceptron and smote technique. *2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, 2022. [42](#)
- [70] DEY, I.; PRATAP, V. A comparative study of smote and borderline-smote and adasyn oversampling techniques using different classifiers. *3rd International Conference on Smart Data Intelligence (ICSMDI)*, 2023. [42](#)
- [71] DENG, Y. et al. Umap for dimensionality reduction in sleep stage classification using eeg data. *46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2024. [43](#)

- [72] PAL, K.; SHARMA, M. Performance evaluation of non-linear techniques umap and t-sne for data in higher dimensional topological space. *Proceedings of the Fourth International Conference on t-SMAC*, 2020. [43](#)
- [73] WATANABE, S. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv:204.11127v4*, 2025. [44](#)
- [74] OZAKI Y. TANIGAKI, S. W. Y.; ONISHI, M. Multiobjective tree-structured parzen estimator for computationally expensive optimization problems. *GECCO*, 2020. [44](#)
- [75] JAMIESON, K.; TALWALKAR, A. Non-stochastic best arm identification and hyperparameter optimization. *Journal of Machine Learning*, 2016. [44](#)
- [76] VARGAS, R. R. de; BEDREGAL, B. R. C. A way to obtain the quality of a partition by adjusted rand index. *2nd Workshop-School on Theoretical Computer Science*, 2013. [44](#)
- [77] BALAMURALI, A.; KUMAR, K. V. Early detection and classification of type-2 diabetes using stratified k-fold validation. *International Conference on Innovative Computing Intelligent Communication and Smart Electrical Systems (ICSSES)*, 2024. [45](#)
- [78] YADAV, S.; SHUKLA, S. Analysis of k-fold cross-validation over hold-out validation on colossal dataset for quality classification. *6th International Conference on Advanced Computing*, 2016. [45](#)
- [79] RIBEIRO, S. S. M. T.; GUESTRIN, C. Why should i trust you? explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. [45](#)
- [80] STRUMBELJ, E.; KONONENKO, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 2014. [45](#)
- [81] POWERS, D. M. W. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *School of Informatics and Engineering - Flinders University*, 2007. [46](#)
- [82] GONZALEZ-VILLANUEVA, R.; CARRANZA-ROJAS, J. Improving balanced accuracy for minority plant species under data imbalance: A multi-architectural ensemble approach. *5th International Conference on BioInspired Processing (BIP)*, 2023. [46](#)
- [83] SHAHAPURE, K. R.; NICHOLAS, C. Cluster quality analysis using silhouette score. *7th International Conference on Data Science and Advanced Analytics (DSAA)*, 2020. [47](#)
- [84] TATIKONDA, R. et al. Customer behaviour analysis for e-commerce and retail using improved k-means clustering with calinski harabasz index. *International Conference on Intelligent Computing and Knowledge Extraction (ICICKE)*, 2025. [47](#)
- [85] VERGANI, A. A.; BINAGHI, E. A soft davies-bouldin separation measure. *International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018. [47](#)
- [86] VARGAS, R. R. de; BEDREGAL, B. R. C. A way to obtain the quality of a partition by adjusted rand index. *2nd Workshop-School on Theoretical Computer Science*, 2013. [47](#)

A Attachments

A.1 Published Papers

- A.1.1 E. V. F. Rech, L. E. de L. de Oliveira, and W. C. Sant'Ana, "Detection of Nonstationary Events in Motor Signals Using Nonparametric Statistical Hypothesis Testing", IEEE Access, vol. 13, 29/09/2025
- A.1.2 E. V. F. Rech, L. E. de L. de Oliveira, and W. C. Sant'Ana, "Flow Rate Classification in Industrial Pump Systems Using Koopman-Based Feature Extraction and Machine Learning- IEEE Access, accepted for publication on 26/05/2026

A.2 Submitted Papers

- A.2.1 E. V. F. Rech, L. E. de L. de Oliveira, and W. C. Sant'Ana, "Electric Motor Fault Detection under Nonstationary Operation using Koopman Operator Spectral Analysis- Submitted to IEEE Access, 16/05/2026

Received 15 August 2025, accepted 22 September 2025, date of publication 29 September 2025, date of current version 3 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3615553



RESEARCH ARTICLE

Detection of Nonstationary Events in Motor Signals Using Nonparametric Statistical Hypothesis Testing

EDUARDO VAZ FAGUNDES RECH^{1,2}, LEVY ELY DE LACERDA DE OLIVEIRA^{1,2}, AND WILSON CESAR SANT'ANA^{1,3}, (Senior Member, IEEE)¹Office of Research and Graduate Studies (PRPPG—PPGEEL), Federal University of Itajubá (UNIFEI), Itajubá 37500-903, Brazil²Research and Development Department, Gnarus Institute, Itajubá 37502-485, Brazil³Institute of Systems Engineering and Information Technology (IESTI), UNIFEI, Itajubá 37500-903, Brazil

Corresponding author: Wilson Cesar Sant'Ana (wilson_santana@ieee.org)

The Article Processing Charge (APC) for the publication of this research was funded by the Coordination for the Improvement of Higher Education Personnel (CAPES) (ROR ID: 00x0ma614). For open access purposes, the authors have assigned the Creative Commons CC BY license to any accepted version of the manuscript.

ABSTRACT This work presents a lightweight statistical method for detecting nonstationarities in motor current signals, aiming to improve the diagnosis of transient events such as sudden load changes or inverter disturbances. Unlike traditional MCSA techniques that rely on spectral analysis, the proposed framework segments the signals into non-overlapping windows and extracts global statistical features to capture distributional changes over time. These features are compared using nonparametric hypothesis tests with p-value fusion strategies, enhancing robustness, with an adaptive buffering scheme based on Wasserstein distance to allow the system to adjust to signal variability. Results indicates accuracy reaches values as high as 95.9% with the false positive and false negative rates go as low as 7.7% and 0% respectively for both Monte Carlo simulations and real-world tests (from 529 MCSA signals), demonstrating strong potential for real-time, embedded, and scalable motor health monitoring without requiring complex signal decomposition or machine learning models.

INDEX TERMS Adaptive buffer, condition monitoring, hypothesis testing, induction motors, motor current signature analysis (MCSA), Monte Carlo simulation, nonstationary detection, predictive maintenance, P-value fusion, Wasserstein distance.

I. INTRODUCTION

Monitoring and diagnosing the condition of electric motors is essential to ensuring the reliability and operational continuity of industrial processes. Electric motor faults can lead to costly downtime, damage to equipment, and safety hazards [1]. Among various monitoring techniques, Electrical Signature Analysis (ESA), particularly Motor Current Signature Analysis (MCSA), has gained prominence due to its noninvasive nature and ability to detect a wide range of faults such as broken bars, eccentricity, and winding short-circuits [2].

However, traditional MCSA techniques often rely on frequency domain tools like the Fourier Transform, which

assumes the signal is stationary [3]. This assumption implies that the statistical characteristics of the signal - such as its mean, variance and higher-order moments - remain constant over time [4]. In reality, motor signals are often nonstationary due to mechanical load changes, power supply fluctuations, or system transients.

These nonstationarities present a fundamental challenge to spectral methods, resulting in distorted spectral representations such as frequency leakage, side lobes, and ambiguous harmonic content. This issue leads to reduced sensitivity and accuracy in fault detection [3].

Recent MCSA-based fault detection methods for induction motors have explored advanced signal processing and intelligent classification techniques. Garcia-Bracamonte et al. combined Independent Component Analysis and Neural

The associate editor coordinating the review of this manuscript and approving it for publication was Pinjia Zhang ⁵.

Received 1 May 2026, accepted 23 May 2026, date of publication 26 May 2026, date of current version 29 May 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3697346



RESEARCH ARTICLE

Flow Rate Classification in Industrial Pump Systems Using Koopman-Based Feature Extraction and Machine Learning

EDUARDO VAZ FAGUNDES RECH^{1,2}, LEVY ELY DE LACERDA DE OLIVEIRA^{1,2}, AND WILSON CESAR SANT'ANA^{1,3}, (Senior Member, IEEE)

¹Office of Research and Graduate Studies (PRPPG-PPGEEL), Federal University of Itajubá (UNIFEI), Itajubá 37500-903, Brazil

²Research and Development Department, Gnarus Institute, Itajubá 37502-485, Brazil

³Institute of Systems Engineering and Information Technology (IESTI), Federal University of Itajubá (UNIFEI), Itajubá 37500-903, Brazil

Corresponding author: Wilson Cesar Sant'Ana (wilson_santana@ieee.org)

This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) through the Article Processing Charge (APC) under ROR ID: 00x0ma614.

ABSTRACT Accurate estimation of flow rate in industrial centrifugal pumps is essential for diagnostics, performance monitoring, and process optimization, yet traditional frequency-domain techniques such as MCSA often rely on fixed spectral bands and assume stationarity, making them inadequate under variable-speed operation and multiphase flow conditions. This work introduces a Koopman-theoretic framework for flow-rate classification based on Hankel-embedded Higher-Order Dynamic Mode Decomposition (HODMD), enabling extraction of physically interpretable spectral-dynamical features from multichannel current, voltage, and vibration measurements. The proposed pipeline computes Koopman eigenvalues, modal energies, growth rates, temporal coherence, and spectral entropy to construct a compact descriptor of pump electromechanical behaviour, which is subsequently evaluated through supervised and unsupervised learning pipelines incorporating SMOTE balancing, nonlinear UMAP projection, and extensive hyperparameter optimization via Tree-structured Parzen Estimator Search and Stratified Kfold Cross-Validation algorithms. Using a laboratory dataset of 1304 signals spanning multiple rotational speeds, heterogeneous fluid compositions, and eleven unbalanced flow-rate range classes, the system achieves up to 97.4% cross-validation balanced accuracy and up to 91.7% balanced test accuracy while revealing coherent manifold structure aligned with physical flow-rate regimes. Although overlapping operating states and speed-dependent spectral shifts introduce classification challenges, the results demonstrate that Koopman spectral feature capture the essential dynamic structure of the pump more effectively than traditional Fourier-based approaches, highlighting their potential for robust, generalizable monitoring and diagnostics of industrial pumping systems.

INDEX TERMS Centrifugal pumps, dynamic mode decomposition, flow rate classification, higher-order dynamic mode decomposition, Koopman operator, Koopman spectral analysis, machine learning, neural networks.

I. INTRODUCTION

Centrifugal pumps are among the most widely used machines in industrial fluid-transport systems, where they operate

continuously under varying process conditions and often in harsh environments [1]. Their performance and reliability directly influence production efficiency, energy consumption, and operational safety in sectors such as oil and gas, water treatment, chemical processing, and power generation [2], [3]. Because of this critical role, real-time knowledge of the

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung¹.

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.4429600

Electric Motor Fault Detection under Nonstationary Operation using Koopman Operator Spectral Analysis

EDUARDO VAZ FAGUNDES RECH^{1,2}, LEVY ELY DE LACERDA DE OLIVEIRA², and WILSON CESAR SANT'ANA^{1,3}, (Senior Member, IEEE)

¹Office of Research and Graduate Studies (PRPG) - (PROGEL), Federal University of Itajubá (UNIFEI), Itajubá, MG 37500-900 Brazil

²R&D Department, Osama Institute, Itajubá, MG 37502-485 Brazil

³Institute of Systems Engineering and Information Technology (IESTD), Federal University of Itajubá (UNIFEI), Itajubá, MG 37500-900 Brazil

Corresponding author: Wilson C. Sant'Ana (e-mail: wilson_santana@fisee.org).

The Article Processing Charge (APC) for the publication of this research was funded by the Coordination for the Improvement of Higher Education Personnel (CAPES) (ROR ID: 00x0ma614). For open access purposes, the authors have assigned the Creative Commons CC BY license to any accepted version of the manuscript.

ABSTRACT Early detection of faults in induction motors is fundamental to ensuring reliability, reducing maintenance costs, and preventing unexpected downtime in industrial systems. Conventional Motor Current Signature Analysis (MCSA) techniques typically rely on predefined speed ranges and stationarity assumptions, which limits their effectiveness under variable-speed operation and time-varying electromechanical loading. This work proposed a Koopman-theoretic framework for motor fault classification based on Hankel-embedded Higher-Order Dynamic Mode Decomposition (HODMD), enabling extraction of compact, physically interpretable spectral-dynamical descriptors directly from observed measurements. The proposed feature extraction pipeline characterizes the motor dynamics through Koopman eigenvalues, modal energies, growth rates, temporal coherence, and entropy-based measures, forming a low-dimensional representation of the system's nonlinear behaviour. The discriminative capability of these features is evaluated using a supervised learning framework with class balancing, nonlinear manifold projection, and hyperparameter optimization via stratified cross-validation. Experiments conducted on a laboratory dataset of multiple fault conditions and variable operating speeds demonstrate that the Koopman-based representation consistently outperforms raw time-domain signals and classic Fourier-based descriptors, achieving results as high as 96.8% cross-validation balanced accuracy at 40.7% balanced test accuracy. The results highlight the ability of Koopman spectral features to capture non-steady dynamical signatures while remaining robust to speed variations, indicating strong potential for practical condition monitoring and intelligent diagnostics of industrial electric drives.

INDEX TERMS

Dynamic Mode Decomposition; Fault State Classification; Higher-Order Dynamic Mode Decomposition; Induction Motor; Koopman Operator; Koopman Spectral Analysis; Machine Learning; Neural Networks.

I. INTRODUCTION

INDUCTION motors are among the most widely deployed electromechanical energy conversion devices in modern industry, powering pumps, compressors, conveyors, and countless other critical processes. Their robustness, low cost, and relatively simple construction make them the preferred choice for continuous operation in harsh industrial environments[1]. Despite their reliability, induction motors remain subject to a wide range of electrical and mechanical faults that

progressively degrade performance, reduce efficiency, and ultimately lead to catastrophic failure if not detected at an early stage[2][3]. Unexpected motor downtime can cause substantial financial losses, production interruptions, and safety risks, which has motivated the development of advanced condition monitoring and fault diagnosis techniques.

Motor Current Signature Analysis (MCSA) has emerged as one of the most attractive monitoring approaches due to its non-intrusive nature and minimal instrumentation re-